Stochastic Environmental Research and Risk Assessment (2025) 39:4949–4983
https://doi.org/10.1007/s00477-025-03061-6

ORIGINAL PAPER

Deep learning and vine copula-based sequencing: approaches under investigation for forecasting soil temperature dynamics

Mehri Saeidinia[1] · Amir Hamzeh Haghiabi[1] · Mohammad Nazeri Tahroudi[1] · Aliheidar Nasrolahi[1] · Carlo De Michele[2]

Accepted: 10 July 2025 / Published online: 1 August 2025

Abstract
Accurate soil temperature forecasting is crucial for agricultural planning, ecosystem management, and climate resilience, yet complex interactions between atmospheric variables and soil thermal dynamics challenge traditional approaches, particularly under increasing climate variability. This study investigates the integration of vine copula models with deep learning models (Long Short-Term Memory (LSTMs), Convolutional Neural Networks (CNNs), and Gated Recurrent Units (GRUs)) for multi-depth forecasting (5, 10, 30, and 50 cm) and multi-step forecasting (1, 3, 5, 7, 10, 15, 20, and 30 days) in semi-arid regions. Vine copulas were employed to identify the most influential predictors, while deep learning models were selected and fine-tuned using Bayesian optimization. Four input variables—soil temperature, air temperature, sunshine duration, and relative humidity—selected based on vine copula rankings, were used to evaluate model accuracy for forecasting horizons of 1, 3, 5, and 7 days. Two vine copula structures (C-vine copulas and R-vine copulas). Deep learning models, particularly CNNs, performed well for short-term predictions, CNNs (3-day sequence) achieving the highest accuracy (e.g., RMSE = 1.21 °C, NSE = 0.99 at 5 cm depth). Their strong performance persisted for 7-day forecasts (RMSE = 1.58 °C, NSE = 0.98), with 1-day forecasts yielding the lowest errors (MAE = 0.67 °C). GRUs (7-day sequence) exhibited slightly higher errors (e.g., RMSE = 1.88 °C) in some configurations. LSTMs performed slightly better than GRUs but still underperformed relative to CNNs. Model accuracy declined predictably with longer forecast horizons, though CNNs maintained NSE ≥ 0.82 across all configurations. However, the vine-based models showed limitations, with errors increasing substantially at greater depths and longer forecast horizons. The C-vine and R-vine models exhibited similar performance, with errors increasing substantially at greater depths. While deep learning reduced error rates by approximately 30–50% compared to vine copula models, vine copula improvements—approximately 72%, 58%, 71%, and 72% at 5, 10, 30, and 50 cm depths, respectively. The vine copula-based method performed best at all depths with a lag-1 configuration, while deep learning models required longer sequences (3–7 days) for optimal performance. These findings highlight the potential of deep learning models for accurate soil temperature forecasting, offering valuable insights for agricultural applications.

Keywords CNN · C-vine · Environmental · LSTM · Multi-step sequences

1 Introduction

Soil temperature is a critical agro-environmental parameter that influences plant growth, atmosphere gas exchange (Onwuka 2018), and soil biological processes (Onwuka and Mang 2018). Its dynamics are governed by complex interactions between meteorological conditions and soil properties (Ahmadi et al. 2015), with temperature variations affecting water retention through altered water viscosity (Hillel 1998; Churchman 1999) and stimulating microbial activity (Allison et al. 2010; Wallenstein and Weintraub 2008). Despite its importance, direct measurements are limited due to spatial variability challenges and instrumentation requirements (Jamaniantgali et al. 2022), making predictive models essential for agricultural planning, ecosystem management, and climate resilience (Ghamariadeh et al. 2024; Kisi et al. 2019; Tabari et al. 2012).

Mehri Saeidinia
saeedinia.m@lu.ac.ir

Amir Hamzeh Haghiabi
haghiabi.a@lu.ac.ir

Mohammad Nazeri Tahroudi
nazeri.mh@lu.ac.ir

Aliheidar Nasrolahi
nasrolahi.a@lu.ac.ir

Carlo De Michele
carlo.demichele@polimi.it

[1] Department of Water Engineering, Lorestan University, Khorramabad, Iran

[2] Department of Civil and Environmental Engineering, Politecnico di Milano, Milan, Italy

emerged as an efcient (Ahmadi et al. 2019; Mehdizadeh 2020). Recent deep learning approaches for various soil-related applications (Dong et al. 2022a, b; Chen et al. 2022), including soil temperature prediction at different depth levels (Song et al. 2024), have demonstrated remarkable capabilities in capturing complex nonlinear patterns in soil dynamics (Wang et al. 2023). In soil temperature modeling, studies have compared Multiple Linear Regression (MLR), Artificial Neural Networks (ANN), and Adaptive Neuro-Fuzzy Inference System (ANFIS) models, with ANFIS often showing superior performance (e.g., $R^2 = 0.96$, MAPE = 5%). Subsequent studies have demonstrated various approaches: a hybrid framework combining strong results with an Ensemble Empirical Mode Decomposition-Convolutional Neural Network-Bidirectional Long Short-Term Memory model, while Wang et al. (2021) utilized Deep Echo State Networks (Deep ESN) for predicting soil temperatures, achieving Root Mean Square Error (RMSE) reductions of 30–60% compared to traditional machine learning models for 5–10 cm depths. Comparative studies have also been conducted, such as Elbeltagi et al. (2022), who found that Extreme Gradient Boosting (XGBoost) outperformed other algorithms including Ridge Regression, Support Vector Machine, Elastic Net, Deep Neural Networks (DNN), and Random Forest (RF), while Gan et al. (2023) identified Bidirectional Long Short-Term Memory (BiLSTM) networks as the best performing model for daily soil temperature prediction. However, Bayrossoglu et al. (2023a, b) and Devi et al. (2023) found that Support Vector Regression (SVR) and Multivariate Adaptive Regression Splines (MARS) models were more suitable for specific climate zones and scale analysis, respectively. Most recently, Yu et al. (2024) developed a Multilayer Perceptron-Sperm Swarm Optimization (MLP-SSO) model that outperformed standard ANN models, while existing AI models tend to struggle with extreme temperature events and lack physical consistency (Kim et al. 2024).

Conventional AI approaches for soil temperature forecasting primarily focus on capturing complex patterns in time series data, often overlooking the underlying dependencies between variables. This limitation can be addressed through copula theory, which provides a mathematical framework for modeling dependencies by coupling marginal distributions (De Michele and Salvadori 2003; Salvadori et al. 2016; De Michele, De Michele and Salvadori 2016; Salvadori et al. 2007). Recent applications demonstrate copulas' unique ability to model complex nonlinear dependencies in environmental systems (Zhang et al. 2019; Lü et al. 2020), while offering flexible uncertainty quantification (Cai et al. 2021). However, despite their theoretical advantages, copula applications in soil temperature modeling remain limited, with most studies focusing on single-depth predictions and lacking comprehensive comparisons with modern machine learning approaches (e.g., LSTMs, CNNs) and hybrid frameworks. This gap is particularly significant given the importance of multi-depth soil temperature profiles in agricultural and environmental modeling, and the potential benefits of combining copula theory with deep learning architectures—a critical knowledge gap, as existing studies typically treat these approaches separately rather than exploring their integration. Hybrid models, such as CNN-LSTM-SVR (Wang et al. 2020) or Deep ESN (Wang et al. 2021), have shown promise in capturing both temporal and spatial dependencies, but their potential is often limited by relying solely on copula applications without leveraging advanced machine learning techniques. Furthermore, while copulas have been successfully applied in various environmental fields, including hydrology and climate studies, their integration with deep learning models for soil temperature prediction remains largely unexplored. This study addresses these gaps by developing a novel framework that integrates copula theory with deep learning models, specifically Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) architectures, for multi-depth soil temperature prediction. The framework leverages copulas to model complex soil temperature dependencies across different depths and time lags, while deep learning models capture nonlinear temporal patterns. This integration aims to enhance prediction accuracy, especially during extreme temperature events, and provide more physically consistent predictions. The study's objectives are to: (1) identify optimal copula families for modeling soil temperature dependencies, (2) develop and evaluate hybrid copula-deep learning models for multi-depth soil temperature prediction, (3) assess the models' performance under different temperature regimes, including extreme events, and (4) compare the proposed approach with standalone deep learning models and traditional statistical methods. Furthermore, copulas to model complex soil temperature dependencies across different depths and time lags, while deep learning models capture nonlinear temporal patterns. This integration aims to enhance prediction accuracy, especially during extreme temperature events, and provide more physically consistent predictions. The study's objectives are to: (1) identify optimal copula families for modeling soil temperature dependencies, (2) develop and evaluate hybrid copula-deep learning models for multi-depth soil temperature prediction, (3) assess the models' performance under different temperature regimes, including extreme events, and (4) compare the proposed approach with standalone deep learning models and traditional statistical methods.

2 Material and methods

2.1 Study region and dataset

The study area is located in Khanmanabad, Lorestan Province, Iran, at coordinates 33°31′N, 48°21′E, with an elevation of 1285 m above sea level. The region experiences a semi-arid climate with significant seasonal temperature variations, ranging from −15 °C in winter to 40 °C in summer. Soil temperature data were collected at depths of 5, 10, 20, 30, 50, and 100 cm using calibrated sensors with an accuracy of ± 0.1 °C. The dataset spans from January 2015 to December 2020, with hourly measurements. Meteorological data, including air temperature, precipitation, wind speed, and relative humidity, were also recorded at a nearby weather station. The dataset was divided into training (70%), validation (15%), and testing (15%) sets, ensuring that each set contained representative seasonal patterns.

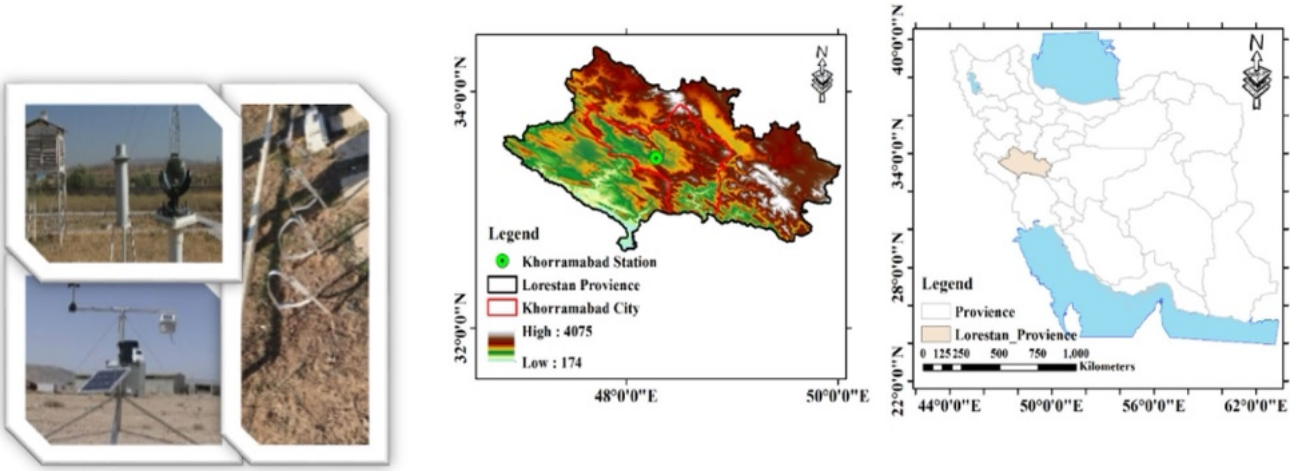

Fig. 1 Location of Khorramabad station in Iran

Table 1 Statistics of the considered variables observed at a daily scale for the Khorramabad station

Variables	Depth	Mean	Standard deviation	Skewness
Soil temperature	5 cm	21.21	11.76	0.16
	10 cm	21.13	11.43	0.21
	30 cm	20.08	10.07	0.08
	50 cm	20.17	8.97	0.08
Air temperature (°C)	–	17.38	8.94	0.02
Relative humidity (%)	–	44.80	19.96	0.24
Wind speed (m/s)	–	6.22	2.39	1.50
Sunny hours (hrs)	–	8.44	3.60	-0.85
Rain (mm)	–	1.29	5.02	6.95

variations, with a mean of 21.21 °C and a decreasing trend of 0.16 °C, indicating a mild thermal gradient. Air temperature averages 17.38 °C, suggesting a relatively moderate climate. Relative humidity is at 44.80%, creating a dry environment, while wind speed averages 6.22 m/s with occasional strong gusts. The average of 8.44 sunny hours per day, combined with negligible rainfall (1.29 mm) and a significant positive skewness (6.95) in rainfall, suggests the region experiences long dry periods but intense rainfall events.

2.2 Deep learning models

Deep learning is a specialized subset of machine learning that focuses on learning deep representations of data, emphasizing multiple levels of abstraction to improve performance in complex tasks (Chollet 2017; Alzubaidi et al. 2021). This study employs three deep learning models: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU).

2.2.1 Convolutional neural network (CNN) model

Convolutional Neural Networks (CNNs) have shown success in computer vision, but they are also effective in analyzing time series data by treating time as the spatial dimension (Kiranyaz et al. 2021). CNNs include convolutional layers that apply learnable filters to process input data, capturing local patterns in the time series, reducing computational complexity, and minimizing the risk of overfitting (Zhang et al. 2024). The architecture typically consists of multiple convolutional layers followed by pooling layers, which progressively extract higher-level features from the input data (Li et al. 2024). In this study, the CNN model is configured with 128 filters in the first convolutional layer, 64 filters in the second layer, and a kernel size of 3 for both layers. The pooling layers use a pool size of 2, and the model is trained with a batch size of 32 and 100 epochs. The Adam optimizer is employed with a learning rate of 0.001, and the mean squared error (MSE) is used as the loss function.

2.2.2 Long short-term memory (LSTM) model

LSTM, developed by Hochreiter and Schmidhuber (1997), is designed to handle long-term dependencies in sequential data, addressing the limitations of traditional RNNs (Hochreiter and Schmidhuber 1997). It overcomes the issue of vanishing gradients by introducing a novel unit structure consisting of a memory cell and three gates: input, forget, and output gates (Graves et al. 2013). These gates regulate the flow of information, allowing the network to selectively retain or discard information over long sequences (Sherstinsky 2020). The LSTM architecture is particularly well-suited for time series forecasting due to its ability to capture temporal dependencies and patterns (Kizito et al. 2021; Al-Musaylh et al. 2022). The LSTM model in this study is configured with 128 units in the first layer and 64 units in the second layer. The model is trained with a batch size of 32 and 100 epochs. The Adam optimizer is employed with a learning rate of 0.001, and the mean squared error (MSE) is used as the loss function.

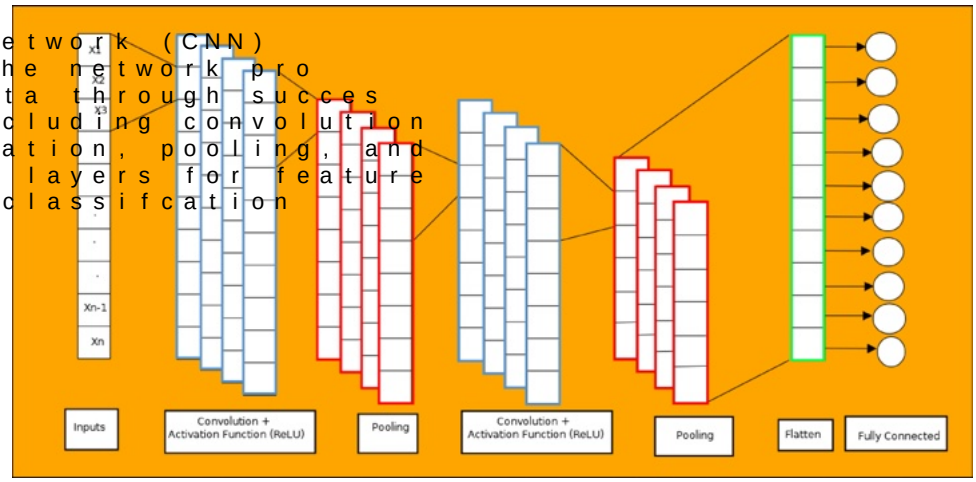

Fig. 2 Schematic of a Convolutional Neural Network (CNN) architecture. The network processes input data through successive layers, including convolution with ReLU activation, pooling, and fully connected layers for feature extraction and classification

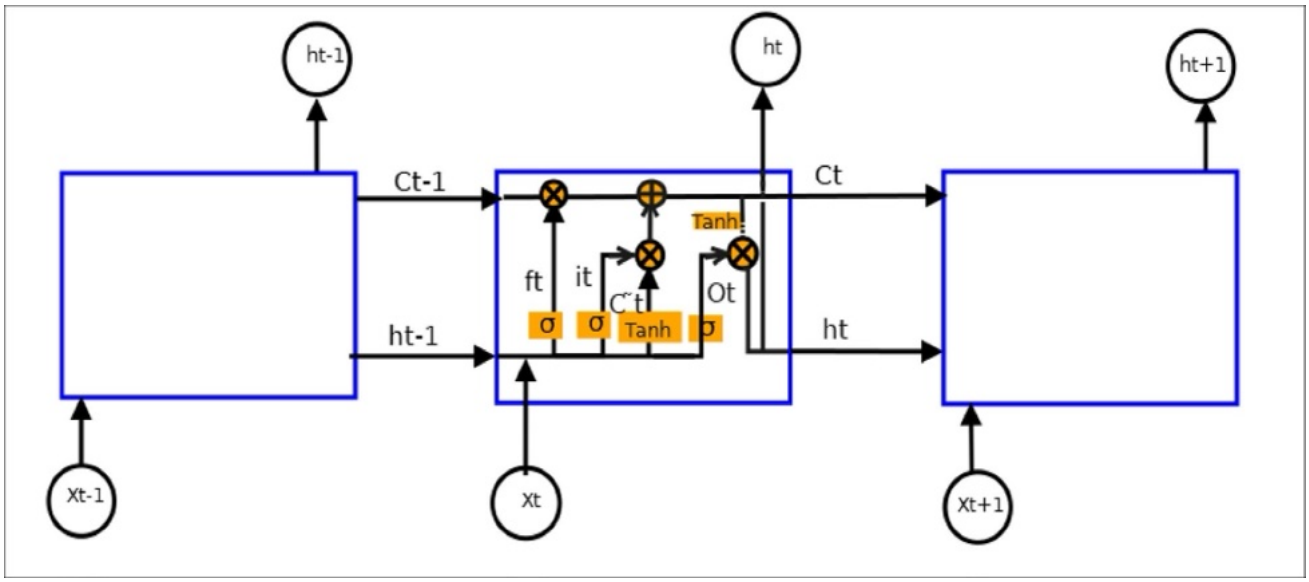

Fig. 3 Structure of an LSTM unit, showing the flow of information between consecutive units. The architecture highlights the forget gate, input gate, and output gate, which regulate the cell state and hidden state

LSTM architecture, a sequential neural network is used, consisting of a single LSTM layer followed by a Dropout layer to prevent overfitting, and a Dense layer with a linear activation function to produce the final output. The model is compiled using the Adam optimizer and the mean squared error loss function, which is suitable for regression tasks. The model is trained for 200 epochs with a batch size of 64, and the validation data is used to monitor the model's performance during training. The LSTM layer has 50 units, which determines the dimensionality of the output space. The input shape is specified as (1, 4), indicating that the model expects input sequences with one time step and four features. The Dropout layer has a dropout rate of 0.2, meaning that 20% of the input units are randomly set to zero during training to reduce overfitting. The Dense layer has one unit, which produces the final output of the model. The model is trained using the fit function, which takes the training data, validation data, number of epochs, batch size, and verbosity level as arguments. The verbosity level is set to 2, which displays one line per epoch during training. After training, the model is used to make predictions on the test data, and the predictions are stored in the variable y_pred. The model's performance is evaluated using the mean squared error metric, which measures the average squared difference between the predicted and actual values. The lower the mean squared error, the better the model's performance. The model's architecture and training process are designed to capture the temporal dependencies in the input data and make accurate predictions of the target variable.

