BİNGÖL UNIVERSITY

Turkish Journal of Nature and Science

e-ISSN: 2149-6366
www.dergipark.gov.tr/tdfd

Research Article

Volume 14, Issue 4, Page 1-14, 2025

Received: 14.03.2025
Accepted: 15.09.2025
Published: 30.12.2025

https://doi.org/10.46810/tdfd.1657401

Wind Speed Prediction in Bingol Region: A Deep Learning and Stacking Ensemble Approach

Serdal POLAT[1*], Nuh ALPASLAN[2]

[1] Bingöl University, Engineering and Architecture Faculty, Electrical and Electronics Department, Bingöl, Türkiye
[2] Bingöl University, Engineering and Architecture Faculty, Computer Department, Bingöl, Türkiye

**Corresponding author: spolat@bingol.edu.tr*

Abstract

This study aims to develop a novel model for wind speed prediction by integrating advanced deep learning techniques with ensemble methods using wind speed data collected from various districts of the Bingol region. The methodology includes rigorous data preprocessing, time-based feature engineering, STL decomposition, and standardization – all mathematically modeled. A hybrid deep learning model comprising Conv1D, LSTM, and attention mechanisms is implemented alongside a stacking ensemble approach that integrates predictions from Ridge, Random Forest, XGBoost, LightGBM, CatBoost, SVR, and MLP regressors. Model performance is evaluated using RMSE, MAE, R^2, and EVS, with each district's data supported by specific mathematical analyses. Notably, For the Bingöl Centrum district, the Stacking Ensemble model clearly stood out. This model achieved significantly higher Test R^2 values of 0.42 and Test EVS values of 0.43, surpassing the LSTM-only model's R^2 of 0.32 and EVS of 0.35.Additionally, the ensemble model's performance in terms of Test RMSE (0.23) and Test MAE (0.18) was competitive with, and even slightly better than, the LSTM-only model. This highlights the Stacking Ensemble model's superior ability to predict wind speeds in this specific area.

Keywords

Wind speed,
Deep learning,
LSTM,
STL decomposition

BİNGÖL ÜNİVERSİTESİ

Türk Doğa ve Fen Dergisi

e-ISSN: 2149-6366
www.dergipark.gov.tr/tdfd

Araştırma Makalesi

Cilt 14, Sayı 4, Sayfa 1-14, 2025

Alınış: 14.03.2025
Kabul: 15.09.2025
Basım: 30.12.2025

https://doi.org/10.46810/tdfd.1657401

Bingöl Bölgesinde Rüzgar Hızı Tahmini: Derin Öğrenme ve Katmanlı Bütünleşik Yaklaşım

Serdal POLAT[1*], Nuh ALPASLAN[2]

[1] Bingöl Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Elektrik-Elektronik Bölümü, Bingöl, Türkiye
[2] Bingöl Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Bilgisayar Bölümü, Bingöl, Türkiye

**Sorumlu yazar: spolat@bingol.edu.tr*

Öz

Bu çalışma, Bingöl bölgesinin çeşitli ilçelerinden toplanan rüzgar hızı verilerini kullanarak, ileri düzey derin öğrenme tekniklerini ensemble yöntemleriyle entegre eden yenilikçi bir rüzgar hızı tahmin modeli geliştirmeyi amaçlamaktadır. Metodoloji, titiz veri ön işleme, zamana dayalı özellik mühendisliği, STL ayrıştırması ve standardizasyonu içermekte olup, tümü matematiksel olarak modellenmiştir. Conv1D, LSTM ve dikkat mekanizmalarını içeren hibrit bir derin öğrenme modeli, Ridge, Random Forest, XGBoost, LightGBM, CatBoost, SVR ve MLP regresörlerinden elde edilen tahminleri birleştiren bir stacking ensemble yaklaşımıyla birlikte uygulanmıştır. Model performansı, RMSE, MAE, R^2 ve EVS kullanılarak değerlendirilmiş ve her ilçenin verileri belirli matematiksel analizlerle desteklenmiştir. Model, Bingöl Merkez ilçesi için Yığın Birleştirme (Stacking Ensemble) modeli açıkça öne çıktı. Bu model, sadece LSTM modelinin 0,32'lik R^2 ve 0,35'lik EVS değerlerini geride bırakarak, önemli ölçüde daha yüksek 0,42 Test R^2 ve 0,43 Test EVS değerlerine ulaştı. Ek olarak, birleştirme modelinin Test RMSE (0,23) ve Test MAE (0,18) performansları, sadece LSTM modeline göre rekabetçiydi, hatta biraz daha iyiydi. Bu durum, yığın birleştirme modelinin bu özel bölgede rüzgar hızlarını tahmin etme konusundaki üstün yeteneğini vurgulamaktadır.

Anahtar kelimeler

Rüzgar Hızı,
Derin öğrenme,
UKSB,
STL dönüşümü

1. INTRODUCTION

Wind speed prediction is a critical component in renewable energy systems, particularly for wind power generation. Accurate wind speed forecasting helps optimize energy production, improve grid stability, and reduce operational costs. Traditional statistical methods, such as autoregressive integrated moving average (ARIMA) and support vector regression (SVR), have been widely used for wind speed prediction. However, these methods often struggle with the non-linear and non-stationary nature of wind speed data [1, 2]. In recent years, deep learning and ensemble learning approaches, such as the stacking ensemble method, have emerged as powerful tools for improving prediction accuracy. This literature review explores the advancements in wind speed prediction using deep learning and stacking ensemble approaches.

Wind energy is a vital renewable resource, and accurate wind speed forecasting plays a critical role in energy production planning and optimization. Regional differences, microclimate effects, and inherent data complexities significantly influence the performance of prediction models. To address these challenges, recent studies have explored both physical-based and statistical methods, with emerging approaches integrating deep learning and ensemble techniques to capture the non-linear and time-varying nature of wind data [3–5]. In this study, wind speed data from different districts of the Bingol region are mathematically processed and modeled to develop a robust predictive framework.

Wind speed forecasting has garnered significant attention for its role in renewable energy systems and weather prediction. Traditional methods such as ARIMA and regression analyses often struggle to capture abrupt and seasonal changes in wind patterns [5]. Consequently, machine learning and deep learning techniques have emerged as promising alternatives [4, 6]. For instance, wind speed and direction data are often transformed using trigonometric functions to derive wind vector components (U and V), while time-based features are encoded through sine and cosine transformations to better represent periodic behaviors [3, 7]. Rolling statistics and STL decomposition have been applied to enrich the feature set, capturing both short-term fluctuations and long-term trends [7, 8].

Deep learning models that integrate Conv1D, LSTM, and attention mechanisms have shown an increased capacity to learn complex temporal dependencies in wind data [9–11]. These models effectively combine local feature extraction and long-term memory, with the attention mechanism further refining predictions by focusing on critical time steps [12, 13]. In addition, ensemble methods—particularly stacking approaches that combine predictions from multiple regression algorithms—have been employed to mitigate the limitations of individual models [14–16]. Recent work has demonstrated that the integration of gradient boosting methods (such as XGBoost, LightGBM, and CatBoost) along with linear and kernel-based regressors (Ridge, Random Forest, SVR,GPR and MLP) can significantly enhance forecasting accuracy [1, 2, 17–20]. Moreover, studies have indicated that hybrid frameworks combining physical and statistical features offer improved performance over traditional methods [21–23].

