1. Abderrahim BAKIR, 2. Abdelkader RAMI, 3. Riyadh BOUDDOU

ORCID: 1. 0000-0003-2880-7589; 2. 0000-0001-9599-6178; 3. 0000-0001-6555-5785

OPEN ACCESS

DOI: 10.15199/48.2025.05.46

A novel optimized hybrid deep neural network for accurate short-term electricity price forecasting in smart grids

Nowatorska hybrydowa sieć neuronowa optymalizowana do dokładnego prognozowania krótkoterminowych cen energii elektrycznej w inteligentnych sieciach.

Abstract. *Accurate short-term electricity price forecasting (ST-EPF) is critical for managing smart grids and ensuring efficient market operations. This paper presents a novel Hybrid Deep Neural Network (HDNN) model that combines five powerful architectures: CNN, LSTM, BiLSTM, GRU, and MLP. This model is specifically designed to handle the complex spatial and temporal patterns inherent in electricity price data. Using Bayesian optimization, the HDNN adapts its structure and fine-tunes its hyperparameters to best fit the unique characteristics of the dataset. We tested our model on two real-time datasets from ISO New England's WCMA and RI zones, where it consistently outperformed traditional machine learning methods, standalone deep learning models, and other hybrid variants on key metrics: MAPE, MAE, RMSE, and R². The results highlight the model's exceptional accuracy and adaptability, making it a valuable tool for electricity market stakeholders. Its dynamic optimization and flexible design provide a solid framework for future smart grid forecasting applications*

Streszczenie. *Dokładne prognozowanie cen energii elektrycznej w krótkim okresie (ST-EPF) jest kluczowe dla zarządzania inteligentnymi sieciami i zapewnienia efektywnego funkcjonowania rynków. W niniejszym artykule przedstawiono nowy model Hybrydowej Głębokiej Sieci Neuronowej (HDNN), który łączy pięć zaawansowanych architektur: CNN, LSTM, BiLSTM, GRU oraz MLP. Model ten został zaprojektowany specjalnie do analizy złożonych wzorców przestrzennych i czasowych, które są nieodłączne w danych dotyczących cen energii elektrycznej. Dzięki optymalizacji Bayesowskiej, HDNN dostosowuje swoją strukturę i precyzyjnie reguluje hiperparametry, aby najlepiej dopasować się do unikalnych cech zbioru danych. Nasz model został przetestowany na dwóch zbiorach danych w czasie rzeczywistym z regionów WCMA i RI organizacji ISO New England, gdzie konsekwentnie przewyższał tradycyjne metody uczenia maszynowego, samodzielne modele głębokiego uczenia oraz inne hybrydowe warianty w kluczowych metrykach: MAPE, MAE, RMSE i R². Wyniki podkreślają wyjątkową dokładność i elastyczność modelu, czyniąc go cennym narzędziem dla uczestników rynku energii. Jego dynamiczna optymalizacja i elastyczna konstrukcja stanowią solidne ramy dla przyszłych zastosowań w prognozowaniu dla inteligentnych sieci*

Keywords: Hybrid Deep Neural Network, Short-Term Electricity Price Forecasting, Smart Grid, Hyperparameter Optimization.
Słowa kluczowe: Hybrydowa Głęboka Sieć Neuronowa, Krótkoterminowe Prognozowanie Ceny Energii Elektrycznej, Inteligentna Sieć, Optymalizacja Hiperparametrów.

Introduction

Electricity Price Forecasting (EPF) is crucial for managing modern smart grids, which are increasingly complex and integrate renewable energy sources [1]. Accurate EPF impacts bidding strategies, market efficiency, and risk management, while also supporting grid stability and resource optimization. EPF faces challenges due to the volatile, non-stationary nature of electricity prices, driven by factors like fluctuating demand, renewable energy generation, weather, regulations, and disturbances [2], [3]. Traditional statistical methods often struggle to capture the complex temporal and spatial dependencies in price data. Recently, Machine Learning (ML) and Deep Learning (DL) techniques have gained prominence, offering better performance in addressing these complexities. However, due to the complexity of electricity prices, hybrid DL models have been developed, combining the strengths of individual models to improve accuracy, robustness, and adaptability in forecasting [4].

According to several studies [5], hybrid DL models for EPF can be broadly classified in the literature into two main types: Enhanced individual models and fusion models. Enhanced individual models focus on improving the performance of a single DL model through various techniques such as data decomposition, feature selection (FS), and advanced architectures such as encoder-decoder frameworks.