2.2.3 Gated recurrent unit (GRU) model

The GRU is a variant of gated recurrent neural networks that features a more streamlined architecture. It simplifies the design by combining the forget and input gates into a single update gate and merging the cell state and hidden state. This results in a model that is computationally more efficient and faster to train, while still capturing long-term dependencies in sequential data. The GRU model's key operations are governed by the following equations:

2.2.4 Preparing the data

The data was prepared for the LSTM model by first checking for missing values. Since the dataset is complete and ready for this purpose, the SimpleImputer from Python was utilized. It guarantees that the model can be trained on a full dataset without any missing values. The data was then split into training and testing sets, with 80% of the data used for training and 20% for testing. The input features were scaled using the StandardScaler from Python, which standardizes the features by removing the mean and scaling to unit variance. This is important for the LSTM model because it helps the model converge faster and prevents the model from being biased towards features with larger scales. The target variable was also scaled using the same StandardScaler. The scaled data was then reshaped into the required format for the LSTM model, which is a 3D array with the dimensions (samples, time steps, features). In this case, the time steps is set to 1, and the features is set to 4. The reshaped data was then used to train the LSTM model.

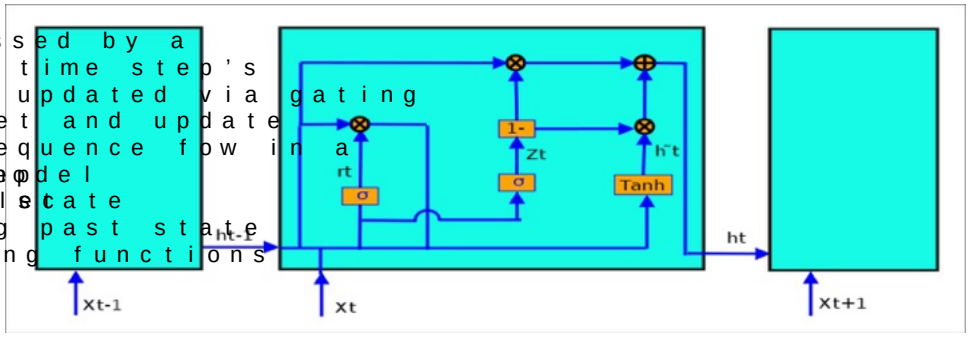

Fig. 4 Sequential inputs (X_{t-1}, X_t, X_{t+1}) processed by a GRU, where each time step's hidden state is updated via gating mechanisms (reset and update gates). Input sequence flow in a GRU: At each time step, the model computes the hidden state by selectively combining the past state (h_{t-1}) and new input using gating functions

2.3 Data normalization

Preprocessing the data in a way that a deep learning (DL) model can easily ingest is a key step. An effective approach to manage data with different scales is normalization, which centers the data around zero and has a standard deviation close to one (Chollet 2017). This helps the model converge faster during training and prevents features with larger values from dominating the learning process. In this study, the Min-Max scaler from the scikit-learn library (Pedregosa et al. 2011) was used to normalize the input features. This scaler transforms each feature to a specified range, typically between 0 and 1, by subtracting the minimum value and dividing by the range (the difference between the maximum and minimum values). The formula for Min-Max scaling is:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X is the original feature value, X_{min} is the minimum value of the feature, and X_{max} is the maximum value of the feature. This ensures that all features contribute equally to the model's learning process, regardless of their original scales.

2.4 Model training and test

The forecasts were based on training time series models using daily measurements of river discharge, air temperature, wind speed, relative humidity, and rainfall, collected from January 1, 1993, to December 31, 2019. To ensure unbiased evaluation, the GRU model was divided into training, validation, and test sets, with 70% of the data (1 January 1993 to 31 December 2012) used for model training and the remaining 30% (1 January 2013 to 31 December 2019) reserved for evaluating the model's performance on unseen data. The components of the deep learning models are summarized in Table 1.

This study presents an advanced deep learning (DL) model that integrates Copula functions. Hyperparameters can be defined using two methods: assigning a range of values and evaluating model performance to find the best settings, or setting standard values based on previous studies and relevant factors (Imani et al. 2022).

In this study, some of the number of epochs and GRU layers, were selected through manual testing across a wide range of values. A validation set is used for each model to determine the optimal hyperparameters that achieve the best performance. The model's performance is then evaluated on the test set, which is kept separate from the training and validation sets to ensure an unbiased assessment of its generalization ability. Additionally, techniques like early stopping and learning rate reduction are employed to prevent overfitting and optimize the training process. The Adam optimizer and Mean Absolute Error (MAE) loss function are utilized, along with callbacks for early stopping and a CSV logger for tracking the training process. The model similarly sets a batch size of 32 for training, balancing computational efficiency and model stability. The GRU model was trained for 150 epochs, with a learning rate of 0.001, and a dropout rate of 0.2 to prevent overfitting. The model's performance was evaluated using the validation set after each epoch, and the best model was saved for testing. The components of the deep learning models are summarized in Table 1.

2.4.1 Key parameters in DL models

There are several key parameters in the deep learning models that influence both the training process and model performance such as subsampling (Batch Size), sequence length (Lookback Window), and epochs. Batch size refers to the number of samples processed before the model updates its weights, influencing training stability and memory usage. Sequence length determines how many previous time steps are considered for predictions, allowing the model to learn from longer historical patterns. While increasing model complexity. Epochs indicate how many times the dataset passes through the network during training.

2.5 Copula functions

Suppose that $x_1, \ldots, x_n$ are realizations of random variables, the empirical distribution function for x_j is $F_{X_j}(x_j) = \frac{1}{n+1}\sum_{i=1}^{n} I(x_{ij} \le x_j)$, where I is the indicator function. The empirical copula function is then defined as $C_n(u_1, \ldots, u_n) = F_{X_1,\ldots,X_n}(F_{X_1}^{-1}(u_1), \ldots, F_{X_n}^{-1}(u_n))$, where $u_j = F_{X_j}(x_j)$ for $j = 1, \ldots, n$. The empirical copula function is a non-parametric estimator of the copula function that does not assume any specific parametric form for the copula. It is based on the observed data and provides a flexible way to model the dependence structure between variables. The empirical copula function is useful when the true copula is unknown or when the data exhibit complex dependence patterns that cannot be captured by parametric copulas. It is also robust to outliers and can handle non-linear dependence structures. The empirical copula function is widely used in various fields, including finance, hydrology, and environmental science, for modeling multivariate dependence structures and simulating joint distributions.

Table 2 Comparative analysis of LSTM, GRU, and CNN architectures for soil temperature prediction

Component	LSTM	GRU	CNN
Input features	Air temp, sunshine duration, relative humidity (Kendall's Tau-selected)	Same as LSTM	Same as LSTM
Dara preprocessing	Handle missing values, Scale Data, target variable creation	Same as LSTM	Same as LSTM
Model architecture	3 stacked LSTM layers 1 Dropout (50%) 1 Dense output	3 stacked GRU layers 1 Dropout (50%) 1 Dense output	3 Conv1D layers 2 MaxPool1D 1 GlobalAveragePooling
Key parameters	Units: 150 Dropout: 25% (LSTM1–2), 50% (LSTM3) Adam optimizer MAE loss	Same as LSTM (with GRU cells)	Filters: 1–30 Kernel sizes: 24, 12, 6 Pool size: 2 Adam optimizer MAE loss
Training	Batch size: 240 Epochs: 50	Same as LSTM	Same as LSTM

variables $U_j = F_{X_j}(x_j)$ in [0, 1], then there is a d-dimensional copula $C_{U_1,...,U_n}$ such that:

$$C_{U_1,...,U_n}\left(F_{X_1}(x_1),...,F_{X_n}(x_n)\right) = F_{X_1,...,x_n}(x_1,...,x_n) \quad (1)$$

According to Nelsen (2006), estimating parameters is crucial in implementing copula functions. This challenge can be addressed by utilizing the diagonal section of the copula, as suggested by Salvadori et al. (2007) and Jaworski (2010). For two-variable scenarios, standard bivariate copulas are typically employed. However, in higher dimensions, more complex structures like nested vine copulas are necessary (Mirabbasi et al. 2012; Ramezani et al. 2020; Nazeri Tahroudi et al. 2020, 2023; Mottaghi et al. …). Vine copulas can be constructed in various ways, categorized into D-type, C-type, and R-type (… et al. 2023). Among these, Bedford and Cooke (2001) specifically identified and defined C-vine and D-vine copulas. They also provided a formula for the density of an n-dimensional distribution in terms of these vine copulas. For instance, the density function $f(x_1,...,x_n)$ corresponding to a D-vine structure can be expressed mathematically as follow:

$$\prod_{k=1}^{n} f(xk) \prod_{j=1}^{n-1}\prod_{i=1}^{n-j} c_{i,i+j|i+1,....,i+j-1}\left\{F(x_i|x_{i+1},...,x_{i+j-1}), F(x_{i+j}|x_{i+1},...,x_{i+j-1})\right\} \quad (2)$$

where j represents the trees that i runs around the … each tree. In the D-vine, no … T_j is connected to more than two edges … T_j is connected to n—j edges, and i… each tree. The n-dimensional …

$$\prod_{k=1}^{n} f(xk) \prod_{j=1}^{n-1}\prod_{i=1}^{n-j} c_{i,j+i|1,....,j-1}\left\{F(x_j|x_1,...,x_{j-1}), F(x_{j+i}|x_1,...,x_{j-1})\right\} \quad (3)$$

2.5.1 Conditional density in the 4-dimensional case

Upon selecting the vine copu… the conditional density func… and assess soil temperature a… chosen dependent variables (R… the conditional density $c(u_4|u_1,u_2,u_3)$, which represents the probability density of u_4 given u_1, u_2, and u_3, can be determined using the following for… This method enables a more … temperature variations at dif… the interdependencies among t…

$$c(u_4|u_1,u_2,u_3) = \left.\frac{\partial^4 C(u_1,u_2,u_3,u_4)}{\partial u_1 \partial u_2 \partial u_3 \partial u_4}\right|_{u_1,u_2,u_3} \quad (4)$$

Specifically, u_1, u_2, u_3 are T, Sun, and RH respectively, and u_4 is soil temperature a… obtain sampled pairs, the fol… carried out:

$$\begin{aligned} &First: Sample\ w_j \overset{i.i.d.}{\sim} U[0,1],\ j=1,...,d \\ &Then: u_1 := w_1 \\ &u_2 := C^{-1}_{2|1}(w_2|u_1) \\ &\vdots \\ &u_d := C^{-1}_{d|d-1,...,1}(w_d|u_{d-1},...,u_1) \end{aligned} \quad (5)$$

To determine the conditional copula $C_{j|j-1,...,1}$ for $j=1,...,d$, … (it is necessary … This approach … R-vine … (Czado 2019).

Vine copulas are flexible … complex, high-dimensional dep… They decompose multivariate d… of bivariate copulas (pairwis… a graphical model called a "v… representation of asymmetric, …

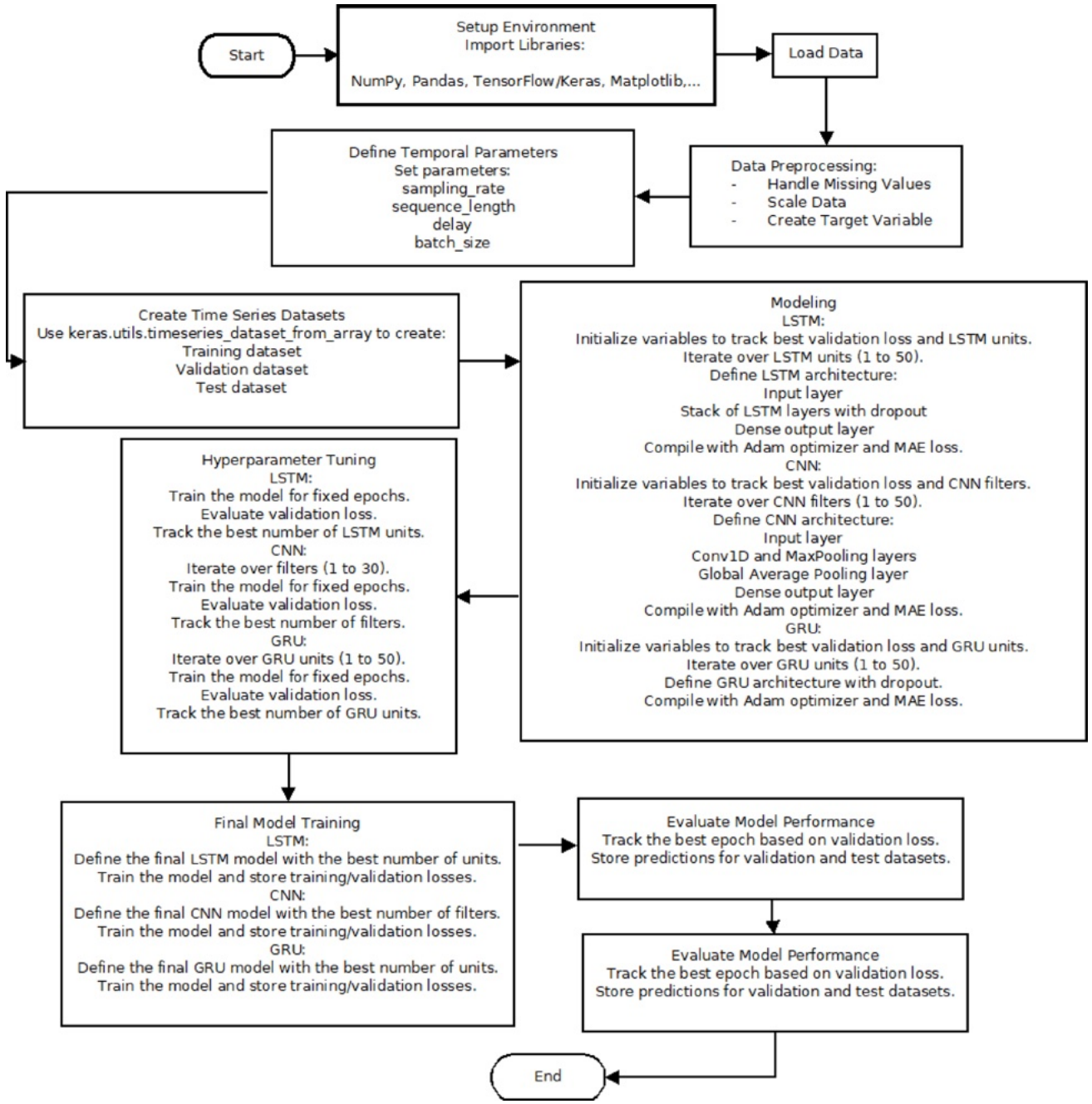

Fig. 5 End-to-end workflow for time-series modeling using LSTM, CNN, and GRU architectures. The approach begins with data loading, preprocessing, and dataset creation, followed by hyperparameter tuning of units/filters via validation loss),

relationships that traditional correlation-based methods (e.g., Gaussian) cannot capture, ensuring the model is sensitive to complex dependencies involving nonlinear interactions and tail behaviors. Additionally, the use of sliding windows (e.g., deeper layers may vary to capture broader trends, while shallower layers are tuned to focus on short-term fluctuations) ensures that temporal dependencies are explicitly modeled, enhancing the model's ability to learn these nuanced dependencies.

Unlike conventional time-series models (e.g., ARIMA) or machine learning (e.g., ANN), vine copulas

2.6 Model evaluation criteria

The Root Mean Square Error (RMSE), Nash–Sutcliffe efficiency coefficient (NSE), and Mean Absolute Error (MAE) are widely used metrics for comparing the studied models and evaluating the forecasting approach. These statistical measures provide a quantitative assessment of the model accuracy and reliability (Nash and Sutcliffe 1970; Singh and Zhang 2007).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2} \quad (6)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i| \quad (7)$$

$$NSE = 1 - \frac{\sum_{i=1}^{N}(x_i - y_i)^2}{\sum_{i=1}^{N}(x_i - x_{ave})^2} \quad (8)$$

where x_i and y_i are equal to the simulated and observed values, respectively. N is the number of data points, and x_{avg} is also the average of observed values.

3 Results and discussion

The analysis is focused on daily soil temperature in the study area, covering a period of 10 years on a daily scale. Forecasting lengths of 1, 3, and 5 days ahead are considered. The results are presented and validated for forecasting horizons of 1, 3, and 5 days. To achieve this objective, the most influential meteorological variables and their lagged values are examined. Based on the highest Kendall's tau coefficient (Kendall 1938), air temperature at lag times of 1, 2, 3, 23, and 24 hours, and relative humidity at lag times of 1, 2, 3, 22, and 23 hours are selected as inputs for the model. It was demonstrated by Nazeri Tahrudd et al. (2021) that air temperature and solar radiation are the most significant factors influencing predictions of soil temperature, while precipitation can be excluded from the input variables due to its weak correlation with soil temperature. Additionally, trend illustrated in Fig. 5 indicates that precipitation has a minimal impact on soil temperature variations. Therefore, the lagged values of soil temperature are determined based on a Kendall correlation coefficient greater than 0.4 (Nazeri Tahrudd et al. 2021). Consequently, only meteorological variables with a dependence greater than 0.4 are considered for modeling and forecasting of soil temperature at various depths. Additionally, various deep learning models and tree

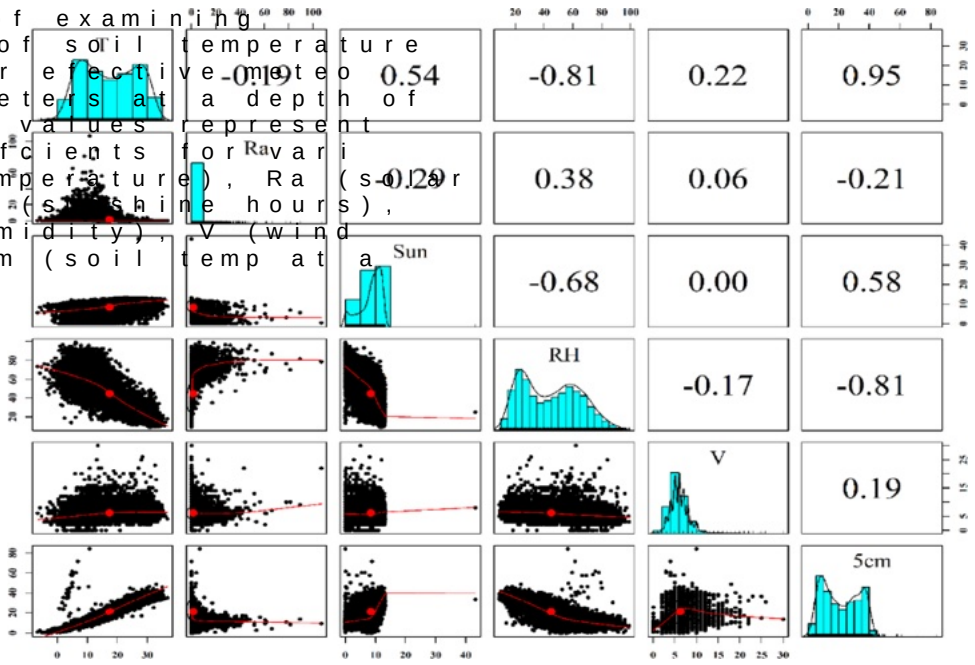

Fig. 6 The results of examining the dependence of soil temperature values and other effective meteorological parameters at a depth of 5 cm. Numerical values represent correlation coefficients for variables T (air temperature), Ra (solar radiation), Sun (sunshine hours), RH (relative humidity), V (wind speed), and T5cm (soil temperature at a depth of 5 cm)

3.1 Forecasting soil temperature at different depths using deep learning models

3.1.1 The performance of deep learning models at a depth of 5 cm

The performance of deep learning models in forecasting daily soil temperature at a depth of 5 cm was evaluated using sequence lengths of 1, 3, 5, and 7 days for prediction horizons of 1, 7, 15, and 30 days. The results are presented in Fig. 7.