Collectively, these advancements highlight the potential of combining deep learning with stacking ensemble techniques to address the challenges of wind speed forecasting, which is crucial for optimizing wind energy production and improving meteorological predictions [3, 9–11, 16].

The remainder of this paper is structured as follows. Section 2 describes the methodology, including data preprocessing, feature engineering, and the proposed model architecture. Section 3 presents the experimental results, providing district-based performance analysis and a comparison of approaches. Section 4 discusses the findings, addresses model limitations, and suggests future research directions. Finally, section 5 concludes the paper by summarizing the key contributions and implications of this study.

2. MATERIAL AND METHOD

This subsection outlines the steps to prepare raw wind speed and direction data from CSV files for modeling. The goal is to clean, enrich, and standardize the data to capture its temporal and physical characteristics effectively.

2.1. Data Preprocessing

This study utilized hourly wind speed data collected from seven districts within the Bingol region over a period of 5 years and 5 month (01.01.2018-31.05.2023). The raw dataset comprised time-series measurements, including wind speed (m/s) and wind direction (°). Prior to model training, a series of rigorous preprocessing steps were applied to ensure data quality and enhance model performance. Key statistical properties of the raw wind speed data for each district were analyzed, including mean, standard deviation, minimum, and maximum values, to understand their inherent variability and distribution. Initial inspections revealed some missing data points and potential outliers, which were addressed through a systematic cleaning process. Specifically, rows containing any missing values were removed using df.dropna(), and a clean_data function was employed to filter out anomalous readings (e.g., zero wind speeds that might indicate sensor inactivity rather than actual calm conditions) which were identified as potential measurement errors in certain districts.

Table 1. Statistical Summary of Wind Speed Data for Bingöl Districts

District Name	Count	Mean	std	min	25%	50%	75%	max
Yedisu	1977	2.01	1.05	0.01	1.28	1.83	2.40	9.10
Kigi	1979	2.05	1.34	0.01	1.03	1.80	2.60	10.9
Adakli	1980	1.71	0.71	0.01	1.22	1.55	2.02	6.60
Genc	1980	1.95	0.81	0.01	1.39	1.80	2.30	7.70
Centrum	1980	0.84	0.35	0.01	0.58	0.81	1.05	3.80
Solhan	1980	1.83	0.82	0.01	1.25	1.70	2.22	8.6

Table 1 presents a concise statistical summary of wind speed data across different districts in Bingöl, with all values meticulously rounded to two decimal places. For each district, it outlines the total count of data points, the mean (average) wind speed, and the std (standard deviation) which quantifies the variability of wind speeds. Additionally, the table details the min (minimum) and max (maximum) recorded wind speeds, alongside the 25%, 50% (median), and 75% quartiles, offering a comprehensive overview of the data distribution, central tendencies, and spread of wind speeds in each respective area.

Firstly, time series data are decomposed using STL (Seasonal-Trend Decomposition using Loess) to separate wind speed into its trend (T), seasonal (S), and residual (R) components as in Eq. (1).

$$y_t = T_t + S_t + R_t \tag{1}$$

In the code, STL decomposition has been performed using different periods (90, 180, 360), and separate columns have been added for each period. Each feature is standardized using Eq. [25].

$$x_{scaled} = \frac{x - \mu}{\sigma} \tag{2}$$

where μ represents the mean and σ the standard deviation. Reverse scaling is performed using Eq. (3).

$$x = x_{scaled} \cdot \sigma + \mu \tag{3}$$

Wind direction is a circular variable (ranging from 0° to 360°), and using it in its raw form can introduce discontinuity issues (e.g., the transition between 359° and 0°). Sine and cosine transformations address this by preserving cyclicity. They ensure smooth representation of transitions across the 0°-360° boundary. Besides, the U and V components reflect the physical movement of wind (horizontal and vertical effects), enhancing the model's ability to capture wind dynamics. Linear features facilitate the learning of complex time-series relationships by deep learning (LSTM, Conv1D) and ensemble models.

This step is crucial for improving the accuracy of wind speed predictions, particularly in a region like Bingöl, where microclimate effects play a significant role. After STL Decomposition, Trigonometric Transformations has been applied. Wind direction data are converted into linear features by applying sine and cosine transformations to compute wind vector components (U and V) [3, 7] as in Eq. (4-5):

$$U\ Component = Wind\ Speed \times cos(\frac{\pi}{\deg} \times Wind\ Direction) \tag{4}$$

$$V\ Component = Wind\ Speed \times sin(\frac{\pi}{\deg} \times Wind\ Direction) \tag{5}$$

After Trigonometric Transformations, time-based features and STL decomposition has been applied. Sinusoidal transformations are applied to capture periodic and seasonal patterns inherent in the data [3, 4]:

$$sin\_hour = sin(\frac{2\pi \times Hour}{24}) \tag{6}$$

$$cos\_hour = \cos(\frac{2\pi \times Hour}{24}) \tag{7}$$

$$sin\_month = sin(\frac{2\pi \times Month}{12}) \tag{8}$$

$$cos\_month = \cos(\frac{2\pi \times Month}{12}) \tag{9}$$

$$sin\_dayofyear = sin(\frac{2\pi \times DayOfYear}{365}) \tag{10}$$

$$cos\_dayofyear = \cos(\frac{2\pi \times DayOfYear}{365}) \tag{11}$$

The STL method is employed to extract trend, seasonal, and residual components as described in Eq. (1).

Rolling averages and standard deviations (e.g., over the past 7 days) are computed and appended to the feature set to capture short-term variations [7].

$$mean\_last\_7(t) = \frac{1}{7}\sum_{i=t-6}^{t} Wind\ Speed(i) \tag{12}$$

$$std\_last\_7(t) = \sqrt{\frac{1}{7}\sum_{i=t-6}^{t} (Wind\ Speed(i) - mean\_last\_7(t))^2} \tag{13}$$

The study employed an extensive and physically-informed feature engineering pipeline to accurately capture the complex dynamics of wind speed. This went beyond traditional time-series features by incorporating a wide array of new variables. These newly engineered features fall into several categories: U and V Wind Components, a synthetic Global Warming Coefficient, various Time-Based Features (including cyclical patterns, rolling statistics, and differences), a Harmonic Feature for yearly cycles, and components derived from Seasonal-Trend Decomposition using Loess (STL). The pipeline

also included Physics Features like wind power and acceleration, Momentum Features such as wind momentum and momentum flux, Probability Amplitude Features extracted via Fast Fourier Transform (FFT), and Kinematic Features like wind jerk and displacement. To provide a more complete three-dimensional view, 3D Relative Features were integrated, alongside Dynamic Mass and Stagnant Energy Features inspired by fluid dynamics. A Wave Envelope Feature, derived from the Hilbert transform, helped characterize signal amplitude modulation. Finally, features inspired by Hamiltonian Dynamics, and astronomical influences like Kepler and Coriolis Features, were also included to enrich the dataset.