Shao et al. (2021) [6] developed a hybrid model integrating Ensemble Empirical Mode Decomposition (EEMD), Max-Dependency Min-Redundancy, and BiLSTM for EPF, significantly improving accuracy and adaptability across various markets. Pourdaryaei et al. (2024) [1] introduced a hybrid model using multi-head self-attention and CNN, along with mutual information-based FS, to reduce complexity and enhance short-term EPF accuracy. Ghimire et al. (2024) [7] proposed a hybrid approach combining CNN and Random Vector Functional Link models for half-hourly EPF, demonstrating superior performance across seasonal conditions. Wang et al. (2023) [8] presented a BiLSTM model incorporating similarity day screening and advanced decomposition, showing better accuracy and volatility handling than traditional models. Chughtata et al. (2023) [9] introduced a GRU model with EEMD and FS for short-term EPF, effectively capturing temporal relationships and improving prediction accuracy. However, these models face challenges: integrating decomposition and FS makes them complex and resource-intensive. They are prone to overfitting, excelling on training data but underperforming on unseen data. Scalability and adaptability also become difficult with new datasets or changing conditions.

Fusion models, also known as Hybrid Deep Neural Networks (HDNNs) involve the hybridization of two different DL architectures to leverage the complementary strengths of each. Shao et al. (2022) [10] introduced an HDNN architecture combining Multi-Head Self-Attention (MHSA), nested LSTM (NLSTM), and CNN. This model captures global and long-term dependencies for interval EPF, outperforming traditional methods and enhancing the effectiveness of FS and pattern classification. Mubarak et al. (2024) [11] proposed a hybrid CNN-BiLSTM-Autoregressive (AR) model. The CNN extracts spatial features, BiLSTM captures temporal dependencies, and the AR component addresses transient linear patterns. Hyperparameter tuning, including Particle Swarm Optimization, significantly boosts EPF accuracy in UK and German markets. Shejul et al. (2024) [12] developed an HDNN model for day-ahead EPF. It combines Exponential Smoothing for capturing seasonality with CNN-LSTM to model spatial and temporal dependencies, outperforming standalone models in the Indian Energy Exchange. Kim et al. (2023) [13] presented a CNN-BiLSTM model for electricity demand and system marginal price forecasting. The 1D-CNN extracts spatial features, while the BiLSTM captures bidirectional temporal patterns, validated using public data from Jeju Island. However, these models face challenges: fusion typically involves only two networks, such as CNN and LSTM, with little exploration of more complex combinations. This limits the potential of the model by missing synergies from additional architectures. In addition, many studies lack thorough hyperparameter tuning, overlooking key factors such as network depth, architecture settings, and training configurations (e.g., solver, activation, batch size, learning rate). This limits performance and raises concerns about robustness and reproducibility. Comprehensive hyperparameter optimization is critical to maximizing the effectiveness of HDNN.

This paper introduces a novel HDNN model for short-term EPF. Unlike conventional models, the proposed HDNN integrates five architectures (CNN, LSTM, BiLSTM, GRU, and MLP) to leverage spatial and temporal features and long-term sequence modeling. The architecture and hyperparameters are optimized using Bayesian Optimization (BO), ensuring each configuration includes at least two network types. This adaptive approach enhances the model's ability to capture complex price patterns, significantly improving forecasting accuracy across various datasets.

Convolutional Neural Network

Originally developed for image data, CNNs have been effectively adapted for time series prediction and excel at capturing spatial and temporal patterns through local dependencies and hierarchical structures. Key components include: 1) Convolutional Layers: These layers use filters (kernels) applied through sliding windows to extract local features such as edges or patterns. 2) Pooling Layers: Typically using maximum or average pooling, these layers reduce spatial dimensions, lowering computational cost while preserving essential information. 3) Fully Connected Layers: After feature extraction, these dense layers are used to map the learned features to the target variable, effectively capturing complex relationships for regression tasks. Fig. 1. depicts the CNN architecture.

Long Short-Term Memory

LSTMs address the limitations of traditional Recurrent Neural Networks, specifically the vanishing/exploding gradient problems, through a gating mechanism that helps preserve long-term dependencies in sequential data. Main components include: 1) Memory Cell: Stores information across time steps for future use. 2) Forget Gate: Controls the amount of previous memory that is discarded, allowing the network to forget irrelevant information. 3) Input Gate: Controls how much new data is added to memory. 4) Output Gate: Determines which information from memory is sent to the next step or output layer. Fig. 2. shows the LSTM structure.