It can be observed from Fig. 7 that as the forecast period extended from 1 to 30 days, accuracy decreased and the model's efficiency ... CNN outperformed LSTM and GRU, achieving the best results at a 3-day sequence length (RMSE: 1.1 °C, NSE: 99%). The difference between this model and the LSTM and GRU models ...

for one-day-ahead forecasting respectively, indicating the ... models in predicting one day ... the performance of all model... CNN maintained its lead with ... (95–96%) and the lowest error ...

GRU (with a 7-day sequence le... the 15-day forecast, CNN ...

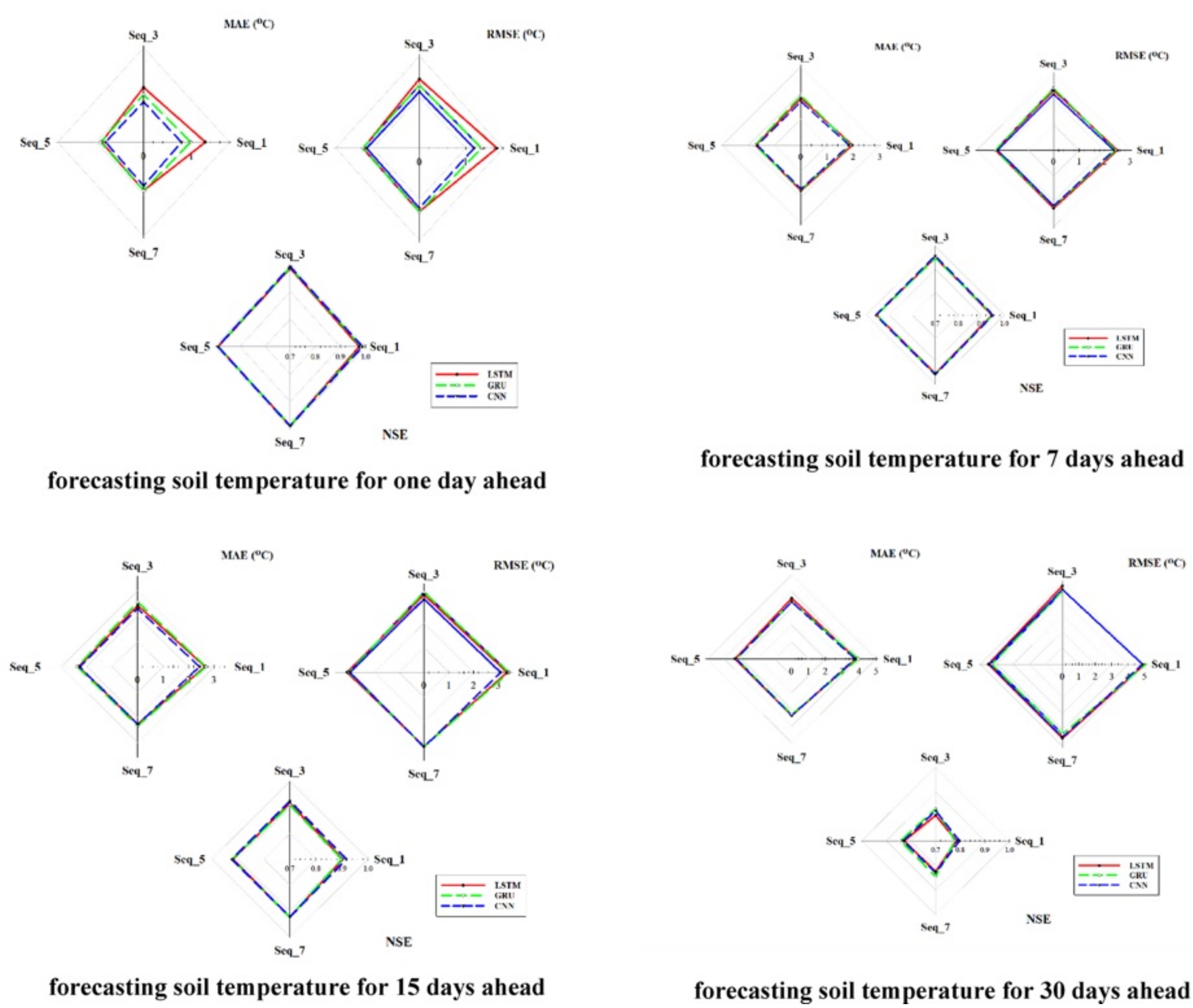

Fig. 7 Performance evaluation of deep learning models for soil tempe... across prediction horizons (1/7/15/30 days). Key metrics: RMSE (...

3.30, NSE: 85%), despite pivotwse rovfeorrælclas ltciwmeg r, pe rmfaoc mag c
compared to shorter horizarcs urNoy ahyd, robes sRMSES, vapaurets
of this model were 4% and s5e%t sl,owmeark itrhga nitt haspowe rfthe be
performing CNN and LSTM mo(d2e0l2s1)l sd,emrmonspetcatiedvelty haAzdeaepte
al. (2019) emphasized the importearn cfeo roefcasetiqungenperfeormaahce f
in enhancing model perforalsnce hiywhthgCtNNd atthaat 3u-sdiaryg
sequence length emerging aismptrhoevemso satc cruerlaicayb,leamfobr fomedim
term forecasting. CNNs halvSeTMbeæmchwiitdeeclyuræpsp ldieeldivaes ha in
efective model for forecastCimngfbeynvoeriotuservaseta(Cshe)sfa
diferent domai(2n02hs,b). Wamgonestrealtse,d(atshas8thowenveianl Ftihgat uncerta
CNNs efectively extract artdhemafporeecaasgirhgpofiezaotnu,rewsithrmma
load sequenc(e2s021.) Cphreopgoseetd apIrsdeyaeaheads (e.g., one day ah
probabilistic residential wideaed ifotrecvaasltsingfomeltchrogd-tuesrim
novel deep learning modeluscianlgletdheCoGnRvUoImuotdieoln)a.l TNheiusrable
Network with Squeeze-and-Excaictiitoantsiosnumchodaulesshq(CNN-tSEm), f
integrated with micrometeaarreloigdieaall fobartaimmNgduyaeae-Dtuecie
al. (2024) introduced a Multi 2oDr-CrNeNalf-otrimsehcornetsotuercme PaVlocat

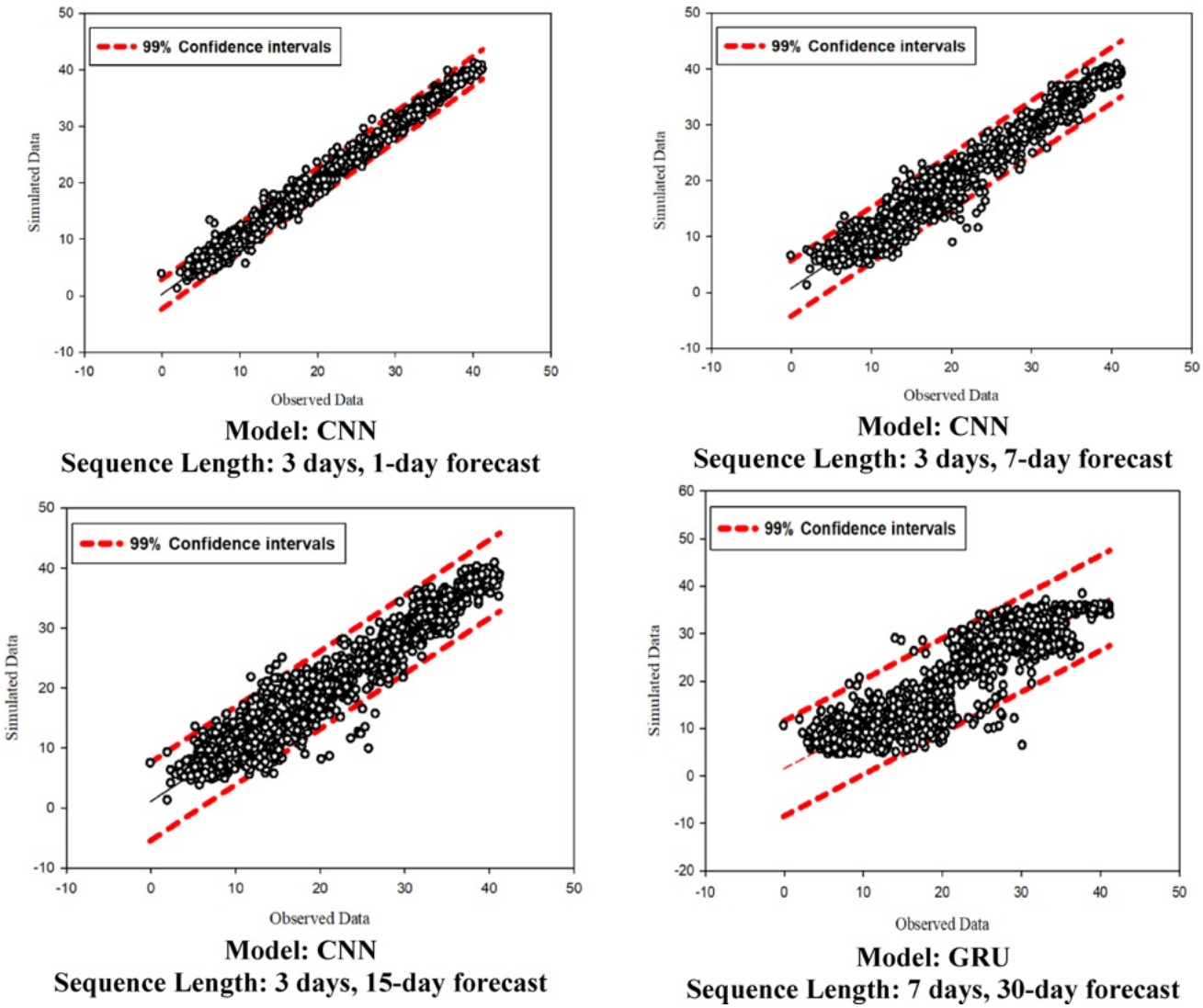

Fig. 8 99% Confdence interval of soil temperature forecasting at a
observed values falling predominantly within the predicted confd

despite their wider interval, the values for RMSE, MAE, and NSE are slightly ou
ning, risk assessment, and 2.5 °C, and NSE: 96%) is slightly ou

GRU. Notably, similar to the

performance of the model was

ence in RMSE values between

than 2%. For 15-day forecasts

3.1.2 The performance of deep learning models at a depth of 10 cm

The performance of deep le[illegible] in forecasting soil temperature at a 10 cm depth, using sequence lengths of 1, 3, 5, and 7 days for forecasts of 1, 7, 15, and 30 days, is illustrated in Figure 9.

As shown in Figure 9, for 1-day forecasts, GRU with a 7-day sequence length achieved the highest accuracy (MAE: 0.67 °C, RMSE: 0.95 °C, NSE: 99%), followed closely by LSTM with a 3-day sequence length (MAE: 0.68 °C, RMSE: 0.96 °C, NSE: 99%). Notably, the RMSE of these models were 17% and 13% lower than that of CNN (with a 7-day sequence length) and LSTM (with a 5-day sequence length), respectively. For 7-day forecasts, GRU with a 3-day sequence length

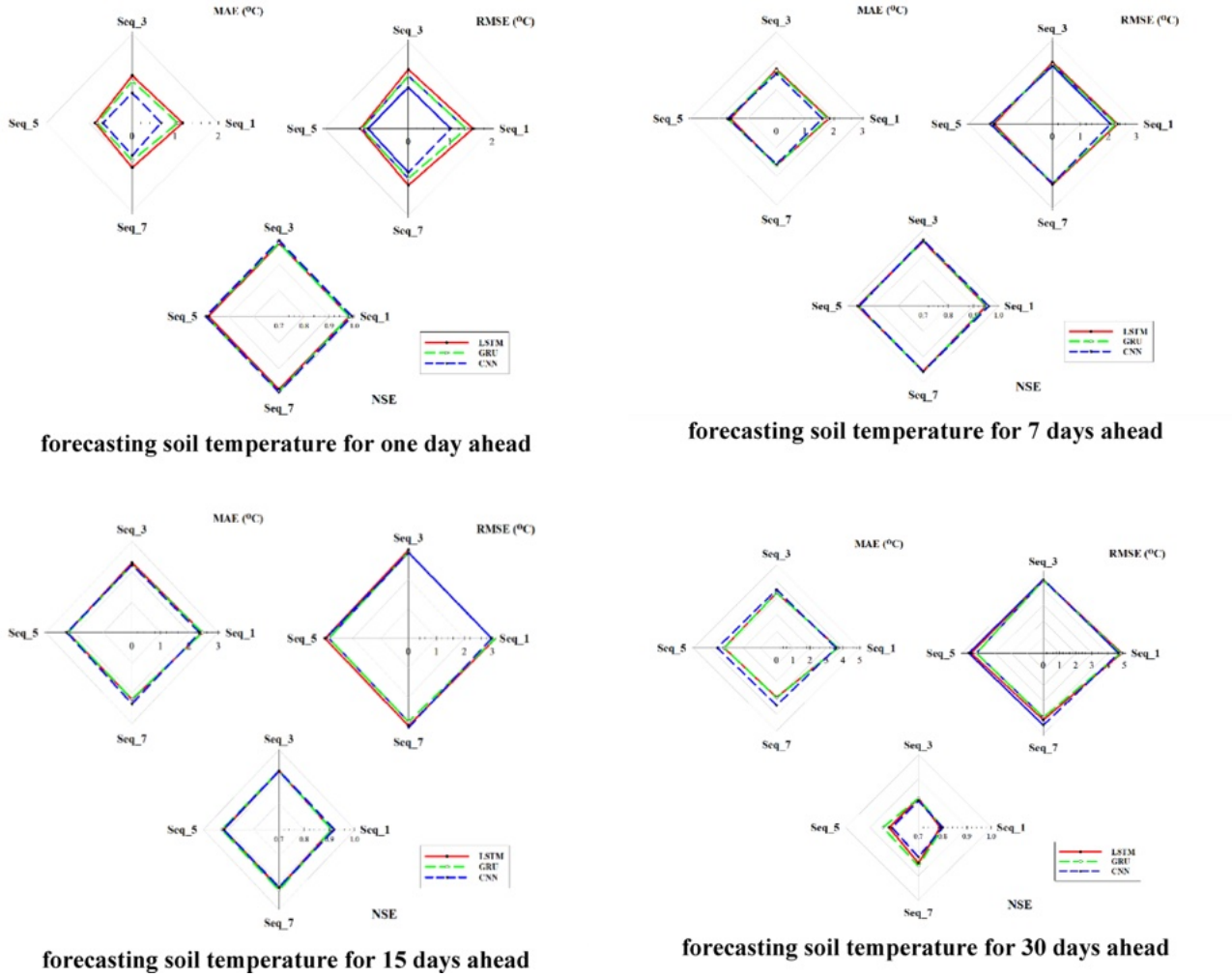

Fig. 9 The performance of deep learning models in forecasting soil te
for forecast horizons of 1, 7, 15, and 30 days. Key metrics: RMS

(30-day). The results of [illegible] perature at a 10 cm depth [illegible] learning models, [illegible] casting results at a 10 cm [illegible] demonstrating that as the [illegible] error and associated uncertainty [illegible] differences in performance [illegible] model selection, with CNN [illegible] precision and GRUs excelling [illegible] lengths was very close (with [illegible] 4%), CNN with a 3-day sequen (NSE: 94%, [illegible] ing its consistency. For 30-d [illegible] daily soil temperature at [illegible] Fig. [illegible]. As shown, for 1-day [illegible] sequence length achieved [illegible] [illegible]

3.1.3 Performance of deep learning models at a depth of 30 cm

The performance of deep [illegible] daily soil temperature at [illegible] Fig. [illegible]. As shown, for 1-day [illegible] sequence length achieved [illegible] [illegible]