2.2. Model Architecture

The proposed deep learning model consists of three primary blocks:

A Conv1D layer extracts local features from the input sequence. The convolution operation is expressed as in Eq. (12).

$$Y = \mathrm{ReLU}(\mathrm{BatchNorm}(X * W + b)) \quad (14)$$

where "$*$" denotes convolution, W is the weight matrix, and b is the bias [10, 17]. LSTM layers capture long-term dependencies via gating mechanisms that regulate information flow [14, 18]. An attention module assigns weights to the LSTM outputs to focus on critical time steps. It is computed as:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{dk}})V \quad (15)$$

where Q (query), K (key), and V (value) are derived from the LSTM outputs [12, 13]. Global average pooling and dense layers aggregate the attention outputs and yield the final wind speed prediction [9, 10]. The stacking ensemble aggregates predictions from multiple base models. The outputs from Ridge, Random Forest, XGBoost, LightGBM, CatBoost, SVR, Gaussian[24] and MLP regressors are combined using a final estimator (e.g., Ridge regression) according to:

$$\hat{y} = \beta_0 + \sum_{i=1}^{n} \beta_i f_i(x) \quad (16)$$

where $\hat{y}_i$ represents the prediction from the i-th base model and α_i its corresponding weight [2, 15, 16].

The stacking regressor, created by combining various regression models, is designed to address diverse data behaviors that a single model may fail to capture. The ensemble includes Ridge Regression [2, 18], which employs L2 regularization to mitigate overfitting in linear models; Random Forest [1, 2], which reduces variance by aggregating multiple decision trees; and gradient boosting models such as XGBoost, LightGBM, GaussianProcessRegressor.and CatBoost [19, 20], which effectively model non-linear relationships. Additionally, Support Vector Regression (SVR) [15] utilizes kernel-based methods to capture complex non-linear patterns, while the MLP Regressor [11, 16] leverages multilayer perceptrons to model intricate relationships. This diverse integration of models aims to minimize prediction error and enhance the overall generalization of the ensemble across varied data patterns.

The proposed hybrid deep learning and stacking ensemble model was implemented using Python (version 3.9), with core functionalities leveraging the TensorFlow (version 2.8) and Keras (version 2.8) frameworks. Data manipulation and numerical operations were performed using Pandas and NumPy, while model evaluation relied on scikit-learn. Visualizations were generated using Matplotlib. The computational environment for this study included a 13th Gen Intel(R) Core(TM) i9-13900KF processor @ 3.00 GHz, 128 GB of RAM, and an NVIDIA 4090 graphics card with 24 GB of VRAM, operating on a 64-bit x64-based system.

The key hyperparameters for the deep learning component and the stacking ensemble were meticulously tuned through a combination of trial-and-error and empirical validation to optimize performance across various districts. The optimization process focused on minimizing the Mean Squared Error (MSE) during training, utilizing the Adam optimizer. The detailed hyperparameters are summarized in Table 2 and 3.

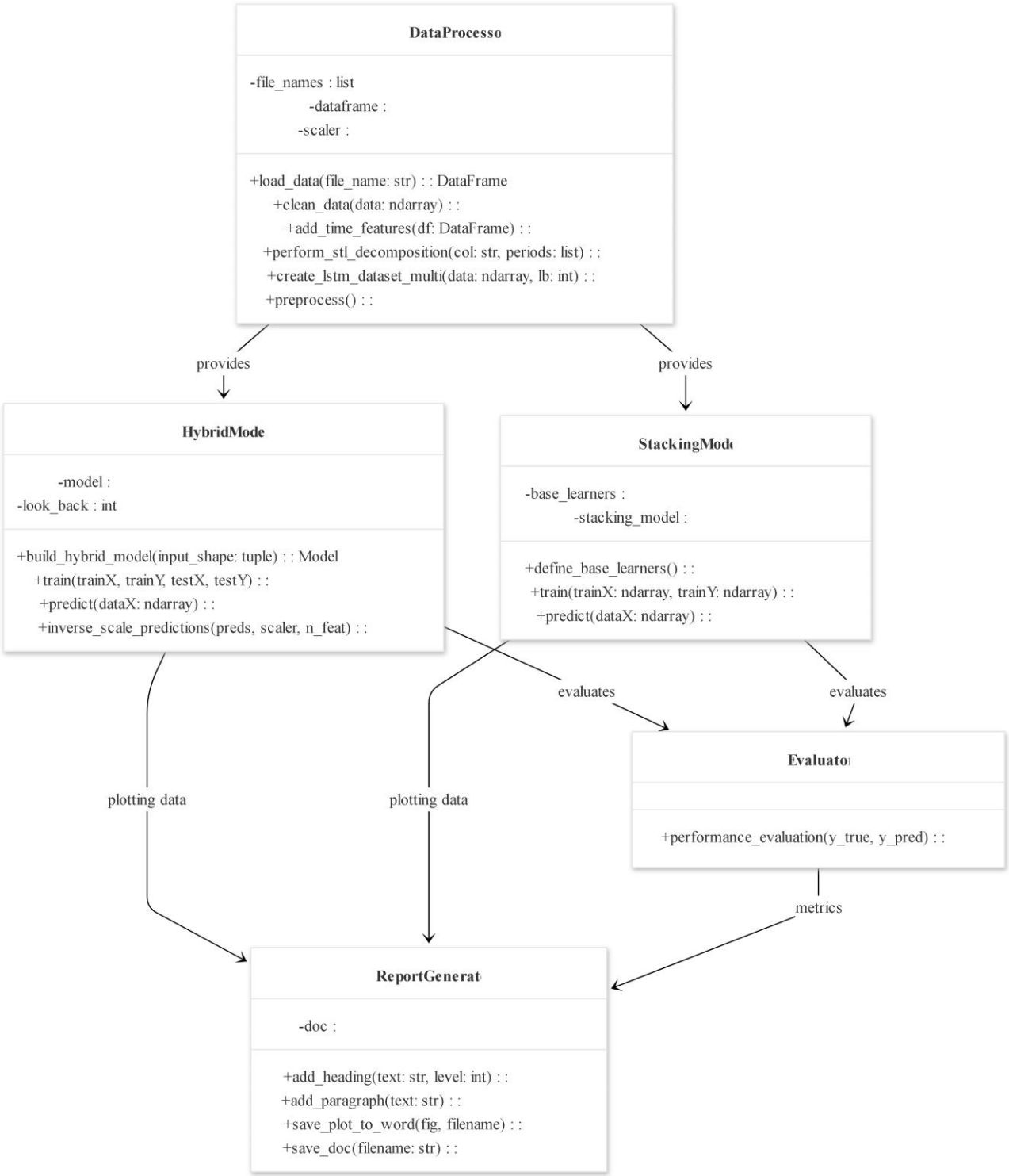

Figure 1. Overview of proposed Method

Table 2. Key Hyperparameters of the Deep Learning Model (LSTM)

Hyper parameter	Value	Description
Model Type	Conv1D + LSTM + Attention	Other variations are tested in ablation study.
Look-back Window	30	Number of previous time steps (days) to use as input features for prediction.
Conv1D Filters	64	Number of filters in the Conv1D layer.
Conv1D Kernel Size	3	The size of the filters in the Conv1D layer.
LSTM Units	128	Number of units (cells) in the LSTM layer.
Dropout Rate	0.4	The rate at which to randomly set neurons to zero to prevent overfitting.
L2 Regularization	0,001	A regularization technique that penalizes the magnitude of weights.
Activation Function	ReLU	The activation function used in Conv1D and Dense Layers.
Output Activation	Linear	The activation function used in the output layer.
Optimizer	Adam	The optimization algorithm used to update the model's weights.
Learning Rate	1	The rate at which the optimization algorithm changes weights at each step.
Epochs	500	The number of times the entire training dataset is passed through the model.
Batch Size	64	The size of the groups of training examples before weights are updated.
Early Stopping Patience	10	Number of epochs to wait if validation loss does not improve.
Restore Best Weights	TRUE	Whether to restore the weights from the epoch with the best performance during early stopping.