Bidirectional Long Short-Term Memory

BiLSTMs improve upon standard LSTM networks by processing input data in both forward and backward directions, enhancing the network's ability to capture both past and future dependencies in sequences. Key components include: 1) Forward LSTM: Processes sequences from the first to the last time step. 2) Backward LSTM: Processes sequences in reverse order. 3) Concatenation Layer: Merges outputs from both forward and backward passes. The BiLSTM architecture is shown in Fig. 3.

Gated Recurrent Unit

GRUs are a streamlined version of LSTMs, designed for faster training and computation with fewer parameters, while

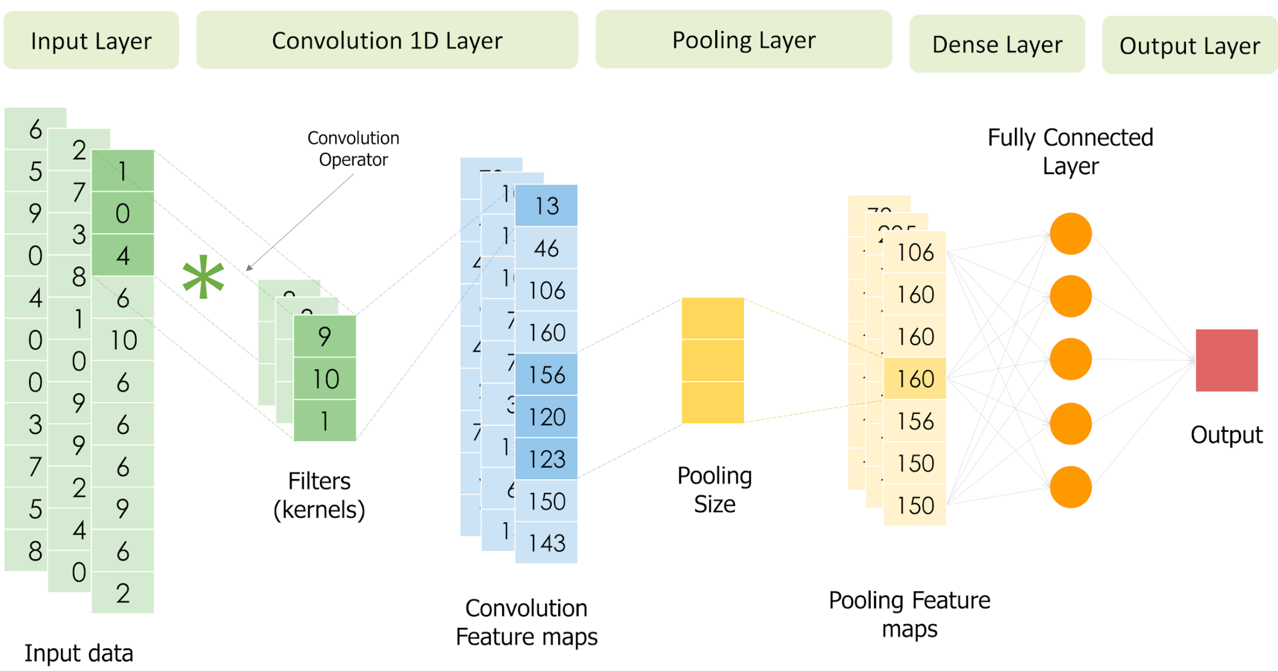

Fig. 1. Convolutional Neural Network (CNN) architecture

still effectively handling sequential data. The key components are: 1) Update Gate: Merges the roles of the LSTM's Forget and Input gates to control the retention and updating of information across time steps. 2) Reset Gate: Manages the influence of previous states on the current state, allowing the model to focus on relevant data. Fig. 4. depicts the GRU model.

Multi-Layer Perceptron

MLPs are fundamental Artificial Neural Networks that consist of fully connected layers and are used for a variety of tasks, including forecasting. Although simple, they can approximate complex relationships between inputs and outputs. Key components include: 1) Input Layer: Receives the raw input data. 2) Hidden Layers: Includes neurons fully connected to previous and next layers, capturing patterns through weighted sums and activation functions. 3) Output Layer: Produces the final predictions. 4) Activation Functions: Introduce nonlinearity, allowing the network to model more complex relationships. The MLP network is illustrated in Fig. 5.