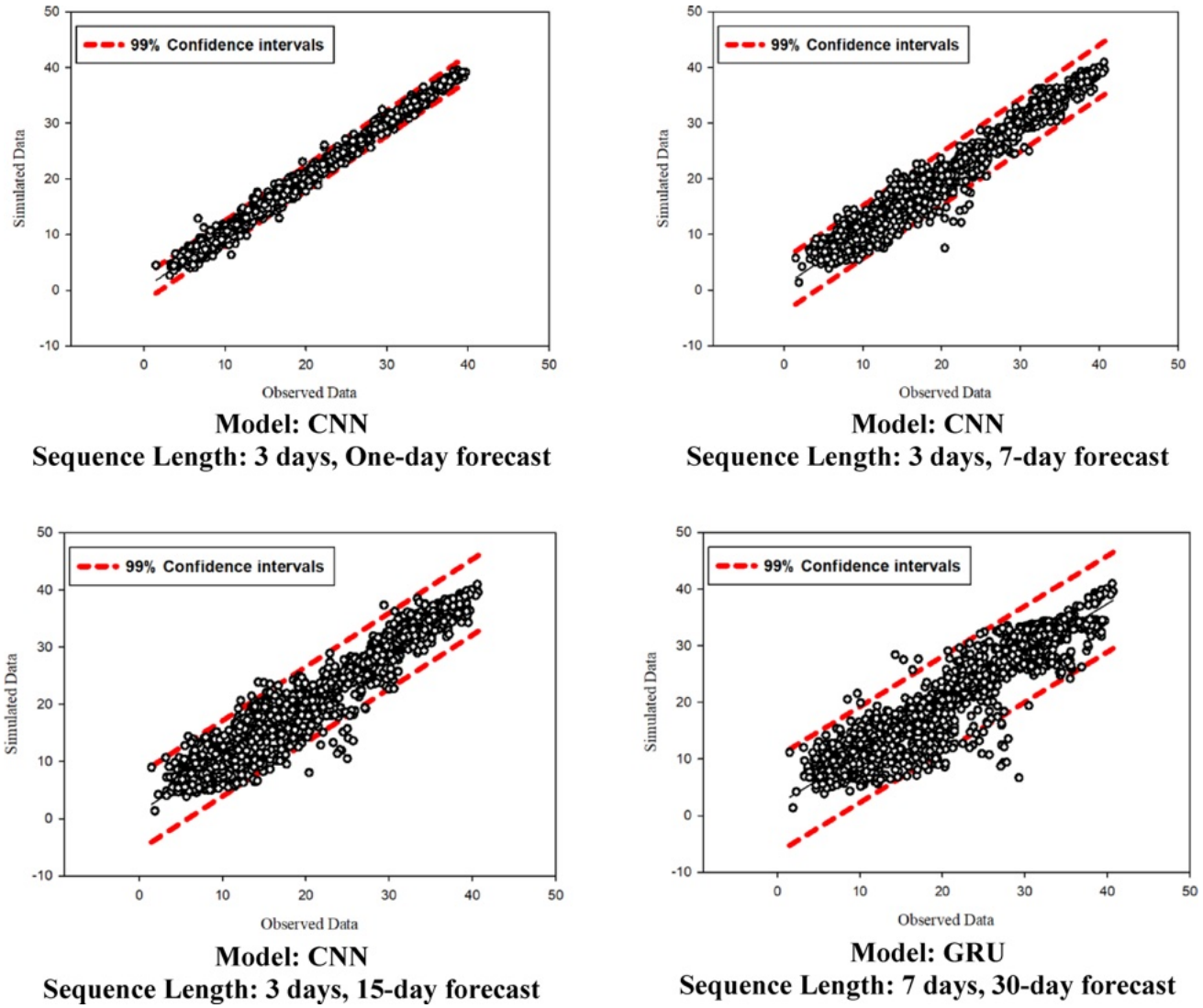

Fig. 10 99% Confdence interval of soil temperature forecasting at a [illegible] observed values falling predominantly within the predicted confd

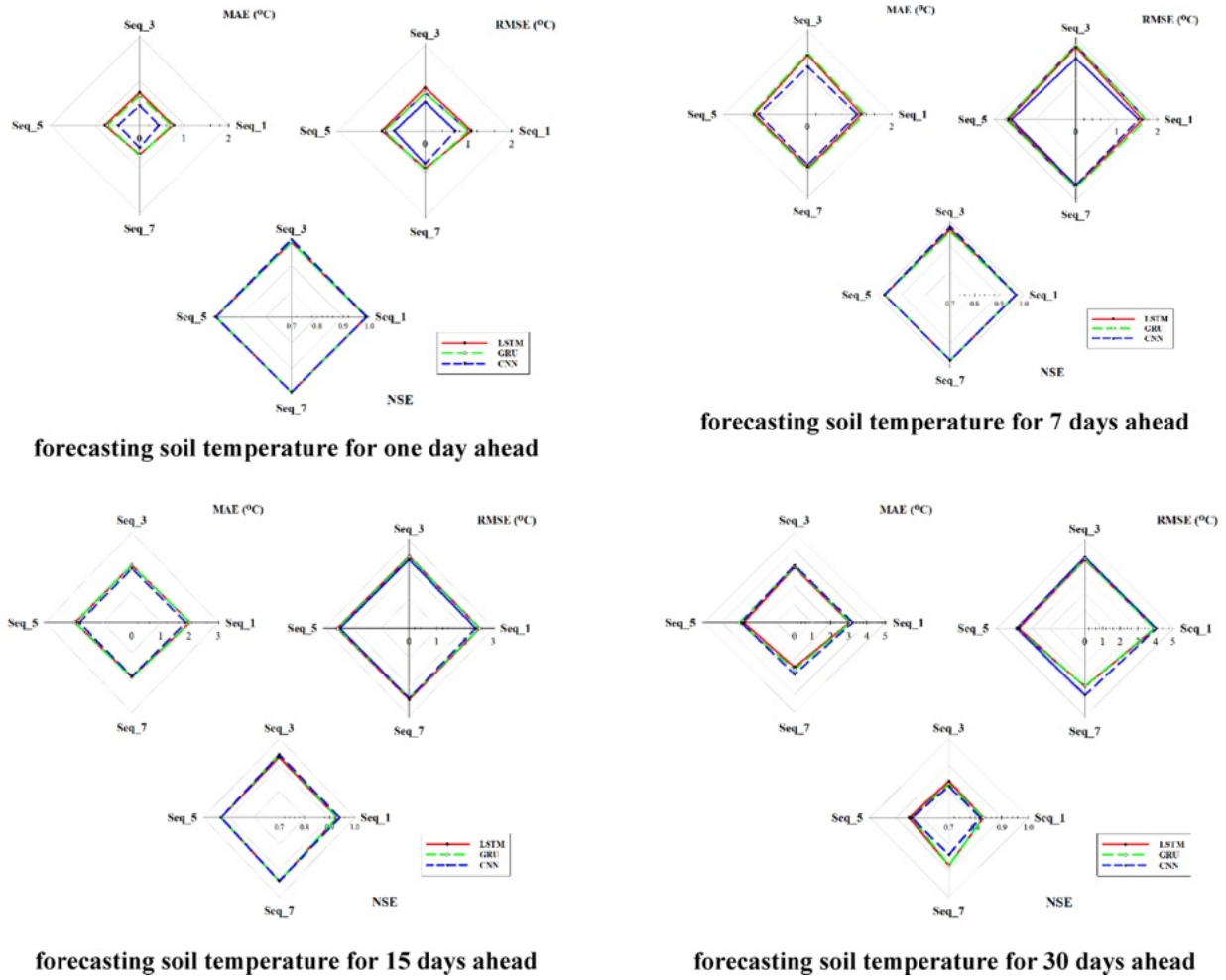

Fig. 11 The performance of deep learning models in forecasting soil t for forecast horizons of 1, 7, 15, and 30 days. Key metrics: RMS

model for short- to medium-term forecasting (1–15 days), while GRU is preferred for simulating long-term trends (30 days). (3) Similar to the trends observed with the best models at the highest depth, uncertainty and error increase as the forecast horizon extends. This confirms that CNN models with a 3-day sequence consistently outperform LSTM and GRU models, with NSE values exceeding 97%, RMSE below 0.6 °C, and MAE under 0.4 °C. The performance of the best models across different forecast horizons is depicted in Fig. 12.

3.1.4 Performance of deep learning models at a depth of 50 cm

The simulation results of deep learning models for forecasting daily soil temperature at a depth of 50 cm (RMSE: 0.5 °C, MAE: 0.4 °C) with input sequence lengths of 1, 3, 5, and 7 days to predict soil temperatures for 1, 7, 15, and 30 days ahead are shown in Fig. 13. Unlike other depths, using a 7-day sequence with the CNN model yielded the highest accuracy for 1-day predictions, achieving an NSE of 92%. For the 7-day forecast, the model with a 3-day sequence (RMSE: 0.9 °C, MAE: 1.09 °C) emerged as the most reliable, demonstrating its reliability. As shown in this Figure, models with a 7-day sequence consistently outperformed those with shorter sequences when forecasting 7, 15, and 30 days ahead. The CNN model with a 7-day sequence delivered the best performance for 7-day forecasts, with RMSE values of 0.8 °C, 1.1 °C, and 1.2 °C for 7, 15, and 30 days, respectively. Similarly, the LSTM model with a 7-day sequence achieved RMSE values of 0.8 °C, 1.1 °C, and 1.2 °C for the same forecast horizons. The GRU model with a 7-day sequence also performed well, with RMSE values of 0.8 °C, 1.1 °C, and 1.2 °C for 7, 15, and 30 days, respectively. These results highlight the importance of using longer input sequences for accurate multi-day soil temperature predictions at greater depths.

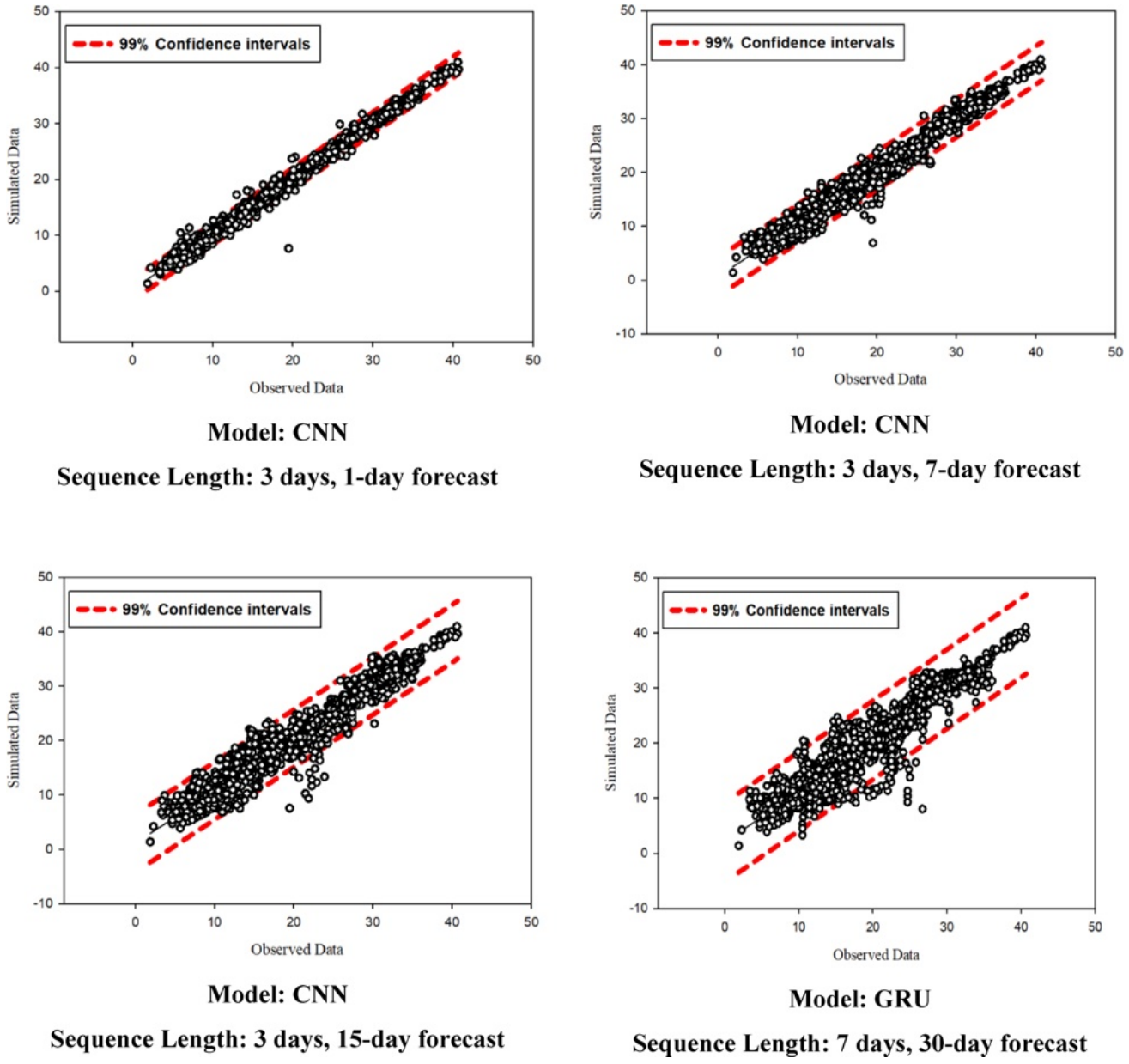

Fig. 12 99% Confdence interval of soil temperature forecasting at a observed values falling predominantly within the predicted confd

the GRU model with a 7-day sequence is the most reliable (NSE: 0.98, RMSE: 3.98 °C, MAE: 2.44 °C). The performance of these best models across different depths and forecast horizons is summarized in Table 8 and illustrated in Fig. 14.

3.1.5 Comparing the performance of the best deep learning models across different depths and forecast horizons

The performance of the best deep learning models (CNN, GRU, and LSTM) in forecasting soil temperature at different depths and forecast horizons is compared using NSE, RMSE, and MAE metrics. As forecast horizons increase from 1 to 30 days, the accuracy of all models decreases, as indicated by higher RMSE and MAE values and lower NSE values. This trend is consistent across all depths, with the most significant performance drop occurring between the 7-day and 15-day forecasts. For example, the GRU model's RMSE increases from 3.98 °C for the 7-day forecast to 5.81 °C for the 15-day forecast, and its NSE decreases from 0.98 to 0.95. This decline is attributed to the increasing complexity and uncertainty in predicting soil temperature over longer periods, where factors like weather variability, soil properties, and interactions between environmental variables have a more pronounced effect. Despite this decrease in accuracy, the models maintain relatively high performance even for the 30-day forecast, with NSE values above 0.90 and RMSE values around 6 °C, indicating that they can still provide reasonably reliable predictions over a month-long horizon. This increased c

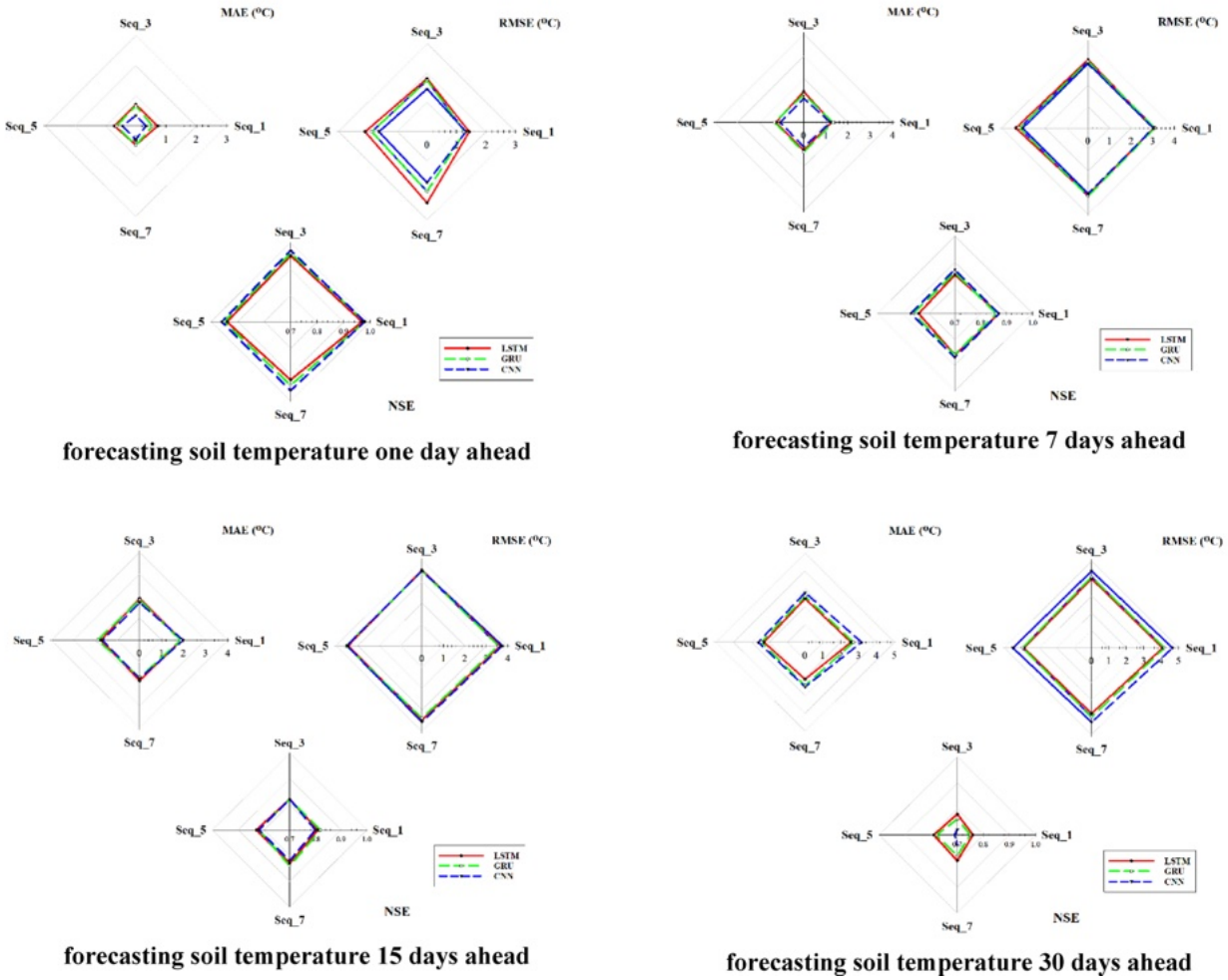

Fig. 13 The performance of deep learning models in forecasting soil t for forecast horizons of 1, 7, 15, and 30 days. Key metrics: RMS

to higher prediction errors. Moadhedl rpeedrufcoerdmannoclee la lpseor fvoarmiae over time. soil depth. For short- to med

Forecasting accuracy also CNdNe cwi ithle saas3 -sdoaiyl sleeqputehn-cie cre For example, at a 5 cm defpotrhms CoNtNhearchmioedveelss aarc rNSS a fl 99% for a 1-day forecast, shpuattitahlisfedartcuprsest of r8o7m% daatt aa 5r depth for the same forecas7t-dlaayr iszeoqru.enTchei sleanlgitghnsi switthhe ings from Kisi et al. (2015) and Na2v0j1i6ewfhoarecast(s (30 days), except a noted that deep learning mpeodeflosmpesfoigmtwyorlsetaetr.grTehait depths. However, th2020)cowh(2021a,b), whfoluindetGRaUl.ef(ective found that BiLSTM performesdoiwlortsetmpaetra5tucmedeepathuaensdabtes at 100 cm depth, a discrepsanadcylowiekredyepdtuhes t(5 dcmfeanend1e0 model architectures and cltoimanteidciumonteirmiofnosr.ecAadsdist,iownha shorter sequence lengths (f1o–r3ecdaasytss). aArte 3sufcomiednetptfho,r CsN term forecasts, while longteor mseedqiuuemm-cteesrm(5f–7rdeadayss)t)sarbur efective for long-term fobeectasetrs,foarslboingh-ltiegmtefodrlaycasla al. (2020) and 2021)e,twahlo.d(emon-stopatiendalthfeoefsehcort- to me-dium- tiveness of models like EEpMeDr-fCoNrNmsanGdRUEEMoDr-C3o0n-vdBadyifnorec daily soil temperature predTihcetidoinf.erence in performance for soil temperature forecas