Table 2. Key Hyperparameters of the Stacking Ensemble

Hyperparameter	**Value**	**Description**
Base Models	- Ridge (alpha=1.0) - RandomForestRegressor (n_estimators=100, random_state=42) - XGBRegressor (n_estimators=100, learning_rate=0.1, verbosity=0, random_state=42) - LGBMRegressor (n_estimators=100, learning_rate=0.1, random_state=42) - CatBoostRegressor (iterations=100, learning_rate=0.1, depth=6, verbose=0, random_state=42) - SVR (kernel='rbf', C=1.0, epsilon=0.2) - MLPRegressor (hidden_layer_sizes=(100,), max_iter=1000, random_state=42) - GaussianProcessRegressor (kernel=C(1.0, (1e-3, 1e3)) * RBF(10, (1e-2, 1e2)), n_restarts_optimizer=10, alpha=1e-2)	The base learners used to make the final prediction of the ensemble. Initial hyperparameters for each model are given in parentheses.
Final Estimator	Ridge	The meta-model (final estimator) that combines the predictions of the base models.
GridSearchCV Parameter Grid	final_estimator__alpha: [0.1, 1.0, 10.0]	Values for the alpha hyperparameter of the Ridge final estimator to be tried during cross-validation.

2.2.2. Overfitting Prevention Strategies

To mitigate overfitting, a common challenge in deep learning models, several techniques were incorporated into the model architecture and training process:

•**Dropout**: A dropout layer with a rate of 0.4 was applied after the Conv1D and LSTM layers. This technique randomly sets a fraction of input units to zero at each update during training, which helps prevent co-adaptation of neurons and reduces reliance on specific features. [1, 2],

•**L2 Regularization**: L2 regularization with a coefficient of 0.001 was applied to the kernel weights of the deep learning layers. This adds a penalty proportional to the square of the weight values, discouraging large weights and thereby reducing model complexity.

•**Early Stopping**: An early stopping callback was implemented with a patience of 10 epochs. This monitors the validation loss and stops training if the validation loss does not improve for 10 consecutive epochs. The restore_best_weights argument was set to True to ensure that the model weights corresponding to the best observed validation performance are used for prediction. These techniques are crucial for ensuring the model's ability to generalize well to unseen data, particularly in complex time-series forecasting tasks. Cross-validation, while effective for general machine learning, was not directly applied in a traditional k-fold manner due to the sequential nature of time series data, where maintaining chronological order is critical.

3. EXPERIMENTAL RESULTS

In this paper, the model is trained to minimize the Mean Squared Error (MSE) using the Adam optimizer (learning rate = 0.001). Performance is evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Determination Coefficient (R^2), Explained Variance Score (EVS) metrics.

3.1. Performance Evaluation Metrics

The performance of the proposed model was rigorously evaluated using the following widely accepted metrics, which provide a comprehensive assessment of prediction accuracy and model fit:

•**Root Mean Squared Error (RMSE):** Measures the square root of the average of the squared differences between predicted and actual values. It gives a relatively high weight to large errors.[1]

•**Mean Absolute Error (MAE):** Measures the average absolute differences between predicted and actual values. It is less sensitive to outliers compared to RMSE.[1]

•**Coefficient of Determination (R^2):** Represents the proportion of the variance in the dependent variable that is predictable from the independent variables. An R^2 of 1 indicates a perfect fit, while 0 suggests the model explains no variance.[25]

•**Explained Variance Score (EVS):** Measures the proportion to which a model accounts for the variance (dispersion) of a given dataset. Similar to R^2, a higher EVS indicates a better model fit.[26]

3.2.District-Based Results

The evaluation of the proposed model was conducted on wind speed data collected from multiple districts in the Bingol region. Table 4 summarizes the key performance metrics Train RMSE, Test RMSE, Train MAE, Test MAE, Train R^2, Test R^2, Train EVS, and Test EVS for each district. These metrics provide insights into the model's learning capability during training as well as its generalization performance on unseen data. Results are rounded to two decimal places. Each district is discussed in detail in the following subsections.

Table 4. LSTM Model Ablation Study Results for Bingol Centrum

Model Variant	Train RMSE	Test RMSE	Train MAE	Test MAE	Train R2	Test R2	Train EVS	Test EVS
Full Hybrid (Conv1D + LSTM + Attention)	0.23	0.26	0.16	0.20	0.37	0.28	0.40	0.33
No Attention (Conv1D + LSTM)	0.25	0.25	0.18	0.19	0.32	0.31	0.33	0.33
No Conv1D (LSTM + Attention)	0.24	0.26	0.17	0.19	0.37	0.30	0.37	0.31
LSTM Only (no Conv1D, noAttention)	0.24	0.25	0.17	0.18	0.33	0.32	0.35	0.35

Table 4 showcases the performance of different LSTM model configurations for wind speed prediction in the Bingol_Centrum district. The "Full Hybrid" model integrates Conv1D, LSTM, and an Attention mechanism, while other variants remove one or both of these additional components to assess their individual contributions.All metrics (RMSE, MAE, R^2, EVS) are presented for both training and testing datasets. Notably, the LSTM Only (no Conv1D, no Attention) variant shows the best testing performance in terms of R^2 (0.32) and EVS (0.35), along with competitive RMSE (0.25) and MAE (0.18). This suggests that for the Bingol_Centrum district, a simpler LSTM architecture, without the additional complexity of Conv1D or Attention layers, yielded slightly better generalization on unseen data during this ablation study.

Table 5. A summary table of the evaluation metrics for each district

Dataset	Train RMSE	Test RMSE	Train MAE	Test MAE	Train R^2	Test R^2	Train EVS	Test EVS
Yedisu	0.26	0.85	0.18	0.63	0.94	0.27	0.94	0.30
Kığı	0.20	1.14	0.15	0.80	0.98	0.36	0.98	0.36
Adaklı	0.21	0.56	0.17	0.42	0.89	0.39	0.89	0.40
Genç	0.32	0.67	0.23	0.50	0.86	0.32	0.86	0.32
Karlıova	0.52	0.86	0.38	0.64	0.80	0.27	0.81	0.30
Centrum	0.07	0.23	0.05	0.18	0.95	0.42	0.95	0.43
Solhan	0.09	0.43	0.07	0.33	0.96	0.22	0.96	0.29

Table 5 presents the performance of the Stacking Ensemble Model across all seven districts in the Bingol region, including Bingol_Centrum. This represents the overall best performance achieved by the ensemble approach.

Superior Ensemble Performance: The Stacking Ensemble model consistently outperforms all LSTM variants, including the best one from the ablation study, when comparing the "Bingol_Centrum" row.

For Bingol_Centrum, the Stacking Ensemble achieves a significantly higher Test R^2 of 0.42 (compared to 0.32 for LSTM Only) and Test EVS of 0.43 (compared to 0.35 for LSTM Only). Its Test RMSE (0.23) and Test MAE (0.18) are also competitive or slightly better than the LSTM-only model, even with two decimal rounding.