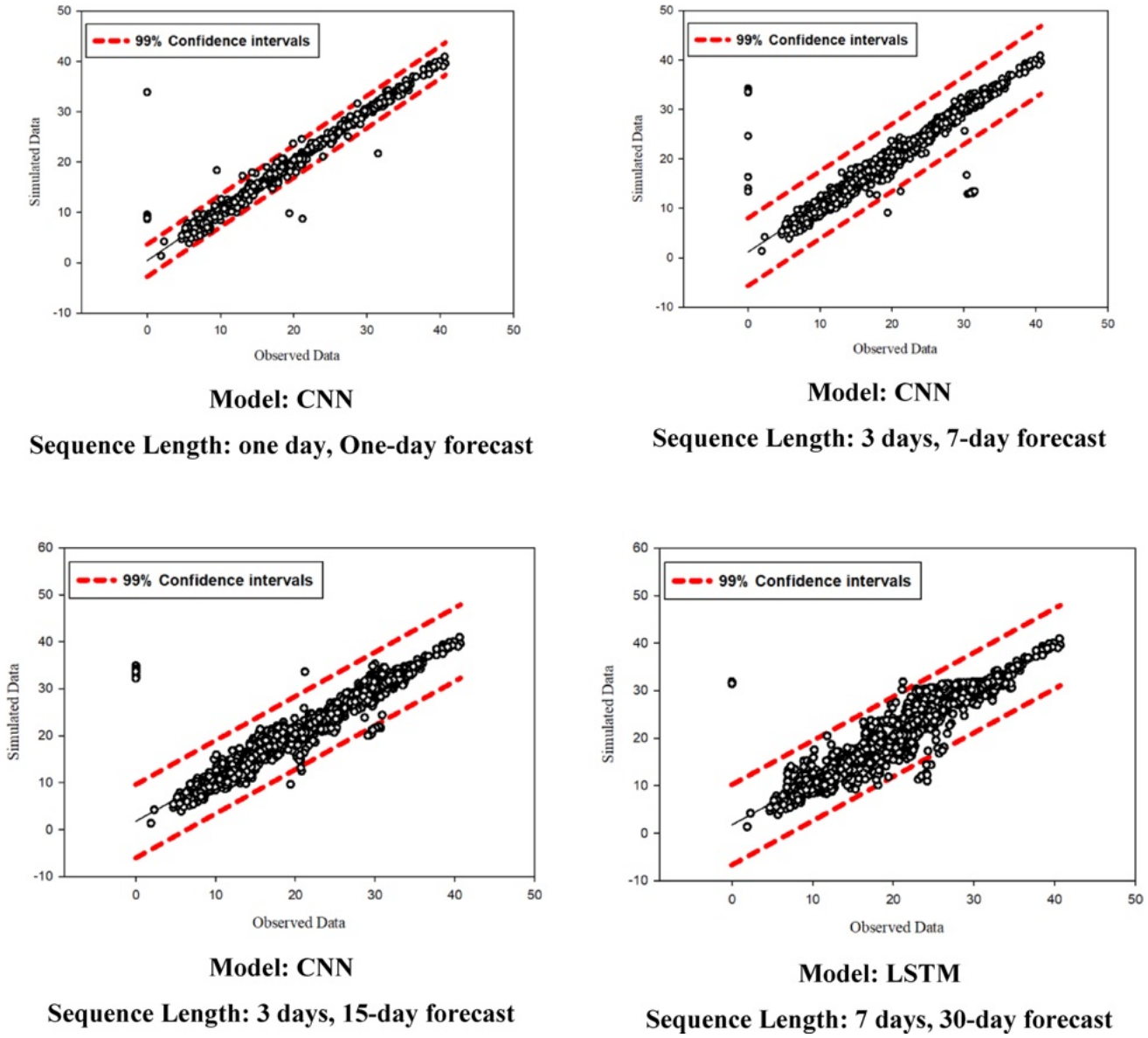

Fig. 14 99% Confidence interval of soil temperature forecasting at a observed values falling predominantly within the predicted confd

horizons can be attributed to the inherent limitations of each model architecture. While CNNs are effective at capturing local temporal patterns (Liang et al. 2021; Tao et al. 2022), their performance tends to degrade over longer forecast horizons due to their limited ability to model long-term dependencies (Wang et al. 2023). For short-term forecasting (e.g., 1 to 5 days), soil temperature trends are often influenced by recent environmental conditions, which exhibit strong temporal correlations (Zhang et al. 2023). In this context, GRU and LSTM models can efficiently extract these patterns, leading to high accuracy (Wang et al. 2023). However, as the forecast horizon extends, the models struggle to retain or forget information appropriately, making it challenging to maintain long-term dependencies, which limits their ability to process fixed-length input sequences (Bai et al. 2018). Extended forecasting periods (Yang et al. 2025) reported that CNN-based models are more effective for short-term predictions, while Attention-based LSTM and GRU models can handle longer sequences, making them more suitable for forecasting solar and wind power.

Table 3 The performance of the best deep learning models (CNN, GRU, a 10 cm, 30 cm, and 50 cm) and forecast horizons (1, 7, 15, and 30

Soil depth (cm)	Forecast horizon (day)	Top models	Sequence length (day)	Error metrics		
				MAE (°C)	RMSE (°C)	NSE
5	1	CNN	3	0.77	1.11	0.99
	7	CNN	3	1.6	2.19	0.96
	15	CNN	3	2.29	2.96	0.92
	30	GRU	7	3.3	4.26	0.85
10	1	CNN	3	0.67	0.95	0.99
	7	CNN	3	1.55	2.07	0.96
	15	CNN	3	2.23	2.9	0.92
	30	GRU	7	3.01	3.96	0.86
30	1	CNN	3	0.67	0.43	0.99
	7	CNN	3	1.14	1.51	0.98
	15	CNN	3	1.79	2.34	0.94
	30	GRU	7	2.59	3.29	0.88
50	1	CNN	1	1.2	3.02	0.87
	7	CNN	3	1.09	2.99	0.87
	15	CNN	3	1.7	3.5	0.82
	30	LSTM	7	2.08	3.78	0.8

CNNs, which are better suited for capturing complex, non-linear patterns.

In the domain of soil temperature forecasting, a variety of studies have explored different approaches (e.g., Nanda 2021) examined the spatiotemporal variability of soil temperature in the Lesser Himalaya region. They applied several machine learning techniques for estimating soil temperature in scarce areas. Their results indicated that the extreme gradient boosting algorithm (XGBoost) outperformed other methods, including artificial neural networks (ANN), multivariate adaptive regression splines (MARS), multiple linear regression (MLR), and support vector machine (SVM). Meanwhile, Li et al. (2022) developed a Long Short-Term Memory (LSTM)-based model that can simultaneously forecast vectors from multiple predictors across different depths. By incorporating attention mechanisms, they enhanced the model's ability to predict soil moisture and temperature. Similarly, Fan et al. (2022) developed an LSTM model for Soil Temperature Prediction (LSTM-STP). This model outperformed several traditional machine learning algorithms, such as Random Forest (RF), Support Vector Regression (SVR), Elastic-Net (ENET), standard LSTM, and Attention-based LSTM (A-LSTM), in most tested scenarios. These findings highlight the potential of advanced machine learning and deep learning techniques in improving soil temperature predictions.

Imanian et al. (2024) conducted a comparative analysis of various deep learning models, including LSTM, CNN, and MLP, for predicting soil temperatures across Canada, from Dawson to Resolute. The study found that traditional models struggled with accurately capturing the complex, non-linear relationships in data, particularly in extremely cold regions. The CNN model, which focuses on local patterns, showed superior performance, with

3.2 Forecasting soil temperature at different depths and lags using vine copulas

The results of the vine copula sequence for all scenarios with 4 lags and 30 and 50 cm. Similarly, the tree sequences for the temperature studied lags (4 and 7) were evaluated using Gaussian, Clayton, and Log-Likelihood criteria, and these results are presented in Appendix A. Subsequent analysis across different depths revealed that both R- and C-vine copulas performed comparably in modeling the dependencies among the parameters. Figure 15 illustrates the concentrated isoquant and log-likelihood values, highlighting the temperature effectiveness in capturing the complex dependencies within different lags.

By examining the tree sequence of different copulas, including C, D, R, as well as Gaussian models, the vine copulas at 5 cm depth were identified as the best sequences regarding the dependence of the studied variables in 4 dimensions. These findings are crucial for understanding the dependencies among the variables.

The results of the analysis, which focused on depths of 10 and 30 cm, R-vine sequences, as presented in Table 5, revealed that the Tehden-Cho-Tawan sequence was chosen as the ultimate selection for the internal copulas of the pair-copula construction. Regarding the internal copulas of the best sequences, results showed that in the four-dimensional vine copulas, internal copulas of Frank and rotated Gumbel 270° were frequently observed as the best copulas in the studied depths of 5 cm and 10 cm. Rotated Gumbel copula (270 degrees) was selected as the best copulas in the studied depths of 5 cm and 30 cm.

In order to fully cover the remaining seven months in this study, it was tried (RMSE) and obtained a vine-

3.2.1 Forecasting soil temperature at different lags using vine copulas at 5 cm depth

Table 4 Selected vine copulas (F: Frank, G270: Rotated Gumbel 270, R: Rotated Gumbel 90, C90: Rotated Clayton copula (90 degrees), N: Gaussian copula)

	Tree	Edge	Internal copula	AIC	Log likelihood	Best tree sequence
5 cm	1	3, 2	G270	−41,050	20,531	R-vine
		1, 3	G270			
		4, 1	F			
	2	1, 2 | 3	F			
		4, 3 | 1	F			
	3	4, 2 | 1, 3	F			
10 cm	1	3, 2	G270	−39,545	19,778	R-vine
		1, 3	G270			
		4, 1	F			
	2	1, 2 | 3	F			
		4, 3 | 1	F			
	3	4, 2 | 1, 3	F			
30 cm	1	3, 2	G270	−38,153	19,083	R-vine
		1, 3	G270			
		4, 1	F			
	2	1, 2 | 3	F			
		4, 3 | 1	G270			
	3	4, 2 | 1, 3	C			
50 cm	1	1, 2	F	−32,876	16,444	C-vine
		1, 3	G270			
		4, 1	F			
	2	3, 2 | 1	C90			
		4, 3 | 1	F			
	3	4, 2 | 3, 1	N			

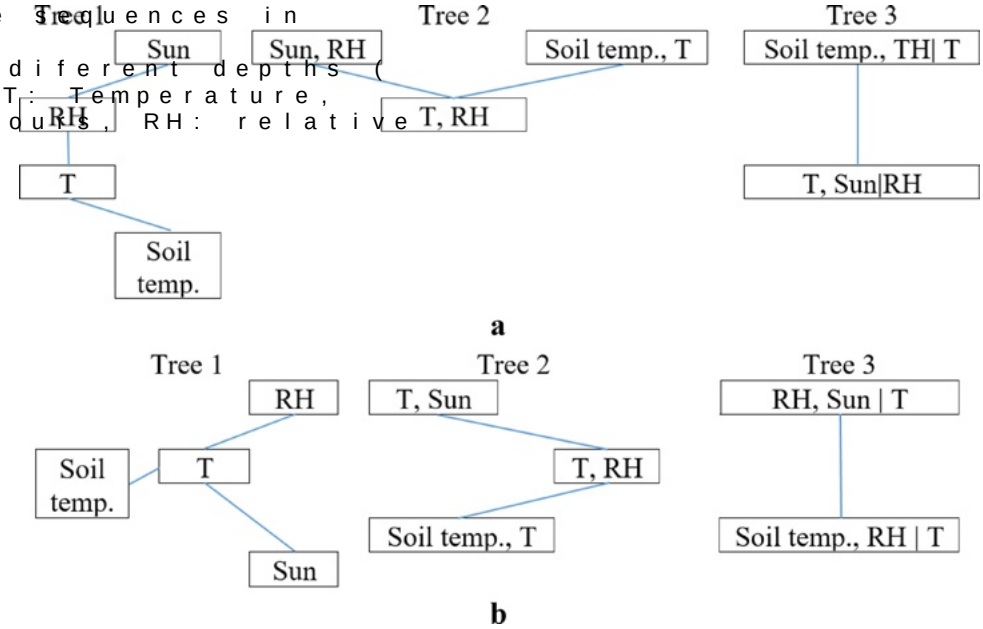

Fig. 15 R and C tree sequences in the forecasting of soil temperature values at different depths (R-vine, C-vine), T: Temperature, Sun: Sunshine hours, RH: relative humidity

Table 5 The results of forecasting soil temperature values at a depth of 5 cm with different lags and forecasting windows using the vine copula approach and R-vine copula-based approach

Simulation	Sim_Lag1	Sim_Lag3	Sim_Lag5	Sim_Lag7
RMSE (°C)				
1-day	0.33	0.19	1.53	2.70
7-day	1.25	2.20	1.90	2.66
15-day	1.37	2.03	1.56	2.96
30-day	1.21	1.92	1.63	2.49
MAE (°C)				
1-day	0.33	0.19	1.53	2.70
7-day	0.87	1.73	1.48	2.21
15-day	1.14	1.53	1.28	2.46
30-day	0.96	1.52	1.29	2.01
NSE				
1-day	0.99	0.99	0.87	0.90
7-day	0.96	0.87	0.90	0.85
15-day	0.97	0.91	0.95	0.77
30-day	0.98	0.95	0.96	0.90

the soil temperature forecasting results in all the 7, 15, and 30-day forecast windows are better than the 1-day forecast period. Also, according to the results, the increase of the forecasting temperature period, the error values are increasing in general and the efficiency coefficient of the model is also decreasing.

Apart from the one-day forecast periods, one-day lag and 3-day lag are the best lags in forecasting soil temperature values at a depth of 5 cm, with a performance according to NSE statistics and the RMSE statistic in forecasting the soil temperature at a depth of 5 cm using the vine copula approach and a 15-day forecast period of RMSE 1.96 °C, which is about average. This forecast period related to the 30-day forecast with one-day lag, which shows error rate of 1.21 °C (RMSE).

To validate the simulation confidence interval for the forecast values at a depth of 5 cm, using 30-day forecast period, as shown in the figure, for lags of 1 and 3, prediction confidence interval while the other falls below the 7-day lag, only one instance in the forecasting of soil temperature the results of predicting soil temperature at different lags using vine copulas at 10 cm depth of 5 cm, the best performance was observed in lag 3.

The sensitivity of the three-day lag can be attributed to the delayed response of soil temperature to changes in air temperature. Soil temperature may not immediately reflect changes in air temperature due to its thermal inertia. The three-day lag may capture the time it takes for changes in air temperature to influence soil temperature at a depth of 5 cm. This delayed response can be influenced by various factors, including soil moisture content, soil composition, and heat transfer processes within the soil.

3.2.2 Forecasting soil temperature at different lags using R-vine copulas

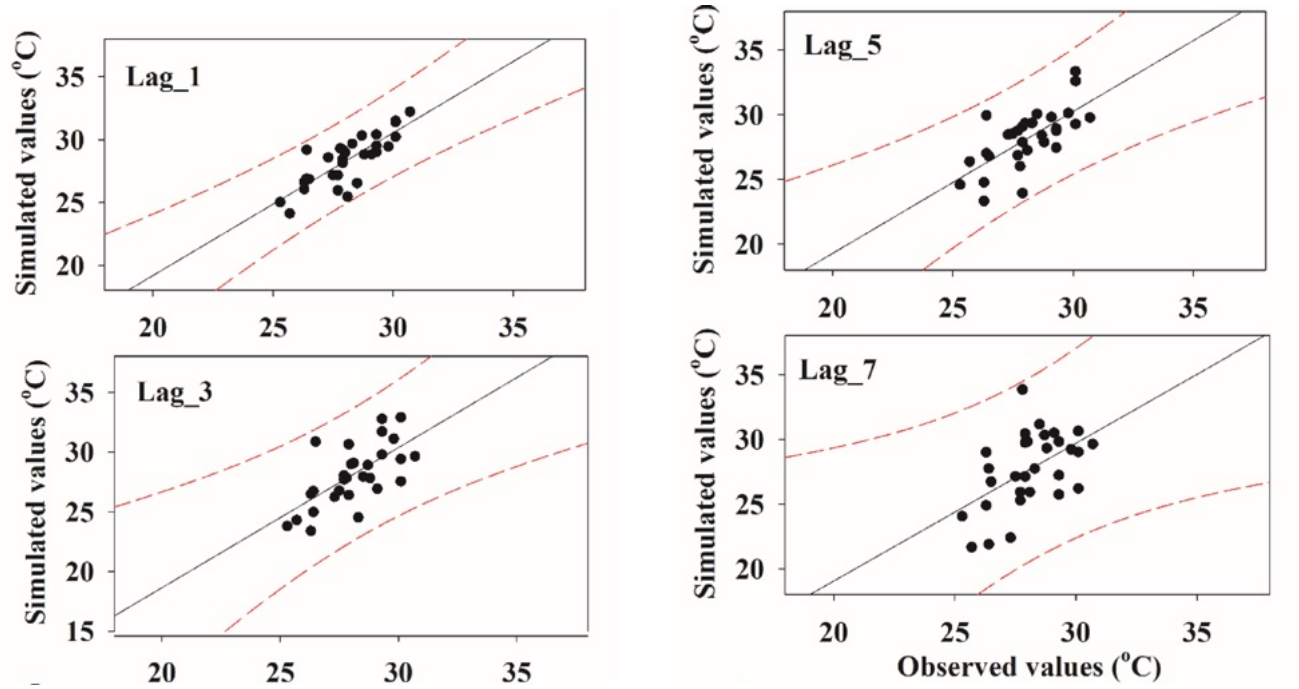

Fig. 16 99% confdence interval of soil temperature forecasting at a d approach

Table 6 The results of forecasting daily soil temperature values of 10 cm with diferent lags and forecasting windows using the v based approach and R-vine copula-based sequential model

Simulation	Sim_Lag_1	Sim_Lag_3	Sim_Lag_5	Sim_Lag_7
RMSE (°C)				
1-day	0.68	1.81	0.71	2.25
7-day	0.93	0.93	0.99	3.97
15-day	0.98	1.16	1.94	3.71
30-day	1.13	1.18	2.21	3.34
MAE (°C)				
1-day	0.68	1.81	0.71	2.25
7-day	0.79	0.79	0.80	3.29
15-day	0.90	0.99	1.29	3.01
30-day	1.01	1.00	1.70	2.71
NSE				
1-day	0.99	0.95	0.98	0.93
7-day	0.98	0.98	0.98	0.81
15-day	0.98	0.98	0.92	0.80
30-day	0.98	0.98	0.93	0.87

The results of forecasting daily soil temperature values at a depth of 10 cm using meteorological data based on the vine copula-based sequential model showed that, as the forecasting period increased, the amount of error value (RMSE) also increases. However, unlike the pattern of forecasting soil temperature at a depth of 5 cm, in a five-day lag, the forecasting soil temperature in the four-day lag and seven-day lag are not better than three-day lag. Also, the RMSE, MAE, and NSE indicators showed that the model decreases with the increasing in general and the model is also decreasing, whi in the forecasting pattern of cm. In the one-day forecastin of 10 cm, the best performanc five-day lags and one-day fore In addition to one-day foreca periods, a one-day lag and a ered the most efective lags f at a depth of 10 cm. This pat ing behavior observed at a de Nash–Sutcliffe Efficiency (NSE) coefficient for the three-day l while for the one-day lag, it Furthermore, the results in

of the long-term mean soil temperature, showing high performance of 81%. Considering the three-day forecast period, the scenario-based examining the forecasting of the soil temperature, the seven-day period and in the seven-day, the two have been forecasting confidence interval, also called the of the upper bound and the lower bound, which indicates an uncertainty caused by the

In the case of a five-day, the estimation was observed in the dependencies between at a depth of 10 cm using the sequence. No overestimation for the other lags that were. Overall, apart from one case in the seven-day lag, the 95% confidence interval. The RMSE, tory performance of the soil temperature over a 30-day period, highlighting its reliability.