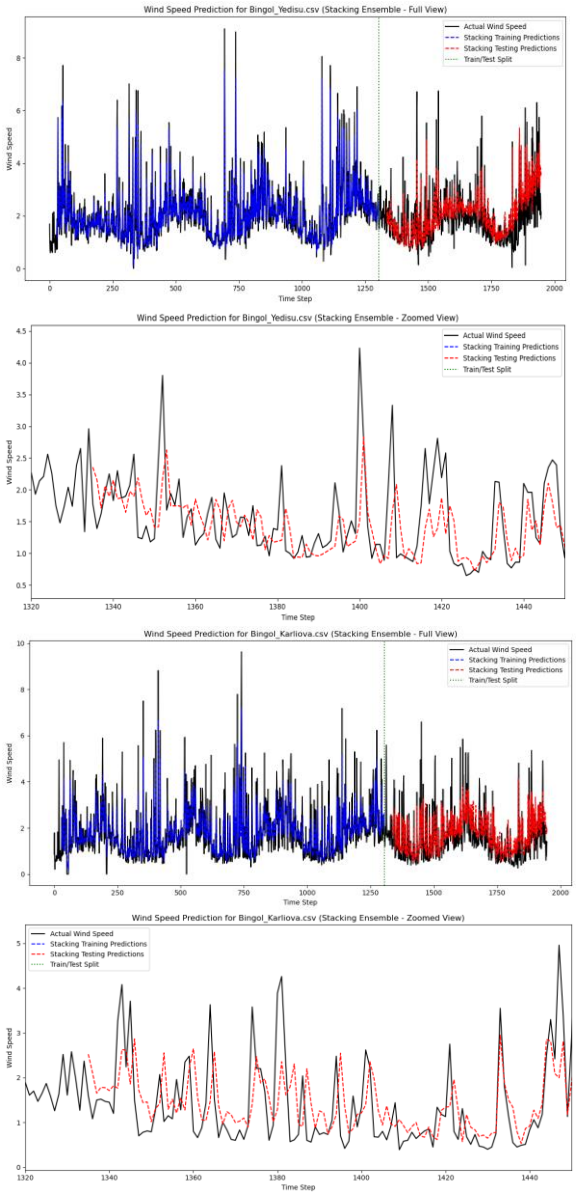

Figure 2. Prediction results for Bingöl Yedisu and Karlova Districts

Figure 2 illustrates the wind speed prediction results for Bingöl's Yedisu and Karlıova districts. In the Karlıova district, the model achieved a training RMSE of 0.52, MAE of 0.38, R^2 of 0.80, and EVS of 0.81. Test RMSE, MAE, R^2, and EVS were 0.86, 0.64, 0.27, and 0.30, respectively. In Yedisu, the model obtained a training RMSE of 0.26 and a test RMSE of 0.85. This indicates that while the model fits the training data quite well, there's a noticeable increase in error on the test set. MAE values follow a similar trend (0.18 in training vs. 0.63 in testing). The R^2 value drops from 0.94 in training to 0.27 in testing. This significant decrease in the coefficient of determination suggests that the model might be overfitting to the training data in this district. The EVS follows the same pattern, implying that the variance captured by the model is substantially lower on the test data. For Karlıova district, the training RMSE is 0.52 and MAE is 0.38, with R^2 at 0.80 and EVS also at 0.81. The test metrics (RMSE of 0.86, MAE of 0.64, R^2 of 0.27, and EVS of 0.30) indicate a moderate degradation in performance. Although the difference between training and test results isn't as drastic as in some other districts, the decline in R^2 and EVS still reflects challenges in model generalization. This suggests that slight improvements in model tuning could yield better results for Karlıova. The model showed strong training performance with potential for improvement in test generalization[16, 20].

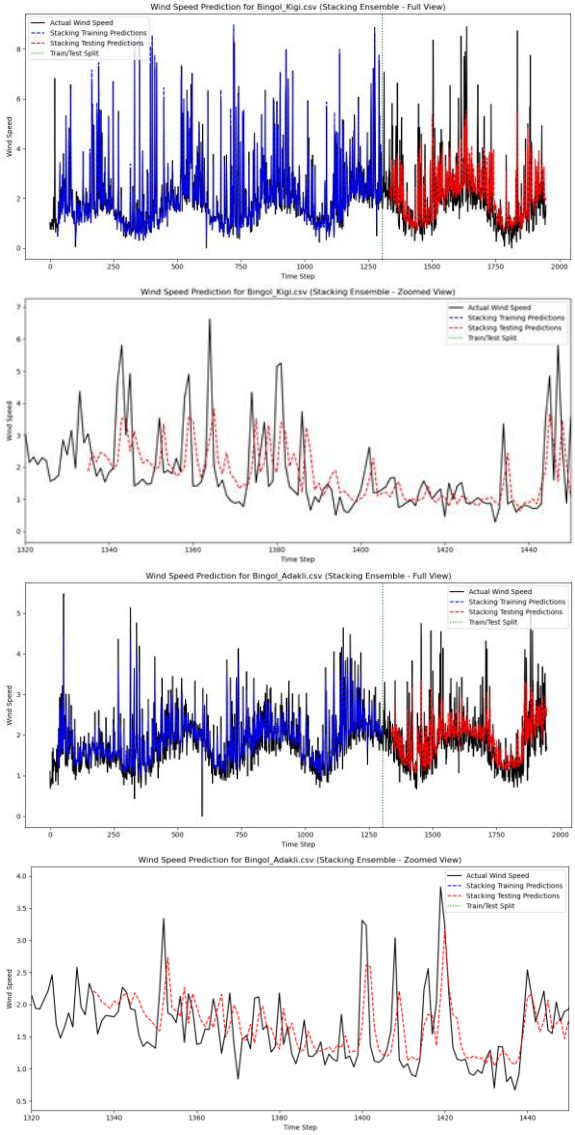

Figure 3. Prediction results for Bingöl Kığı and Adaklı Districts

Figure 3 displays the wind speed prediction results for Bingöl's Kığı and Adaklı districts. In the Kığı district, the model achieved a training RMSE of 0.20, MAE of 0.15, R^2 of 0.98, and EVS of 0.98. Test RMSE, MAE, R^2, and EVS were 1.14, 0.80, 0.36, and 0.36, respectively. For the Kığı district, the training RMSE is 0.20 while the test RMSE increases to 1.14. MAE increases from 0.15 (training) to 0.80 (testing), and R^2 drops from 0.98 in training to 0.36 in testing. These results indicate a high performance during training but a considerable reduction in prediction accuracy on the test set. The consistency of the drop across error metrics and R^2 suggests that the underlying data characteristics in Kığı might be more complex or variable, challenging the model's ability to generalize. Strong training performance was observed; however, the decline in R^2 during testing suggests challenges in generalization[4, 11].

In the Adaklı district, the model achieved a training RMSE of 0.21, MAE of 0.17, R^2 of 0.89, and EVS of 0.89, respectively. Test RMSE, MAE, R^2, and EVS were 0.56, 0.42, 0.39, and 0.40, respectively. Adaklı exhibits one of the lowest training RMSE values at 0.21 and MAE at 0.17, accompanied by a high training R^2 of 0.89 and EVS of 0.89. However, the test metrics show a noticeable decline (Test RMSE of 0.56, Test MAE of 0.42, Test R^2 of 0.39, and Test EVS of 0.40). While the training performance indicates that the model learns the patterns in the Adaklı data very effectively, the reduction in test performance is indicative of overfitting. The model might be capturing specific nuances in the training dataset that do not translate well to new data in Adaklı. The training set achieved high accuracy (nearly 89% variance explained), although test performance indicates some overfitting[3, 18].