3.2.3 Forecasting soil temperature at different lags using vine copulas at 30 cm depth

According to the forecasting, at the depths of 5 and 10 cm, and the RMSE, and MAE, 30 and the three-day, five-day, and

casting of soil temperature, difference was observed, efficiency of the model at 30 cm, examining the error rate (RMSE,

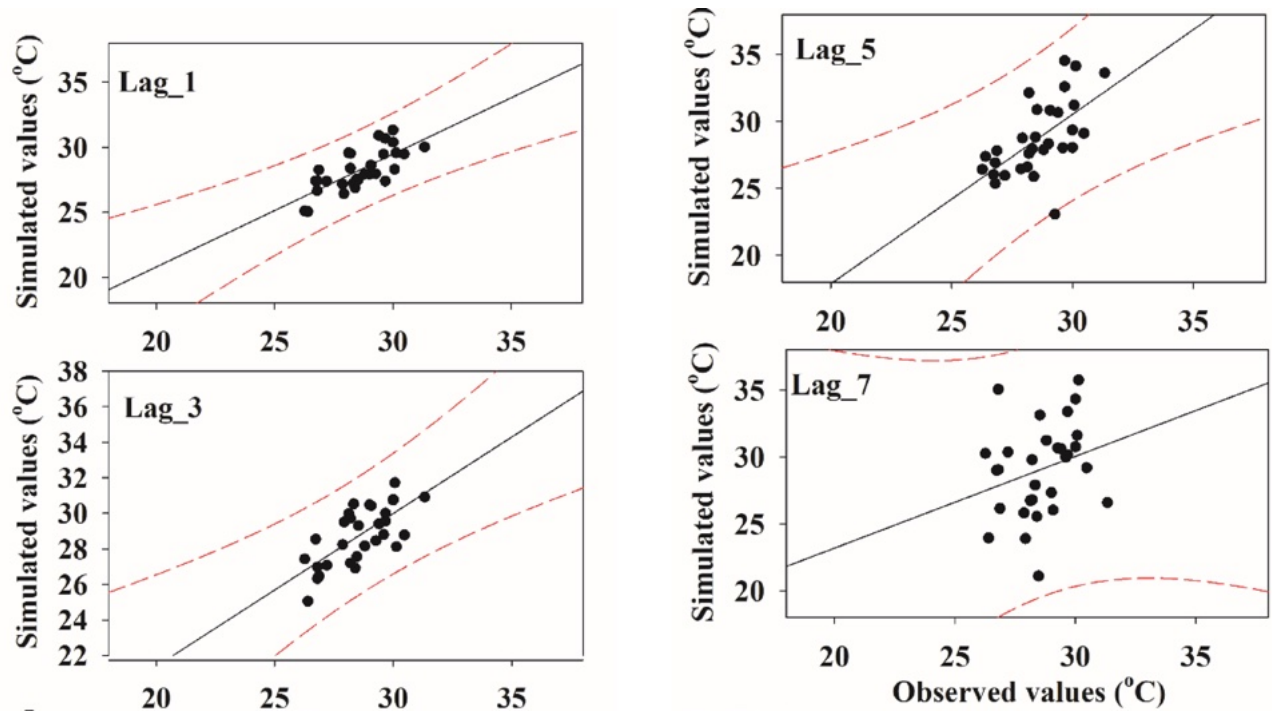

Fig. 17. 99% confidence interval of soil temperature forecasting at a d
approach

Table 7 The results of forecasting soil temperature values at a depth of 30 cm with different lags and forecasting windows using the vine copula-based approach and R-vine copula-based sequence

Simulation	Sim_Lag_1	Sim_Lag_3	Sim_Lag_5	Sim_Lag_7
RMSE (°C)				
1-day	1.65	1.05	1.65	7.19
7-day	1.47	1.56	2.38	3.29
15-day	1.21	1.58	2.69	3.29
30-day	1.38	1.61	2.51	3.06
MAE (°C)				
1-day	1.65	1.05	1.65	7.19
7-day	1.26	1.31	2.25	2.40
15-day	0.96	1.41	2.45	2.74
30-day	1.04	1.33	2.09	2.49
NSE				
1-day	0.90	0.98	0.89	0.73
7-day	0.95	0.96	0.85	0.82
15-day	0.98	0.96	0.88	0.86
30-day	0.97	0.96	0.91	0.89

An examination of the 30-day at a depth of 30 cm, it was observed predicted values across different confidence interval. This underscores and reliability of the studies bly, while maximum overestimations occurred in two cases during temperatures at depths of 5 and not observed at the 30 cm depth ness and consistency of the varying soil temperature at greater

3.2.4 Forecasting soil temperature at different lags using vine copulas at 50 cm depth

The forecasting results at a depth of 50 cm on a daily scale with five-day, and seven-day, consistent RMSE statistic, the lowest values, root mean square error, and mean absolute error of soil temperature at a depth of 30 cm, approximately 1.21 °C, as indicated by the forecast, is about 1.21 °C. This value is relatively low compared to the 15-day forecast period. At a depth of 50 cm, the results

Based on the Nash–Sutcliffe efficiency (NSE) it is evident that for forecasting soil temperatures at a depth of 50 cm, and lags of one-day and three-day, with the R tree sequence accounting for 95% in forecasting soil temperature at a depth of 50 cm, as the best sequence based on

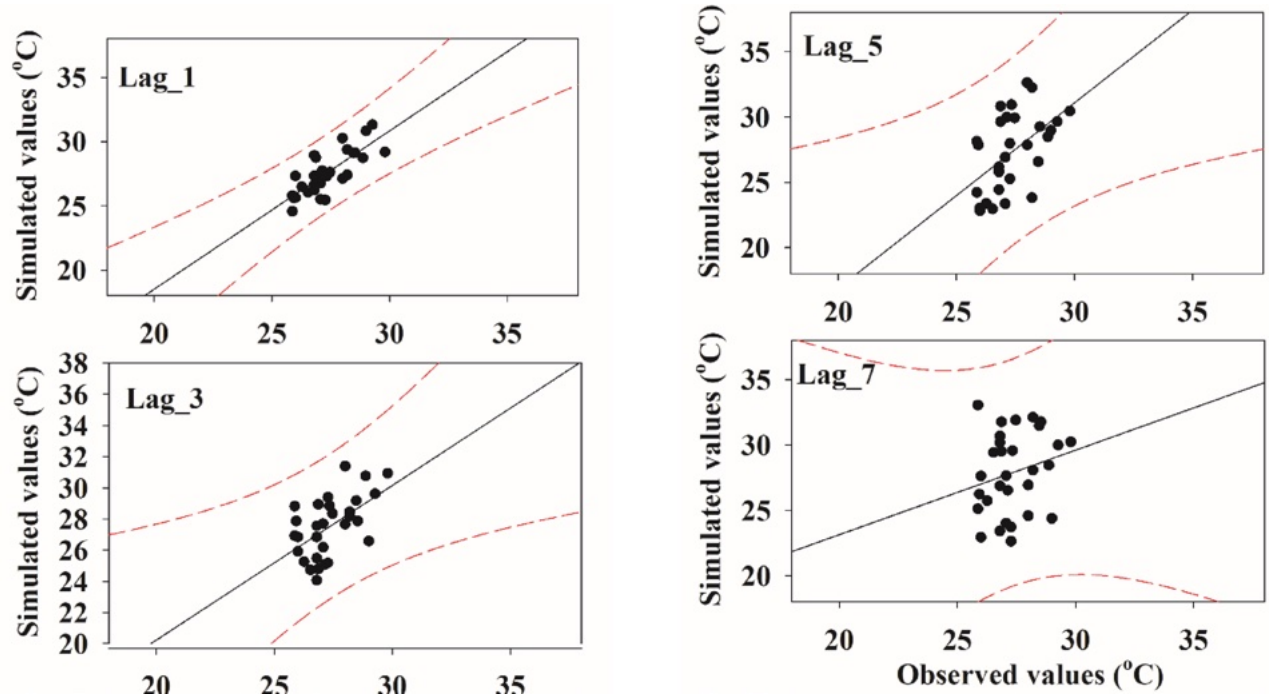

Fig. 18 99% confidence interval of soil temperature forecasting at a d approach

Table 8 The results of forecasting soil temperature values at a depth of 50 cm with different lags and forecasting windows using the vine-based approach and C-vine copula-based sequence

Simulation	Sim_Lag1	Sim_Lag3	Sim_Lag5	Sim_Lag7
RMSE (°C)				
1-day	0.73	2.13	4.12	1.00
7-day	0.97	1.91	3.02	3.24
15-day	0.99	1.84	2.72	3.30
30-day	1.05	2.13	2.66	2.96
MAE (°C)				
1-day	0.73	2.13	4.12	1.00
7-day	0.78	1.72	2.54	2.86
15-day	0.79	1.56	2.43	2.87
30-day	0.83	1.78	2.33	2.47
NSE				
1-day	0.99	0.88	0.89	0.98
7-day	0.99	0.93	0.88	0.88
15-day	0.99	0.95	0.91	0.85
30-day	0.99	0.95	0.92	0.91

selected tree sequence. According to the Table 8, in the depth of 50 cm, rotational copulas in the studied lags. At a depth of 50 cm based on C-vine copula sequence, the Frank copula in all lags, but in the second and third lags, the Clayton's copula (90 and 270 degrees rotational) is the best copula.

According to the Table 8, in the different depths of 50 cm using the temperature values in the vine-based approach and the C-vine copula-based sequence that with increasing lags of 7 days from the root mean square error (RMSE) increased approximately increasing the lags from 1 to 7 day, the amount of RMSE-value increase according to the Table 8, it can be seen that in the forecasting period, the RMSE and MAE values compared to the depths of 5, 10 and 30 cm only in lags 1 day and 3 day, which is in forecasting soil temperature across different depths and lags five-day and seven-day, in forecast horizons lation period, the amount of RMSE and MAE has a slight decrease. In forecasting of the soil temperature at a depth of 50 cm, the best performance was observed at one-day lag. According to Table 9 also be seen that in three-day lag, the values of 7, 15 and 30 days, the RMSE and MAE approach and the C-vine-copula-based sequence with the lowest RMSE and MAE temperature forecasting at a depth of 50 cm is 95%, which is better than other lags in forecasting the soil temperature in the vine-based approach and the C-vine-copula-based sequence at seven-day lag, which is the root mean square error (RMSE) forecast periods of 7 to 30 days. The forecast with a 3 time window was drawn and presented in the form of Fig. 7 for a more detailed review and interval of the forecasting presented in Fig. 10. It can be seen that lag, where an underestimated cast data, in other cases (or lags) no points were observed interval. Figure 11 shows the acceptable vine-based approach and the C in the 30-day forecasting of cm. Dai (2020) also confirmed the the vine-based model and the in the simulation and that the forecasted values are values.

By examining the forecasting ferent time windows, the results window, the best performance with lags of one-day, three-day, 50 cm, the best performance was observed in the vine-based approach and the C-vine copula-based sequence that with increasing lags of 7 days from the root mean square error (RMSE) increased approximately increasing the lags from 1 to 7 day, the amount of RMSE-value increase according to the Table 8, it can be seen that in the forecasting period, the RMSE and MAE values compared to the depths of 5, 10 and 30 cm only in lags 1 day and 3 day, which is in forecasting soil temperature across different depths and lags five-day and seven-day, in forecast horizons lation period, the amount of RMSE and MAE has a slight decrease. In forecasting of the soil temperature at a depth of 50 cm, the best performance was observed at one-day lag. According to Table 9 also be seen that in three-day lag, the values of 7, 15 and 30 days, the RMSE and MAE approach and the C-vine-copula-based sequence with the lowest RMSE and MAE temperature forecasting at a depth of 50 cm is 95%, which is better than other lags in forecasting the soil temperature in the vine-based approach and the C-vine-copula-based sequence at seven-day lag, which is the root mean square error (RMSE) forecast periods of 7 to 30 days. The forecast with a 3 time window was drawn and presented in the form of Fig.

3.2.5 Comparing the performance of the best vine copulas in forecasting soil temperature across different depths and forecast horizons

Table 9 summarizes the soil temperature forecasting results

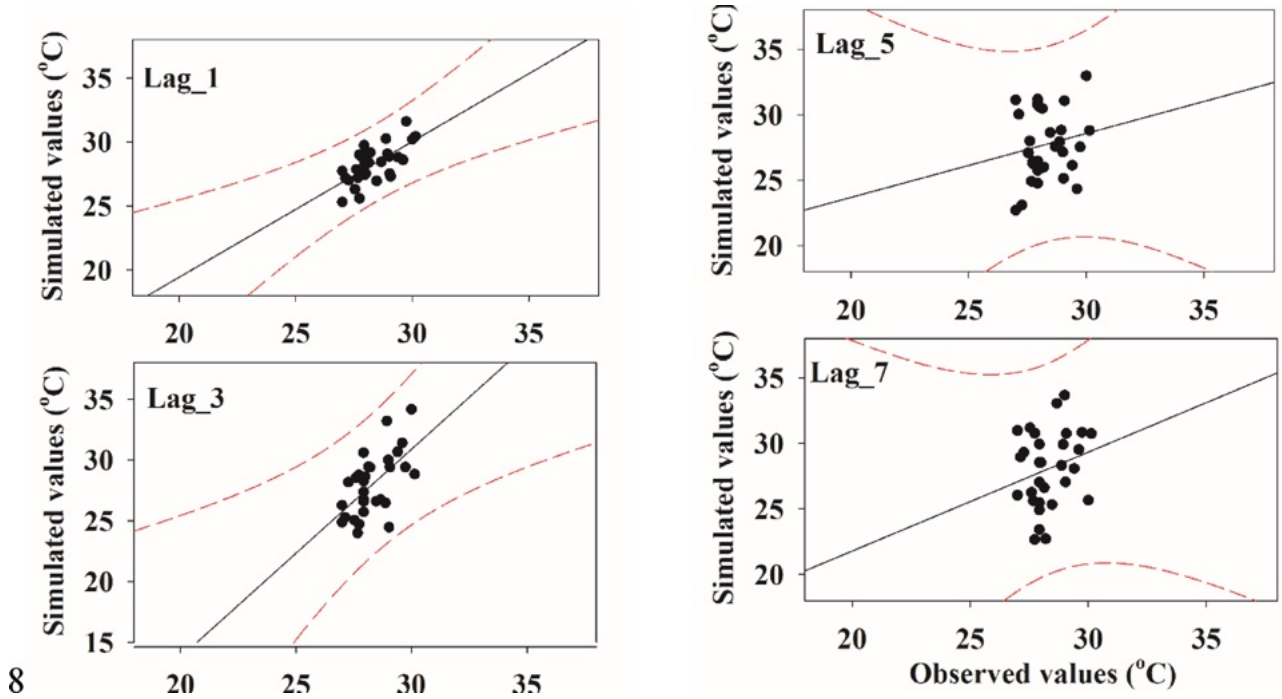

Fig. 19 99% confdence interval of soil temperature forecasting at a d approach

Table 9 The performance of the best vine-based model in soil temperature forecasting across different depths (5 cm, 10 cm, 30 cm, and cm) and forecast horizons (1, 7, 15, and 30 days)

Soil depth (cm)	Forecast horizons (days)	Lag	Metrics		
			MAE (°C)	RMSE (°C)	NSE
5	1	3	0.19	0.19	0.99
	7	1	0.87	1.25	0.96
	15	1	1.14	1.37	0.97
	30	1	0.96	1.21	0.98
10	1	5	0.71	0.71	0.98
	7	3	0.79	0.93	0.98
	15	1	0.9	0.98	0.98
	30	1	1.01	1.13	0.98
30	1	3	1.05	1.05	0.98
	7	1	1.26	1.47	0.98
	15	1	0.96	1.21	0.98
	30	1	1.04	1.38	0.97
50	1	1	0.73	0.73	0.99
	7	1	0.78	0.97	0.99
	15	1	0.79	0.99	0.99
	30	1	0.83	1.05	0.99

3.2.6 Comparing the performance of the best vine copulas and deep learning models in forecasting soil temperature

Vine copulas and deep learning models are powerful approaches to modeling data, each with its own strengths and weaknesses. Vine copulas, as mentioned earlier, decompose multivariate dependencies into a series of bivariate copulas, which are well-suited for modeling low- to moderate-dimensional data in hydrology, and climate science with high-dimensional data and for copula selection and parameter estimation. Deep learning, based on artificial neural networks, is designed to handle large, high-dimensional datasets and can automatically learn complex patterns and relationships in the data, making it particularly effective for tasks like time series forecasting. It offers unparalleled scalability and flexibility, but it can be computationally intensive, requires large amounts of training data, and lacks interpretability and may require significant computational resources and data. The choice between these two approaches depends on the problem context, data characteristics, and the trade-off between model interpretability versus predictive performance.

Comparative Performance of Vine-Based Models (Vine-GRU/LSTM) and Vine-Based Models in Soil Temperature Forecasting Across-Multiple Depths and Forecast Horizons is presented in Table 9. The comparison indicates that the performance of the best vine copula model for one day ahead is deemed acceptable. Additionally, a comparison of various deep learning models at the best performance at depth of 5 cm for one day ahead is achieved. Vine copulas and deep learning models are powerful approaches to modeling data, each with its own strengths and weaknesses. Vine copulas, as mentioned earlier, decompose multivariate dependencies into a series of bivariate copulas, which are well-suited for modeling low- to moderate-dimensional data in hydrology, and climate science with high-dimensional data and for copula selection and parameter estimation. Deep learning, based on artificial neural networks, is designed to handle large, high-dimensional datasets and can automatically learn complex patterns and relationships in the data, making it particularly effective for tasks like time series forecasting. It offers unparalleled scalability and flexibility, but it can be computationally intensive, requires large amounts of training data, and lacks interpretability and may require significant computational resources and data. The choice between these two approaches depends on the problem context, data characteristics, and the trade-off between model interpretability versus predictive performance.