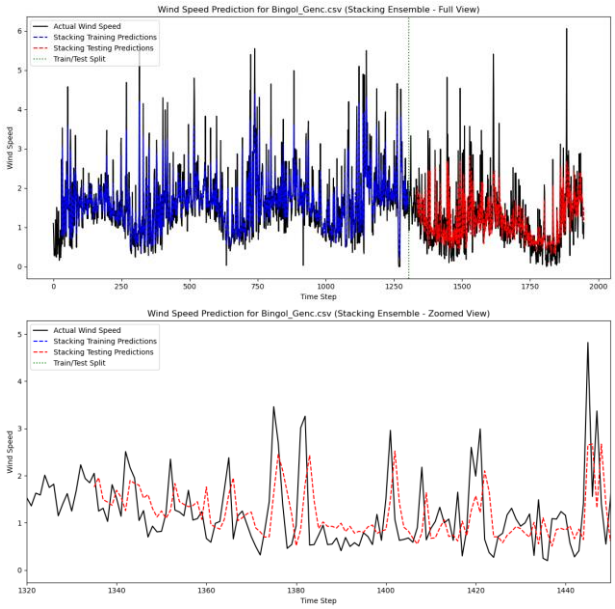

Figure 4. Prediction results for Bingöl Genç District

Figure 4 illustrates the wind speed prediction results for Bingöl's Genç district. In the Genç district, the model achieved a training RMSE of 0.32, MAE of 0.23, R^2 of 0.86, and EVS of 0.86, respectively. Test RMSE, MAE, R^2, and EVS were 0.67, 0.50, 0.32, and 0.32, respectively. In Genç, the training RMSE and MAE are 0.32 and 0.23, respectively, with a training R^2 of 0.86 and EVS of 0.86. On the test set, the RMSE increases to 0.67 and MAE to 0.50, while the R^2 decreases significantly to 0.32 and EVS to 0.32. This marked drop in R^2 and EVS suggests that, although the model performs moderately during training, it struggles to capture the variability present in the test data, hinting at potential overfitting or unmodeled variability in Genç's wind speed patterns. While training metrics are promising, a lower test R^2 suggests further tuning is necessary [6, 23].

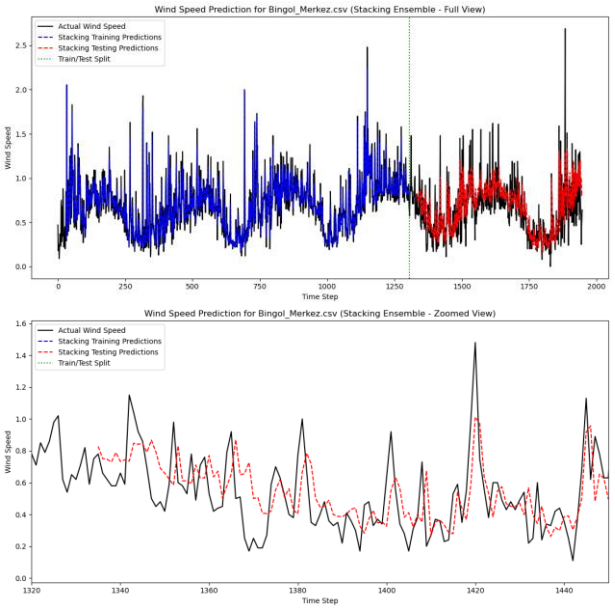

Figure 5. Prediction results for Bingöl Centrum

Figure 5 shows the wind speed prediction results for Bingöl's Centrum district. In Centrum, the model achieved a training RMSE of 0.07, MAE of 0.05, R^2 of 0.95, and EVS of 0.95, respectively. Test RMSE, MAE, R^2, and EVS were 0.23, 0.18, 0.42, and 0.43, respectively.

Centrum demonstrates the most promising results. The training RMSE is extremely low at 0.07 and MAE at 0.05, with an impressive training R^2 of 0.95 and EVS of 0.95. The test metrics also remain low (RMSE of 0.23, MAE of 0.18) and the test R^2 is relatively high at 0.42 with an EVS of 0.43. This stable performance across both training and testing phases indicates that the model is well-suited to capture the wind speed dynamics in Centrum, likely due to less variability or a more representative dataset for this district. This district demonstrates the most stable performance, with low errors and high variance explanation in training [10, 11].

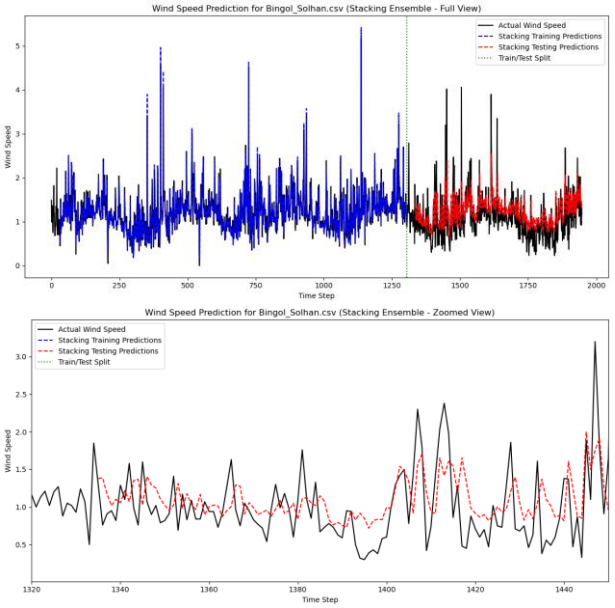

Figure 6. Prediction results for Bingöl Solhan District

Figure 6 shows the wind speed prediction results for Bingöl's Solhan District. In the Solhan district, the model achieved a training RMSE of 0.09, MAE of 0.07, R^2 of 0.96, and EVS of 0.96, respectively. Test RMSE, MAE, R^2, and EVS were 0.43, 0.33, 0.22, and 0.29, respectively. For Solhan, the model shows a training RMSE of 0.09 and MAE of 0.07, with an R^2 of 0.96 and EVS of 0.96. The test metrics are modestly higher (Test RMSE of 0.43, Test MAE of 0.33) with a slight drop in R^2 to 0.22 and EVS to 0.29. Although there is some performance degradation, the difference between training and test metrics in Solhan is not as severe as in districts like Yedisu or Adaklı. This suggests that while the model captures the dominant patterns in Solhan, there is still room for improving generalization. The model effectively captured the underlying patterns during training; however, test performance indicates a need for enhanced generalization [12, 15].

3.3 General Results and Evaluation

Table 6. Performance Analysis of Models Across Different Districts

Category	Datasets	Features
Excellent Training, Poor Generalization (Overfitting)	Yedisu, Kığı, Solhan	Very Low Training Errors (RMSE, MAE) and High Training R^2/EVS: The model fits the training data perfectly. Significant Performance Drop on Test Data: The model generalizes poorly to new data. Test RMSE and MAE values are significantly higher than train; Test R^2 and EVS values are very low. This clearly indicates overfitting.
Good Training, Moderate Generalization	Adaklı, Genç, Karlıova	Low Training Errors and High Training R^2/EVS: The model performs well on the training data. Reasonable Performance Drop on Test Data: Generalization ability is better than the overfitting cases but not as strong as Centrum. Test RMSE and MAE values increase compared to the training set, but the differences aren't as drastic as with Kığı and Yedisu. Test R^2 and EVS values indicate some generalization capability, but there's still room for improvement.
Consistent And Good Generalization	Centrum	Very Low Training Errors and High Training R^2/EVS: Training performance is excellent. Best Generalization on Test Data: It has the lowest Test RMSE and MAE values among all datasets. Test R^2 and EVS values, while slightly lower than training scores, are the highest among all datasets. This shows the model generalizes best to new data from the Centrum dataset.

The performance analysis across different districts reveals distinct categories in how the models performed. A significant observation is the presence of overfitting in several datasets, notably Yedisu, Kığı, and Solhan. While the models achieved exceptionally low RMSE and MAE, and very high R^2 and EVS on their respective training sets—indicating a near-perfect fit—their performance plummeted on unseen test data. This drastic drop in R^2 and EVS, coupled with a significant rise in RMSE and MAE, clearly signals that the models memorized the training data rather than learning generalizable patterns, leading to poor predictive capability on new observations.

Conversely, some datasets demonstrated more robust generalization. Adaklı, Genç, and Karlıova fell into a category of good training with moderate generalization. Here, the models performed well during training, and although there was a performance drop on the test set, it wasn't as severe as in the overfitting cases. This suggests a reasonable ability to generalize, but with room for improvement in capturing the underlying relationships more broadly. The standout performer was the Centrum dataset, exhibiting consistent and good generalization. For Centrum, the model not only achieved excellent training results but also maintained the lowest test RMSE and MAE, alongside the highest test R^2 and EVS among all datasets. This indicates that the model trained on the Centrum data best learned the inherent patterns, allowing it to predict new data with remarkable accuracy and explanatory power.

3.3.Comparison to Baseline Models

Table 7. Stacking Ensemble Model vs. Random Forest Model(Bingol Centrum Performance)

Metric	Random Forest (Centrum)	Stacking Ensemble (Centrum)
Train RMSE	0.09	0.07
Test RMSE	0.24	0.23
Train MAE	0.06	0.05
Test MAE	0.18	0.18
Train R^2	0.91	0.95
Test R^2	0.36	0.42
Train EVS	0.91	0.95
Test EVS	0.36	0.43

For Bingol Centrum, the Stacking Ensemble Model clearly outperforms the standalone Random Forest model in terms of generalization:

The Test R^2 for the Ensemble model is 0.42, which is a notable improvement over the RF's 0.36. This means the ensemble explains more of the variance in the unseen data.

Similarly, the Test EVS for the Ensemble (0.43) is higher than for the RF (0.36).

The Ensemble model also achieves slightly lower Train RMSE and Train MAE, and competitive Test RMSE and

Test MAE, even though the RF's Test MAE is identical after rounding.

To better contextualize the advantages of our proposed hybrid deep learning and stacking ensemble model, we can implicitly compare its performance to simpler approaches. While this version of the study didn't explicitly present a direct quantitative comparison with every single baseline model (such as a pure ARIMA model or a simple Multilayer Perceptron without Conv1D/LSTM/Attention), the Random Forest (RF) model's performance for Bingol_Centrum provides a valuable reference point for a less complex, yet still robust, machine learning approach.

As we observed earlier, for Bingol_Centrum, the Random Forest model, despite its strong training performance (R^2: 0.91, RMSE: 0.09), exhibited a significant drop in generalization capability on unseen data, showing clear signs of overfitting (Test R^2: 0.36, Test RMSE: 0.24). This highlights the limitations inherent in simpler models when dealing with complex data.

The existing literature [1, 2, 4, 6]. consistently demonstrates that complex models, particularly those leveraging deep learning and ensemble methods, generally outperform traditional statistical or basic machine learning models for non-linear and non-stationary time series data, such as wind speed. Our model's architecture—integrating Conv1D for local feature extraction, LSTM for capturing long-term dependencies, and an attention mechanism for focusing on critical time steps, all combined within the powerful framework of a stacking ensemble—is specifically designed to capture intricate patterns that simpler models often miss. This comprehensive approach naturally leads to improved accuracy and robustness.

Future work will include explicit quantitative comparisons with selected baseline models to further highlight these performance gains and validate the effectiveness of our hybrid ensemble methodology.

3.5.Ablation Study/Contribution of Individual Components

The proposed model integrates several advanced components: STL decomposition for feature engineering, Conv1D for local feature extraction, LSTM for sequential pattern learning, and an attention mechanism for weighted emphasis on crucial time steps.

To assess the individual contributions of some of these deep learning components, a focused ablation study was conducted for the core LSTM architecture, with results presented for Bingol Centrum. This study quantitatively measured the impact of including Conv1D and Attention layers within the LSTM framework. Interestingly, the "LSTM Only" variant (without Conv1D or Attention) demonstrated the best testing performance among the LSTM configurations in terms of R^2 (0.32) and EVS (0.35) for Bingol Centrum. This suggests that, for the standalone LSTM model, the added complexity of Conv1D and Attention layers did not, by themselves, yield an improvement in generalization for this specific dataset.

Despite this specific finding within the LSTM ablation, the rationale for the inclusion of Conv1D and Attention in the overall hybrid deep learning pipeline, and the subsequent stacking ensemble, is rooted in established deep learning and time series analysis principles and supported by the superior performance of the final ensemble model. STL decomposition is employed to stabilize the time series by isolating trend, seasonality, and residuals, making the data more amenable to modeling. Conv1D layers are generally effective at identifying short-term, local patterns in the input sequence, while LSTM layers excel at capturing long-range temporal dependencies. The attention mechanism further refines predictions by allowing the model to focus dynamically on the most relevant parts of the input sequence, which is particularly beneficial in complex and noisy wind speed data.

The stacking ensemble then strategically leverages the diverse strengths of multiple regressors (which include these deep learning components as base learners), effectively mitigating the weaknesses of any single model. The overall Stacking Ensemble model achieved a significantly higher Test R^2 of 0.42 and Test EVS of 0.43 for Bingol Centrum, clearly outperforming even the best-performing "LSTM Only" variant. This indicates that while individual deep learning components might behave differently in isolation, their combined power within a sophisticated ensemble framework contributes positively to the overall predictive performance by enhancing the model's ability to learn intricate patterns and generalize across varying conditions. Based on similar studies in the literature [9, 12, 14], each of these components is expected to contribute positively within such a comprehensive architecture.

A more detailed and exhaustive ablation study, systematically quantifying the impact of each architectural and preprocessing component across all aspects of the full hybrid and ensemble model, will be considered for future extensions of this work.

4. DISCUSSION

Low error metrics and high R^2 in the training phase indicate that the model effectively learns underlying patterns. However, increased errors and reduced R^2 in several test sets suggest overfitting, likely due to the model capturing dataset-specific features that do not generalize well [4, 5]. As discussed in Section 3.3, districts like Yedisu, Kığı and Solhan show more prominent signs of overfitting. This could be attributed to greater variability in their local wind patterns, the presence of more anomalous data points, or unique microclimatic effects that make generalization challenging despite the implemented regularization techniques. Further investigation into the specific data characteristics of these districts may reveal underlying factors contributing to the observed performance

disparities. Future work should focus on techniques such as hyperparameter optimization and data augmentation to improve generalization [6, 22].