Table 10 Comparative performance of deep learning (CNN/GRU/LSTM) and V depths and time horizons

Soil depth	Horizon (days)	Deep learning models	Sequence length	Error metrics (days)		
				MAE (°C)	RMSE (°C)	NSE
5	1	CNN	3	0.77	1.11	0.99
	7	CNN	3	1.6	2.19	0.96
	15	CNN	3	2.29	2.96	0.92
	30	GRU	7	3.3	4.26	0.85
10	1	CNN	3	0.67	0.95	0.99
	7	CNN	3	1.55	2.07	0.96
	15	CNN	3	2.23	2.9	0.92
	30	GRU	7	3.01	3.96	0.86
30	1	CNN	3	0.67	0.43	0.99
	7	CNN	3	1.14	1.51	0.98
	15	CNN	3	1.79	2.34	0.94
	30	GRU	7	2.59	3.29	0.88
50	1	CNN	1	1.2	3.02	0.87
	7	CNN	3	1.09	2.99	0.87
	15	CNN	3	1.7	3.5	0.82
	30	LSTM	7	2.08	3.78	0.8
		Vine-based models	**Lag time (days)**			
5	1	Vine-Based	3	0.19	0.19	0.99
	7	Vine-Based	1	0.87	1.25	0.96
	15	Vine-Based	1	1.14	1.37	0.97
	30	Vine-Based	1	0.96	1.21	0.98
10	1	Vine-Based	5	0.71	0.71	0.98
	7	Vine-Based	3	0.79	0.93	0.98
	15	Vine-Based	1	0.9	0.98	0.98
	30	Vine-Based	1	1.01	1.13	0.98
30	1	Vine-Based	3	1.05	1.05	0.98
	7	Vine-Based	1	1.26	1.47	0.98
	15	Vine-Based	1	0.96	1.21	0.98
	30	Vine-Based	1	1.04	1.38	0.97
50	1	Vine-Based	1	0.73	0.73	0.99
	7	Vine-Based	1	0.78	0.97	0.99
	15	Vine-Based	1	0.79	0.99	0.99
	30	Vine-Based	1	0.83	1.05	0.99

superior performance in forecasting soil temperature com- 50 cm, GRU with sequence len pared to LSTM models. Howeavreo,ngamheeng alearstiung 2.0 Thedose. lSB for one-day-ahead forecastliinggh,tetdhet hbaets tCNoNvse rhaalvle præmfaodrw at depths of 5, 10, and 50 heminr isonavsoucfiaoneal waitthe cV based approach, while at twheeig3h0t-sc,mechepatnhc,intghetCeNNefmcoideen performs best. The error rdaitcetioofn.thTeheVyineev-ablausætdeaptphreo a forecasting soil temperatufroer ofnoeredcaays taihmeadfnaatncdeapthtsimo 10, and 50 cm is approxima5t0e0l,yu8s3i%n,g 2b5e%t,h aunndc4r3e%i tlioowrearl respectively, compared to Tthheeirtofpn-dpienrgfsorimnidnigcadteeedptlheaatr model. In the one-day-ahead gsroeisls ibempteasrakts,ureasforirtecaasnt, deep learning model demonrsetqruaitreisngsuepxetreinosrivpeerfiocsrtmoarnicea compared to the Vine-based atpipmeoaedhcieonty aand tshtera-30 hcfm depth. rent networks, often ou-tperfo

In the 7-day soil temp-eraetumrefodre2c0a1s8t).sEfaurigtehtetrahks.up(pmpta son results show that theemCpNiNrimcaadeslstudeymoonrstreaqtuesnctehem best performance at depthsutoifon5a,l 1a0r,ch3l0ecmuriesseacchnieovf by CNN model with sequencaebilleintgythco3mpaanrdedattotheculdepetmt

However, in general, the Vine-based graph approach demonstrates the best performance as all top-ranked models lags of 1 and 3. The error comparisons, which the Vine-based in forecasting 7-day soil temperatures at the depths of 5, 10, and 50 cm is approximately 24%, 25%, and 16%, and 6–8% lower, respectively, compared to other soft computing models. As a result, the Vine-based approach outperforms deep learning models in 7-day, 15-day, and 30-day forecasting. In the forecasting of soil temperature 7 days ahead, the LSTM model with a sequence length of 3 achieved the best performance for the Vine-based approach and 30 cm among deep learning models, with three LSTM layers. However, at a depth of 5 cm, the GRU model with a sequence length of 7 was more effective in forecasting temperatures. Various machine learning techniques (XGBoost, CatBoost, LSTM, ANN, Ridge Regression, and Elastic Net) were examined by Ma et al. (2024) to predict soil temperatures at different depths. Their results indicated that the XGBoost model provided the most accurate predictions for soil temperature levels at a depth of 10 cm. Various methods (e.g., multivariate deep learning models) have been employed to address irregularities and gaps in time series data (Fang et al. 2018). In contrast to traditional deep learning models, GRU-based models, such as the Vine-based model for managing irregularities and gaps in data, demonstrating its superiority over other models (Che et al. 2018). Additionally, the Vine-based approach outperforms other machine learning approaches in medical data series (Tachmazidis et al. 2024). Despite the sampling interval (Yang et al. 2018), the Vine-based approach in forecasting of long- and short-term temperatures (LSTM), making it a powerful tool for forecasting, showing that LSTM (GRU), models can learn hierarchical temporal features, a task in which Vine-based models outperform other models. Additionally, the Vine-based approach outperforms other models in the overall best performance, as indicated by the lowest root mean square error rate (RMSE) of the Vine-based approach in forecasting 15-day soil temperatures at depths of 5, 10, and 30 cm is approximately 54%, 65%, and 48% lower, respectively, compared to the other models. Consequently, for 15-day and 30-day forecasting, the Vine-based approach outperforms other models (Ehteram et al. 2023; Ehteram 2024) and can be used to achieve optimal performance.

In the forecasting of 30-day soil temperature, the comparison results show that the best performance at depths of 5, 10, and 30 cm among deep learning models is achieved by the GRU model with a sequence length of 7, while the LSTM model with a sequence length of 7 performs best. For 30-day

4 Conclusions

temperature at multiple deepthsiv(e5 ,ov1e0 ra l3 0 , o faenrdin-5g0 lomw in Khorramabad, Iran. Theabrielsiutlyt sachriogshsliv gahrty itnhge ddeipstthi strengths and limitations keefy eadovhanatpagreoadch .thDeeVpinlee-aran models, particularly CNNs,sedqumeonrcsetrsattreudctsutreomagndcatphaebial ties in capturing temporawipthatthensevlactedCNNsgiancahliedvi optimal performance at shparledowetrivdeepatchsur(a5c,y .1 0T,hiasndst8 cm) using a sequence lengtfhoro fre3s,eawrhcihleersGRAUnsdepxrcaecltlietdi deeper layers (50 cm) withmoadesleiqnugenfcreamleewngrthksobfa7-edHowe the Vine copula-based appmroeancths,coonosnitsrtiebnutliyngoutpeafdovramm deep learning models acrossdemloisntg,deapgtrhiscualntdurpe,redaindtire horizons, achieving signifecxapnltolrye lhoywberrideraropprroacteess (RM and demonstrating superiorboitrhtemreptrheotdasbitloitfyuratnhderroebnuhsatn By leveraging lags of 1 ainndte3r,prtehteabVinley-basRedgaapdpingact efectively modeled complexi,ctnionng-lsionielartemdpeepreantduenceiveasl,u making it a reliable choiccelifmoartipcrolbiambitlaitsitoincsfocerceaduasset uncertainty quantifcation.ginal distribution of the inp

The fndings underscore thhye ccoomnspildeemreinntgartyhenattrueree sefque these approaches: deep learning models are well-suite for scalable, data-driven tasks requiring high-dimensi data processing, while Vi**Appendix A: Tree sequence in different lag** cenaric where interpretability, -p**and different depth** c forecasting, and dling smaller datasets are critical. For soil tempera forecasting, the Vine-baseSdeeapa,bd,aeclh3demerged as the mo

Table 11 The results of evaluation trheen ttrdeeepthsesquweintche tohfreve-ndeaycolpauglas

	Tree	Edge	Internal Copula	AIC	Log Likelih	Best tree sequ
5 cm	1	3 , 2	G 2 7 0	−3 5 , 5 2 2	1 7 , 7 6 7	R - v i n e
		1 , 3	G 2 7 0			
		4 , 1	F			
	2	1 , 2 | 3	F			
		4 , 3 | 1	F			
	3	4 , 2 | 1 , 3	F			
10 cm	1	3 , 2	G 2 7 0	−3 5 , 1 5 8	1 7 , 5 8 5	R - v i n e
		1 , 3	G 2 7 0			
		4 , 1	F			
	2	1 , 2 | 3	F			
		4 , 3 | 1	F			
	3	4 , 2 | 1 , 3	F			
3 0 c m	1	3 , 2	G 2 7 0	−3 9 , 8 5 9	1 9 , 9 3 6	R - v i n e
		1 , 3	F			
		4 , 1	F			
	2	1 , 2 | 3	F			
		4 , 3 | 1	G 2 7 0			
	3	4 , 2 | 1 , 3	C			
50 cm	1	1 , 2	F	−3 4 , 7 7 5	1 7 , 3 9 3	C-vine
		1 , 3	F			
		4 , 1	G 2 7 0			
	2	3 , 2 | 1	C 2 7 0			
		4 , 3 | 1	C90			
	3	4 , 2 | 3 , 1	C 2 7 0			

Table 12 The results of examining three nttr edee pstehqsu ewnicteh offv ev-idnaey cloapgu las a

	Tree	Edge	Internal Copula	AIC	Log Likelih	Boeosdt tree sequ
5 cm	1	3 , 2	G 2 7 0	−3 3 , 9 7 8	1 6 , 9 9 5	R - v i n e
		1 , 3	G 2 7 0			
		4 , 1	F			
	2	1 , 2 | 3	F			
		4 , 3 | 1	F			
	3	4 , 2 | 1 , 3	F			
10 cm	1	3 , 2	G 2 7 0	−3 3 , 6 3 1	1 6 , 8 2 1	R - v i n e
		1 , 3	G 2 7 0			
		4 , 1	F			
	2	1 , 2 | 3	F			
		4 , 3 | 1	F			
	3	4 , 2 | 1 , 3	F			
3 0cm	1	3 , 4	G 2 7 0	−3 8 , 8 0 9	1 9 , 4 1 1	R - v i n e
		2 , 3	F			
		4 , 1	F			
	2	1 , 2 | 3	F			
		4 , 3 | 1	G 2 7 0			
	3	4 , 2 | 1 , 3	F			
50 cm	1	1 , 2	F	−3 5 , 1 7 0	1 7 , 5 9 2	C-vine
		1 , 3	F			
		4 , 1	t			
	2	3 , 2 | 1	C 2 7 0			
		4 , 3 | 1	C90			
	3	4 , 2 | 3 , 1	C 2 7 0			

Table 13 The results of examining three nttr edee pstehqsu ewnicteh osfe vveinn-ed acyo plualga s a

	Tree	Edge	Internal copula	AIC	Log likelih	Boeosdt tree sequ
5 cm	1	3 , 2	G 2 7 0	−3 2 , 7 3 8	1 6 , 3 7 6	R - v i n e
		1 , 3	G 2 7 0			
		4 , 1	F			
	2	1 , 2 | 3	F			
		4 , 3 | 1	t			
	3	4 , 2 | 1 , 3	F			
10 cm	1	3 , 2	G 2 7 0	−3 2 , 3 8 0	1 9 , 1 9 6	R - v i n e
		1 , 3	G 2 7 0			
		4 , 1	F			
	2	1 , 2 | 3	F			
		4 , 3 | 1	F			
	3	4 , 2 | 1 , 3	F			
3 0cm	1	3 , 2	F	−3 1 , 1 4 2	1 8 , 1 8 2	R - v i n e
		1 , 3	G 2 7 0			
		4 , 1	F			
	2	1 , 2 | 3	C90			
		4 , 3 | 1	F			
	3	4 , 2 | 1 , 3	N			
50 cm	1	1 , 2	F	−3 6 , 9 1 3	1 8 , 4 6 3	C-vine
		1 , 3	F			
		4 , 1	F			
	2	3 , 2 | 1	C 2 7 0			
		4 , 3 | 1	C90			
	3	4 , 2 | 3 , 1	C 2 7 0			

Appendix B

Results of additional accuracy evaluation criteria for evaluating the performance of studied models (Willmott's Index, Absolute Percentage Bias and Explained Variance Score)

Note:

- Willmott's Index = perfect model = 1 (range 0 to 1)
- Absolute Percentage Bias = perfect model = 0 (range 0 to ∞, positive values indicate overestimation bias, negative values indicate underestimation bias)
- Explained Variance Score = perfect model = 1 (range 0 to 1, lower values indicate less accurate prediction)

See Tables 14, 15, 16, 17, 18, 19, 20, 21, and 22

Table 14 The results of forecasting the temperature values at a depth of 5 cm with different lags and forecasting window using the vine copula-based approach and R-vine copula-based approach

Lag	Willmott's index	Absolute percentage bias	Explained variance score
30-day			
Sim_Lag_1	0.99	3.42	0.28
Sim_Lag_3	0.99	5.43	−0.92
Sim_Lag_5	0.99	4.61	−0.37
Sim_Lag_7	0.98	7.23	−2.10
15-day			
Sim_Lag_1	0.98	4.02	0.21
Sim_Lag_3	0.97	5.65	−2.09
Sim_Lag_5	0.98	4.73	−0.83
Sim_Lag_7	0.96	9.07	−5.85
7-day			
Sim_Lag_1	0.91	3.33	−7.61
Sim_Lag_3	0.87	6.60	−26.20
Sim_Lag_5	0.87	5.64	−19.76
Sim_Lag_7	0.87	8.47	−34.00

Table 15 The results of forecasting the temperature values at a depth of 10 cm with different lags and forecasting window using the vine copula-based approach and R-vine copula-based approach

Lag	Willmott's index	Absolute percentage bias	Explained variance score
30-day			
Sim_Lag_1	0.99	3.52	0.31
Sim_Lag_3	0.99	3.51	0.21
Sim_Lag_5	0.98	5.88	−1.87
Sim_Lag_7	0.98	9.55	−5.42
15-day			
Sim_Lag_1	0.98	3.24	0.16
Sim_Lag_3	0.98	3.57	−0.13
Sim_Lag_5	0.96	4.56	−2.54
Sim_Lag_7	0.93	10.90	−12.39
7-day			
Sim_Lag_1	0.92	2.98	−9.58
Sim_Lag_3	0.86	2.96	−10.88
Sim_Lag_5	0.86	2.98	−9.19
Sim_Lag_7	0.82	12.33	−122.71

Table 16 The results of forecasting the temperature values at a depth of 30 cm with different lags and forecasting window using the vine copula-based approach and R-vine copula-based approach

Lag	Willmott's index	Absolute percentage bias	Explained variance score
30-day			
Sim_Lag_1	0.99	3.77	0.37
Sim_Lag_3	0.99	4.90	−1.30
Sim_Lag_5	0.98	7.73	−4.67
Sim_Lag_7	0.98	9.44	−7.64
15-day			
Sim_Lag_1	0.97	3.62	−1.64
Sim_Lag_3	0.96	5.30	−6.33
Sim_Lag_5	0.95	9.20	−22.12
Sim_Lag_7	0.94	10.61	−33.36
7-day			
Sim_Lag_1	0.88	4.84	−18.94
Sim_Lag_3	0.81	5.04	−20.28
Sim_Lag_5	0.86	8.65	−46.08
Sim_Lag_7	0.75	9.50	−118.42

Table 17 The results of forecasting the temperature values at a depth of 50 cm with different lags and forecasting window using the vine copula-based approach and R-vine copula-based approach

Lag	Willmott's index	Absolute percentage bias	Explained variance score
30-day			
Sim_Lag_1	0.99	2.95	0.46
Sim_Lag_3	0.98	6.28	−4.91
Sim_Lag_5	0.97	8.25	−8.00
Sim_Lag_7	0.97	8.77	−10.55
15-day			
Sim_Lag_1	0.96	2.84	0.02
Sim_Lag_3	0.94	5.64	−21.96
Sim_Lag_5	0.93	8.83	−52.84
Sim_Lag_7	0.92	10.38	−76.20
7-day			
Sim_Lag_1	0.89	2.87	−11.13
Sim_Lag_3	0.88	6.30	−27.91
Sim_Lag_5	0.83	9.35	−115.10
Sim_Lag_7	0.83	10.45	−130.53

Table 18 The results of forecasting 5 cm soil moisture using different deep learning models (CNN, GRU, and LSTM)

Lag	Willmott's index			Absolute percentage			Explained variance score		
Model	CNN	GRU	LSTM	CNN	GRU	LSTM	CNN	GRU	LSTM
30-day									
Sim_Seq_01	1.941	0.937	0.940	28.269	29.575	28.940	0.797	0.780	0.785
Sim_Seq_03	0.950	0.955	0.946	24.162	24.697	25.588	0.825	0.830	0.810
Sim_Seq_05	5.953	0.956	0.954	23.516	22.940	24.348	0.834	0.846	0.828
Sim_Seq_07	0.952	0.958	0.953	24.857	22.363	23.552	0.832	0.854	0.828
15-day									
Model	CNN	GRU	LSTM	CNN	GRU	LSTM	CNN	GRU	LSTM
Sim_Seq_01	1.977	0.974	0.975	17.988	18.240	18.455	0.919	0.904	0.904
Sim_Seq_03	0.980	0.976	0.978	17.002	17.464	17.279	0.926	0.918	0.918
Sim_Seq_05	0.980	0.979	0.979	17.391	17.557	16.344	0.923	0.923	0.922
Sim_Seq_07	0.979	0.979	0.980	17.364	16.385	16.378	0.923	0.928	0.926
7-day									
Model	CNN	GRU	LSTM	CNN	GRU	LSTM	CNN	GRU	LSTM
Sim_Seq_01	1.987	0.987	0.986	13.782	13.753	14.796	0.954	0.949	0.946
Sim_Seq_03	0.989	0.986	0.988	13.209	13.856	12.957	0.958	0.953	0.956
Sim_Seq_05	0.989	0.988	0.989	13.023	13.268	12.801	0.959	0.956	0.957
Sim_Seq_07	0.989	0.989	0.988	13.277	12.761	12.842	0.959	0.959	0.958

Table 19 The results of forecasting 10 cm soil moisture using different deep learning models (CNN, GRU, and LSTM)

Lag	Willmott's index			Absolute percentage			Explained variance score		
Model	CNN	GRU	LSTM	CNN	GRU	LSTM	CNN	GRU	LSTM
30-day									
Sim_Seq_01	1.943	0.943	0.941	26.945	26.668	27.795	0.806	0.797	0.789
Sim_Seq_03	0.945	0.950	0.952	25.739	24.175	23.258	0.817	0.820	0.818
Sim_Seq_05	0.943	0.958	0.955	29.563	23.149	22.136	0.809	0.846	0.823
Sim_Seq_07	0.947	0.961	0.960	27.733	22.121	21.258	0.822	0.860	0.849
15-day									
Model	CNN	GRU	LSTM	CNN	GRU	LSTM	CNN	GRU	LSTM
Sim_Seq_01	1.978	0.977	0.977	17.328	17.278	16.916	0.920	0.912	0.913
Sim_Seq_03	0.979	0.980	0.979	17.138	15.890	16.242	0.921	0.924	0.921
Sim_Seq_05	0.979	0.980	0.979	17.593	16.514	15.762	0.921	0.928	0.922
Sim_Seq_07	0.978	0.981	0.979	18.227	15.355	15.250	0.917	0.932	0.927
7-day									
Model	CNN	GRU	LSTM	CNN	GRU	LSTM	CNN	GRU	LSTM
Sim_Seq_01	0.990	0.988	0.988	12.381	12.635	12.836	0.960	0.953	0.951
Sim_Seq_03	0.990	0.989	0.989	12.438	12.109	12.073	0.961	0.958	0.956
Sim_Seq_05	0.988	0.989	0.990	13.777	12.035	12.003	0.955	0.959	0.960
Sim_Seq_07	0.990	0.990	0.989	12.079	11.798	11.698	0.960	0.961	0.959

Table 20 The results of forecasting 30 cm with the inter-annual values using deep learning models (CNN, GRU, and LSTM)

Lag	Willmott's index			Absolute percentage			Explained variance score		
Model	CNN	GRU	LSTM	CNN	GRU	LSTM	CNN	GRU	LSTM
30-day									
Sim_Seq_01	0.955	0.956	0.955	22.814	22.395	22.293	0.833	0.838	0.834
Sim_Seq_03	0.954	0.958	0.958	23.337	22.558	21.902	0.834	0.846	0.846
Sim_Seq_05	0.958	0.959	0.961	22.012	22.130	20.231	0.850	0.847	0.853
Sim_Seq_07	0.960	0.969	0.970	21.034	20.170	17.972	0.850	0.889	0.883
15-day									
Model	CNN	GRU	LSTM	CNN	GRU	LSTM	CNN	GRU	LSTM
Sim_Seq_01	0.984	0.983	0.982	13.778	14.159	14.817	0.942	0.933	0.933
Sim_Seq_03	0.985	0.983	0.984	13.878	14.420	13.606	0.943	0.939	0.936
Sim_Seq_05	0.982	0.982	0.982	15.063	14.493	14.143	0.938	0.934	0.931
Sim_Seq_07	0.984	0.984	0.983	14.509	14.028	13.901	0.939	0.941	0.937
7-day									
Model	CNN	GRU	LSTM	CNN	GRU	LSTM	CNN	GRU	LSTM
Sim_Seq_01	0.993	0.992	0.993	9.359	9.384	9.543	0.977	0.969	0.971
Sim_Seq_03	0.994	0.991	0.991	9.188	10.318	9.731	0.976	0.969	0.966
Sim_Seq_05	0.993	0.992	0.992	9.697	10.052	10.083	0.973	0.970	0.972
Sim_Seq_07	0.993	0.992	0.993	9.939	10.319	9.579	0.972	0.971	0.972

Appendix C

The mathematical formulations governing our deep learning architecture are presented below the CNN model

The CNN architecture for regression tasks comprises the following key components.