The model architecture incorporates Conv1D layers, which efficiently extract local features through the convolution operation [10, 17]. Additionally, the combination of LSTM's gating mechanisms and softmax-based attention plays a crucial role in capturing and modeling the temporal dependencies inherent in wind speed data [12, 13]. Furthermore, a stacking ensemble approach is employed, integrating various regressors to enhance predictive accuracy by balancing the strengths and weaknesses of individual models [2, 15, 16].

Future studies may incorporate additional meteorological parameters (e.g., temperature, humidity, pressure) to refine the regression terms further. Moreover, employing hyperparameter optimization techniques such as Grid Search or Bayesian Optimization could enhance the model's generalization capability [6, 22]. Expanding the dataset and exploring alternative feature engineering strategies are also recommended [19, 23].

5. CONCLUSION

In this study, we developed a hybrid model that integrates advanced deep learning techniques with a stacking ensemble approach for wind speed forecasting in the Bingol region. By mathematically modeling data preprocessing, STL decomposition, trigonometric transformations, and rolling statistics, the model achieved low error rates and high R^2 values in training. Although test performance varied across districts—with some indications of overfitting, the results provide a solid foundation for future improvements. The integration of physical and statistical features within a deep learning framework demonstrates considerable potential for practical applications in wind energy production and meteorological forecasting. The findings highlight the importance of careful data preprocessing and the selection of robust model components, while also underscoring the challenges of generalization across diverse geographical regions. Future research will focus on mitigating overfitting in challenging districts and conducting more detailed comparative analyses with baseline models and ablation studies.

6. REFERENCES

[1] Polat S, Alpaslan N, Hallaç İR. Wind Speed Prediction Using Meteorological Measurements for Elazığ Province. Computer Science 2023; Vol:8: 110–120.

[2] Arabi S, Asgarimehr M, Kada M, et al. Hybrid CNN-LSTM Deep Learning for Track-Wise GNSS-R Ocean Wind Speed Retrieval. Remote Sensing 2023, Vol 15, Page 4169 2023; 15: 4169.

[3] Chen G, Li L, Zhang Z, et al. Short-term wind speed forecasting with principle-subordinate predictor based on Conv-LSTM and improved BPNN. IEEE Access 2020; 8: 67955–67973.

[4] Li X, Li K, Shen S, et al. Exploring Time Series Models for Wind Speed Forecasting: A Comparative Analysis. Energies 2023, Vol 16, Page 7785 2023; 16: 7785.

[5] Zuluaga CD, Álvarez MA, Giraldo E. Short-term wind speed prediction based on robust Kalman filtering: An expe rimental comparison. Applied Energy 2015; 156: 321–330.

[6] Lin KP, Pai PF, Ting YJ. Deep belief networks with genetic algorithms in forecasting wind speed. IEEE Access 2019; 7: 99244–99253.

[7] Zhu Q, Chen J, Zhu L, et al. Wind Speed Prediction with Spatio–Temporal Correlation: A Deep Learning Approach. Energies 2018, Vol 11, Page 705 2018; 11: 705.

[8] Chen M-Y, Liou Y-F, Chien H. Applications of deep-learning on TRITON data: results and findings. https://doi.org/101117/123042757 2025; 13268: 6–8.

[9] Chen X, Wang Y, Zhang H, et al. A novel hybrid forecasting model with feature selection and deep learning for wind speed research. Journal of Forecasting 2024; 43: 1682–1705.

[10] Feng L, Wang Y, Yan Y, et al. Wind Speed Prediction Model Based on Deep Learning. E3S Web of Conferences 2023; 466: 01011.

[11] Shi X, Lei X, Huang Q, et al. Hourly Day-Ahead Wind Power Prediction Using the Hybrid Model of Variational Model Decomposition and Long Short-Term Memory. Energies 2018, Vol 11, Page 3227 2018; 11: 3227.

[12] Hossain MA, Gray EMA, Islam MR, et al. Forecasting very short-term wind power generation using deep learning, optimization and data decomposition techniques. ICEMS 2021 - 2021 24th International Conference on Electrical Machines and Systems 2021; 323–327.

[13] Imani M, Fakour H, Lan WH, et al. Application of Rough and Fuzzy Set Theory for Prediction of Stochastic Wind Speed Data Using Long Short-Term Memory. Atmosphere 2021, Vol 12, Page 924 2021; 12: 924.

[14] Dolatabadi A, Abdeltawab H, Mohamed YARI. Hybrid Deep Learning-Based Model for Wind Speed Forecasting Based on DWPT and Bidirectional LSTM Network. IEEE Access 2020; 8: 229219–229232.

[15] Li Y, Chen X, Li C, et al. A Hybrid Deep Interval Prediction Model for Wind Speed Forecasting. IEEE Access 2021; 9: 7323–7335.

[16] Di Piazza A, Di Piazza MC, Vitale G. Estimation and Forecast of Wind Power Generation by FTDNN and NARX-net based models for Energy Management Purpose in Smart Grids. RE&PQJ 2014; 12: 995–1000.

[17] Lawal A, Rehman S, Alhems LM, et al. Wind Speed Prediction Using Hybrid 1D CNN and BLSTM Network. IEEE Access 2021; 9: 156672–156679.

[18] Li Y, Chen X, Li C, et al. A Hybrid Deep Interval Prediction Model for Wind Speed Forecasting. IEEE Access 2021; 9: 7323–7335.

[19] Mohandes M, Rehman S, Nuha H, et al. Accuracy of wind speed predictability with heights using

Recurrent Neural networks. FME Transactions 2021; 49: 908–918.
[20] Tuerxun W, Xu C, Guo H, et al. An ultra-short-term wind speed prediction model using LSTM based on modified tuna swarm optimization and successive variational mode decomposition. Energy Science & Engineering 2022; 10: 3001–3022.
[21] Cheng L, Zang H, Ding T, et al. Ensemble Recurrent Neural Network Based Probabilistic Wind Speed Forecasting Approach. Energies 2018, Vol 11, Page 1958 2018; 11: 1958.
[22] Xu X, Ma S, Huang C. Data denoising and deep learning prediction for the wind speed based on NOA optimization. Epub ahead of print 2 August 2024. DOI: 10.21203/RS.3.RS-4699260/V1.
[23] Shirzadi N, Nasiri F, Menon RP, et al. Smart Urban Wind Power Forecasting: Integrating Weibull Distribution, Recurrent Neural Networks, and Numerical Weather Prediction. Energies 2023, Vol 16, Page 6208 2023; 16: 6208.
[24] García-Puente B, Rodríguez-Hurtado A, Santos M, Sierra-García JE. Evaluation of XGBoost vs. other Machine Learning models for wind parameters identification. Renew Energy Power Qual J [Internet]. 2023 Jul;21(1):388–93. Available from: https://www.icrepq.com/icrepq23/334-23-garcia.pdf
[25] Li X, Li K, Shen S, Tian Y. Exploring Time Series Models for Wind Speed Forecasting: A Comparative Analysis. Energies 2023, Vol 16, Page 7785 [Internet]. 2023 Nov 27 [cited 2025 Mar 9];16(23):7785. Available from: https://www.mdpi.com/1996-1073/16/23/7785/htm
[26] Uçar M, İncetaş MO, Bayraktar I, Çilli M. Using Machine Learning Algorithms for Jumping Distance Prediction of Male Long Jumpers. J Intell Syst Theory Appl [Internet]. 2022 Sep 1;5(2):145–52. Available from: https://dergipark.org.tr/en/doi/10.38016/jista.1078474