The input usually consists of sequential time series data, expressed as:

$$Input = [X_1, X_2, X_3, \ldots, X_t]$$

where T denotes the duration, X_t represents the value at the operation utilizes a 1D filter to extract features. The result is:

$$output\ (i) = \sum_{n=0}^{N-1} Kernel\ (n) . input\ (i+n) \quad (9)$$

Kernel (n) 1D sequence, *input* 1D *(i+n)* represents the input sequence at position $i+n$, and *output (i)* indicates the resulting of the convolution at position i.

Table 21 The results of forecasting 50 cm with the inter-annual values using deep learning models (CNN, GRU, and LSTM)

Lag	Willmott's index			Absolute percentage			Explained variance score		
Model	CNN	GRU	LSTM	CNN	GRU	LSTM	CNN	GRU	LSTM
30-day									
Sim_Seq_01	0.929	0.938	0.940	18.135	16.376	15.716	0.717	0.762	0.768
Sim_Seq_03	0.935	0.943	0.944	15.942	14.965	14.408	0.742	0.774	0.781
Sim_Seq_05	0.931	0.944	0.947	15.571	14.724	13.979	0.726	0.790	0.795
Sim_Seq_07	0.935	0.946	0.948	15.169	14.235	12.538	0.753	0.786	0.800
15-day									
Sim_Seq_01	0.951	0.954	0.953	11.694	11.239	10.882	0.807	0.821	0.818
Sim_Seq_03	0.956	0.956	0.957	10.629	11.200	11.020	0.832	0.833	0.825
Sim_Seq_05	0.957	0.957	0.958	10.495	11.111	10.506	0.831	0.831	0.834
Sim_Seq_07	0.954	0.958	0.957	11.062	10.460	11.103	0.825	0.840	0.837
7-day									
Sim_Seq_01	0.968	0.967	0.967	7.235	7.764	7.333	0.870	0.860	0.866
Sim_Seq_03	0.968	0.965	0.965	6.948	7.739	8.096	0.871	0.864	0.857
Sim_Seq_05	0.966	0.964	0.960	7.132	7.729	7.527	0.869	0.856	0.840
Sim_Seq_07	0.967	0.965	0.965	7.155	7.878	7.457	0.869	0.858	0.862

convolution, an activation function—such as ReLU, Sigmoid, Tanh, or Linear functions—is applied to introduce non-linearity. For regression tasks, ReLU (Rectified Linear Unit) is commonly applied (Wang et al. 2019; Oyebode et al. 2021; Imanian et al. 2022).

$$\mathrm{ReLU}(x) = \max(0, x);\ Sigmoid(x) = \frac{1}{1+e^{-x}};\quad \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}};\ Linear(x) = x \tag{10}$$

Pooling decreases the dimensionality of the feature maps. In the case of 1D CNNs, max pooling is frequently employed, as described below:

$$MaxPooling(i) = \max(output\ (i+m))) \tag{11}$$

Here, m denotes the size of the pooling window. Following multiple convolution and pooling layers, the output is flattened and passed through one or more fully connected (dense) layers. The output can be expressed as (Oyebode et al. 2019; Oyebode and Adeyemo 2021; Wang et al. 2021):

$$y = W.x + b \tag{12}$$

Here, x represents the flattened input from the previous layers, W is the weight matrix, b is the bias vector, and y is the output of the fully connected layer. The loss function measures the discrepancy between the forecasted and actual values. The weights are adjusted during training using gradient descent. The weight update formula is as follows:

$$W_{update} = W_{old} - \left(Learning\ Rate \frac{\partial Loss}{\partial W}\right) \tag{13}$$

where $\frac{\partial Loss}{\partial W}$ is gradient of the loss with respect to the weights. Learning rate determines the speed at which the weights are updated during the training process as described by Shi et al. (2021):

The LSTM model

The equations of the LSTM model are as follows:

$$f_t = \sigma(W_f.(h_{t-1}, x_t) + b_f) \tag{14}$$

Here, σ represents the sigmoid activation function, producing values between 0 and 1, W_f is the weight matrix for the forget gate, h_{t-1} is the hidden state from the previous time step, x_t is the input at the current time step, and b_f is the bias term for the forget gate.

$$i_t = \sigma(W_f.(h_{t-1}, x_t) + b_i) \tag{15}$$

$$\tilde{C}_t = \tanh(W_c.(h_{t-1}, x_t) + b_c) \tag{16}$$

Here, W_i represents input gate weight matrix, b_i is the bias for the input gate, and $\tilde{C}_t$ denotes Candidate memory information to potentially store:

$$C_t = f_t.C_{t-1} + i_t.\tilde{C}_t \tag{17}$$

where C_t shows update memory cell, C_{t-1} is previous memory cell state, and $\tilde{C}_t$ is candidate memory.

$$o_t = \sigma(W_f.(h_{t-1}, x_t) + b_o) \tag{18}$$

$$h_t = o_t.\tanh(C_t) \tag{19}$$

where o_t is the Output gate which should be output and scales the range; and h_t is hidden state (the LSTM each time step, which is passed and employed for making predictions).

The LSTM model

The equations of the GRU model are as follows:

At each time step t, the GRU executes operations:

1. The reset gate determines the extent to which the previous hidden state h_{t-1} should be forgotten, and the update gate determines the extent to which the new candidate hidden state $\tilde{h}_t$ should be used to update the hidden state h_t.

$$r_t = \sigma(W_r.(h_{t-1}, x_t) + b_r) \tag{20}$$

where σ is sigmoid activation function, W_r is the weight matrix for the reset gate, h_{t-1} is the previous hidden state, x_t is the input at the current time step, and b_r is the bias for the reset gate.

The update gate Z_t which determines how much of the previous hidden state h_{t-1} should be forgotten (when the candidate hidden state $\tilde{h}_t$ is used to update the hidden state h_t).

$$Z_t = \sigma(W_z.(h_{t-1}, x_t) + b_Z) \tag{21}$$

where W_z is weight matrix b_z is for the bias term for the update gate

2. Candidate hidden state ($\tilde{h}_t$): This step is a key role in determining the next hidden state in the sequence processing. It computed based on the current input and the previous hidden state.

$$\tilde{h}_t = \mathrm{Tanh}\left(W_h.\left(r_t.h_{t-1}, x_t\right) + b_h\right) \quad (22)$$

where W_h is weight matrix for the candidate hidden state, and b_h is bias term for the candidate hidden.

Hidden state (h_t): which is updated by combining the retained previous hidden state and the new candidate hidden state.

$$h_t = \left(1 - Z_t\right).h_{t-1} + Z_t.\tilde{h}_t \quad (23)$$

Author contributions MS and MNT: Methodology, software, formal analysis, writing—original draft, preparing the revised version. AHH and GDM: conceptualization, validation, writing—review & editing. All authors reviewed the revised manuscript.

Funding No funding was received to assist with the preparation of this manuscript.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Aas K, Nagler T, Jullum M, Løland A (2021) Explaining predictive models using Shapley values and non-parametric vine copulas. Dependence modeling 9(1):154–186. https://doi.org/10.1515/demo-2021-0103

Alizamir M, Kim S, Zounemat-Kermani M, Heddam S, Shiri J, Yu S (2022a) Modeling hydro-meteorological data at different time steps: insights from intelligence model: deep echo state neural network model. Arch Comput Methods Eng 29(8):5051–5084

Alizamir M, Kim S, Zounemat-Kermani M, Heddam S, Shiri J, Yu S (2022b) Modeling hydro-meteorological data at different time steps: insights from intelligence model: deep echo state neural network model. Arch Comput Methods Eng 29(8):5051–5084

Al-Musaylh MS, Al-Dafaie K, Downs NJ, Gazi MY, Hussain A, Deo RC, Parisi AV, Soar J, Igoe DP, Deo RC, Parisi AV, Soar J, Igoe DP, Deo RC, Parisi AV, Soar J (2022) Multi-step solar ultraviolet index prediction using multivariate hybrid convolutional neural networks with long short-term memory for a representative case study in Queensland, Australia. Atmosphere 13(7):1134

Allison SD (2005) Cheaters, diffusion and nutrients constrain decomposition by microbial enzymes in spatially structured environments. Ecol Lett 8(6):626–635. https://doi.org/10.1111/j.1461-0248.2005.00756.x

Azubuike CO, Chikwado CE, Ogbulie JN (2023) Bacterial siderophores: the role in heavy metal detoxification and bioremediation. J Basic Microbiol 63(3–4):282–299. https://doi.org/10.1002/jobm.202200188

Azzouni A, Pujolle G (2018) A long short-term memory recurrent neural network framework for network traffic matrix prediction. arXiv preprint arXiv:1705.05690

Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271

Bashir T, Wang H, Tahir M, Zhang Y (2023) Wind energy modeling for short-term wind power forecasting based on hybrid CNN-MLP models. Renew Energy 239:122–133

Bedford T, Cooke R (2001) Probabilistic risk analysis: foundations and methods. Cambridge University Press

Biazar SM, Shehadeh HA, Ghorbani M (2024) Soil temperature forecasting based on hybrid CNN-MLP models. Renew Energy 239:122–133

Borovykh A, Bohte S, Oosterlee CW (2017) Conditional time series forecasting with convolutional neural networks. arXiv preprint arXiv:1703.04691

Botterud A, Wang J, Bessa RJ, Keko H, Carvalho L, Trindade A, Miranda V (2011) Wind power trading under uncertainty in LMP markets. IEEE Trans Power Syst 27(2):894–903

Brabec M, Konár O, Pelikán E, Malý M (2008) A nonlinear mixed effects model for the prediction of natural mortality. For Ecol Manage 255(3–4):1222–1231

Brownlee J (2018) Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery

Che Z, Purushotham S, Cho K, Sontag D, Liu Y (2018) Recurrent neural networks for multivariate time series with missing values. Sci Rep 8(1):6085

Chen H, Huang JJ, Dash SS, Wei Y, Deng Z (2018) Constructing a random forest model to predict ecosystem respiration from eddy covariance data. Agric For Meteorol 252:11–24

Cheng L, Zhang L, Wang YP, Yan J, Liu S, Wang J, Tang Y, Zhang Z, Xu X, Zheng Z, Yu Q (2017) Parameter sensitivity of CABLE model in simulating water and energy fluxes over a temperate coniferous forest. J Hydrol 551:366–378

Cheng L, Zhang H, Xu Y, Wei Z, Sun G (2015) Beyond the Penman-Monteith equation: the CR-model for daily evapotranspiration. Hydrol Earth Syst Sci 19(11):4413–4427

Cholette ME, Ding Y, Zhou E (2019) On the uncertainty of power conversion efficiency for wind energy systems. IEEE Trans Sustain Energy 5(1):251–258

Chollet F (2017) Deep learning with Python. Manning Publications Co.

Cleveland RB, Cleveland WS, McRae JE, Terpenning I (1990) STL: a seasonal-trend decomposition procedure based on loess. J Off Stat 6(1):3–73

Dai L, Liu J (2020, October) Research on the prediction model and application of LSTM neural network. In: Journal of Physics: Conference Series (Vol. 1584, No. 1, p. 012102). IOP Publishing

Dai M, Ding Z, Zhang P (2016) A fast low-rank Kalman filter algorithm for data assimilation in reservoir simulation. Comput Geosci 20(3):537–549

Dai M, Ding Z, Zhang P (2017) A fast low-rank Kalman filter algorithm for data assimilation in reservoir simulation. Comput Geosci 20(3):537–549

De Michele C, Salvadori G, Passoni G, Vezzoli R (2007) A multivariate model of sea storms using copulas. Coast Eng 54(10):734–751. https://doi.org/10.1016/j.coastaleng.2007.04.002

Ding J, Hu J, Yang S, Zhang J, Li W, Tu Y, Liu Y, Yang Y, Wang Y, Xu Y, Liu Y (2022) Using artificial intelligence to explore the thermal comfort models and the adaptive behavior differences between outdoor and indoor environments. Energy Build 254:111572

Dong YN, Zhu Y, Li J, Xia X, Yan H, Qiao ZM, Igoe DP, Deo RC, Parisi AV, Soar J (2020) Short-term solar irradiance forecasting using MEA-RVM: an ensemble learning method. Measurement 151:107251

Fathalla A, Li J, Ali A (2017) Constructing efficient stochastic models via probabilistic model checking. Int J Softw Eng Knowl Eng 27(08):1231–1255

Fathalla A, Li J, Ali A (2018) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2019) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2020) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2021) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2022) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2023) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2024) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2025) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2026) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2027) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2028) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2029) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2030) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2031) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2032) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2033) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2034) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2035) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2036) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2037) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2038) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2039) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2040) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2041) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2042) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2043) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2044) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2045) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2046) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2047) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2048) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2049) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Fathalla A, Li J, Ali A (2050) Constructing efficient probabilistic model checking formulas using bisimulation. IEEE Access 6:34536–34552

Feng Y, Cui N, Hao W, Gao L, Gong D (2015) Estimation of soil temperature from meteorological data using different machine learning models. Geoderma 338:67–77
Genest C, Favre AC (2007) Everything you always wanted to know about copula modeling but were afraid to ask. J Hydrol Eng 12(4):347–368
Ghimire S, Nguyen-Huy T, AL-Musaylh MS, Deo RC, Casillas-Pérez D, Salcedo-Sanz S (2023) Integrated multi-head self-attention transformer model for electricity demand prediction incorporating local climate variables. Energy and AI 14:100302
Ghimire S, AL-Musaylh MS, Nguyen-Huy T, Casillas-Pérez D, Salcedo-Sanz S, Deo RC, Acharya R (2024) Hybrid deep learning CNN-BiLSTM-MRFO-VMD approach for global solar radiation forecasting: case studies in Australian states. Energy Convers Manag 314:118674
Guo J, Chen C, Wen H, Cai G, Liu Y (2024) A prediction model of coal temperature based on PSO-GRU-based neural network. Stud Therm Eng 53:103813
Hao H, Yu F, Li Q (2020) Soil temperature prediction based on convolutional neural network based on ICEEMDAN. IEEE Access 9:40841–40851
Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
Imanian H, Hiedra Cobo J, Payeur P, Mohammadian A (2022) A comprehensive study of soil temperature predictions for soil depths and extremely hot events. Sustainability 14(13):8065
Imanian H, Mohammadian A, Farhangmehr V, Payeur P, Hiedra Cobo J, Shirkhani H (2024) Comparative analysis of deep learning models for climates. Theor Appl Climatol 155:2571–2587
Jaworski P, Durante F, Hardle WK (2012) Copula theory and its applications, vol 198. Springer
Khashei-Siuki A, Shahidi A, Ramezani Y, Nazeri A (2021) Simulation of potential evapotranspiration using copula. Meteorol Appl 28(5):e2013
Kisi O, Tombul M, Kermani MZ (2015) Modeling soil temperatures at different depths by using three different neural computing techniques. Theor Appl Climatol 121:377–387
Kizito R, Scruggs P, Li X, Devine J, Jaansseden M, Dee K (2018) Comparing short-term memory networks for predicting remaining useful life. In: 2018 IEEE Conference
Lai, G., Chang, W. C., Yang, Y., & Liu, H. (2018) long-and short-term temporal patterns with deep neural networks. In The 41st international ACM SIGIR conference
Lee T, Shin JY, Kim JS, Singh R (2020) reproducing long-term memory of hydrological time series using deep learning model. J Hydrol 587:124954
Li C, Zhang Y, Ren X (2020) Modeling hourly soil temperature using deep BiLSTM neural network. Algorithms 13(7):159
Li Z, Liu F, Yang W, Peng S, Zhou J (2021) A survey of neural networks: analysis, applications, and prospects. IEEE Trans Neural Netw Learn Syst 33:6999–7019
Li Q, Zhu Y, Shangguan W, Wang X (2022) Attention-aware LSTM model for soil moisture and soil temperature prediction. Geoderma 409:115651
Lü TJ, Tang XS, Li DQ, Qi XH (2020) Multiple soil parameters prediction. Geotech 118:103340
Ma L, Pan SB, Dai XG, Song S (2010) forecasting model of working face based on PSO-GRU. J Xi'an Univ Sci Technol 40:363–368
Mampitiya L, Rozumbetov K, Rathnayake N, Welivita Y, Arachchi S, Kantamaneni K, Rathnayake U (2024) Enhancing

Salvadori G, De Michele C, Kottegoda NT, Rosso R (2007) Extremes in nature: an approach using copulas, vol 56. Springer Science & Business Media
Shi H, Wang L, Scherer R, Woźniak M, Zhang P, Wei W (2021) Short-term load forecasting based on adabelief optimized temporal convolutional network and gated recurrent unit hybrid neural network. IEEE Access 9:66965–66981
Sklar A (1959) Fonctions de répartition à n dimensions et leurs marges. Publications De L'institut de Statistique de l'Université de Paris, Paris 8:229–231
Taheri M, Schreiner HK, Mohammadian H, Cobo JH (2023) A review of empirical and machine learning approaches to soil temperature estimation. Int J Remote Sens 44(9):2677–2719
Wadoux AMC (2019) Using deep learning for multivariate mapping of soil with quantified uncertainty. Geoderma 351:59–70
Wang J, Chen X, Zhang F, Chen Y (2021a) Building load forecasting using deep neural network with efficient feature fusion. J Mod Power Syst Clean Energy 9(1):160–169
Wang X, Li W, Li Q (2021b) A new embedded estimation model for soil temperature prediction. Sci Program 2021(1):5881018
Wang Y, Chen S, Hong Y, Hu B, Peng J, Shi Z (2023) A comparison of multiple deep learning methods for predicting soil organic carbon in Southern Xinjiang, China. Comput Electron Agric 212:108067