Proposed Hybrid Deep Neural Network

The architecture of our hybrid model, as illustrated in Fig. 6., consists of a sequential combination of five types of neural networks, with three layers of each network type, as follows:

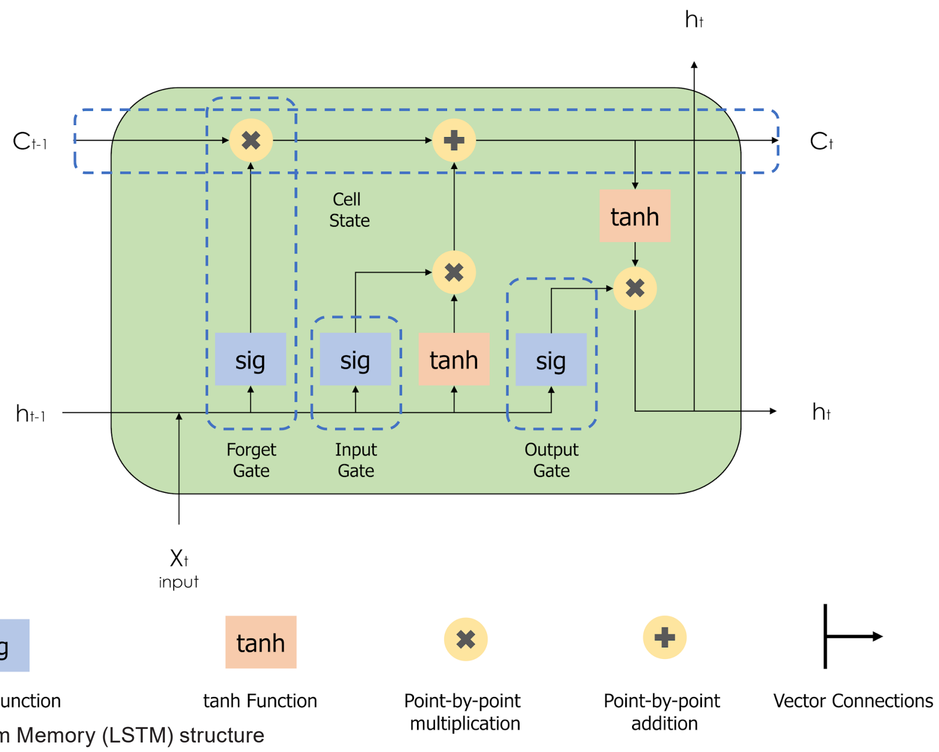

Fig. 2. Long Short-Term Memory (LSTM) structure

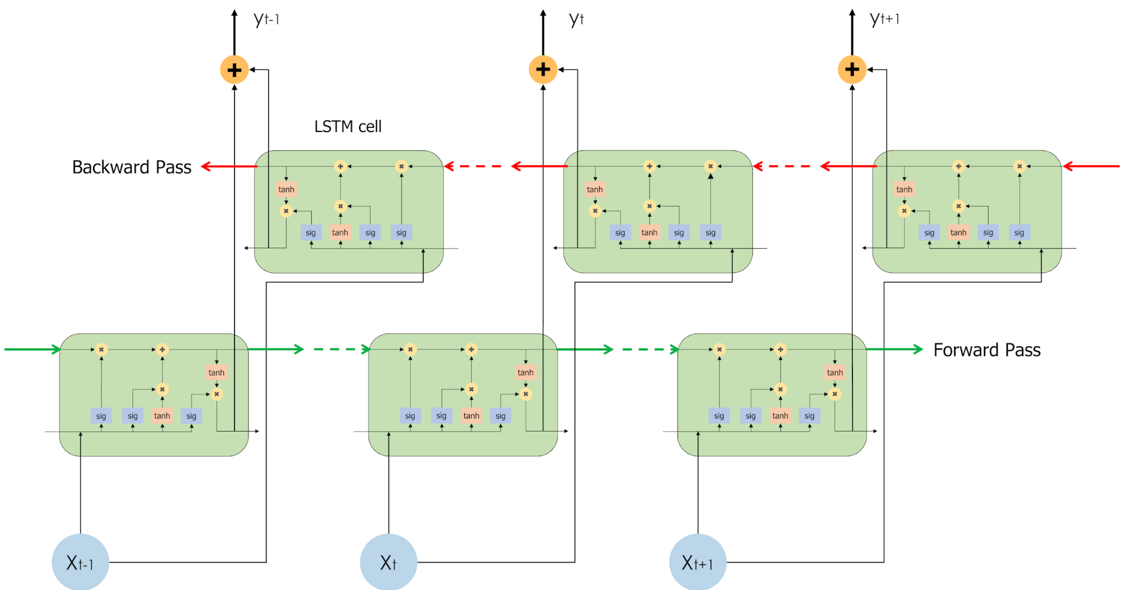

Fig. 3. Bidirectional LSTM (BiLSTM) architecture

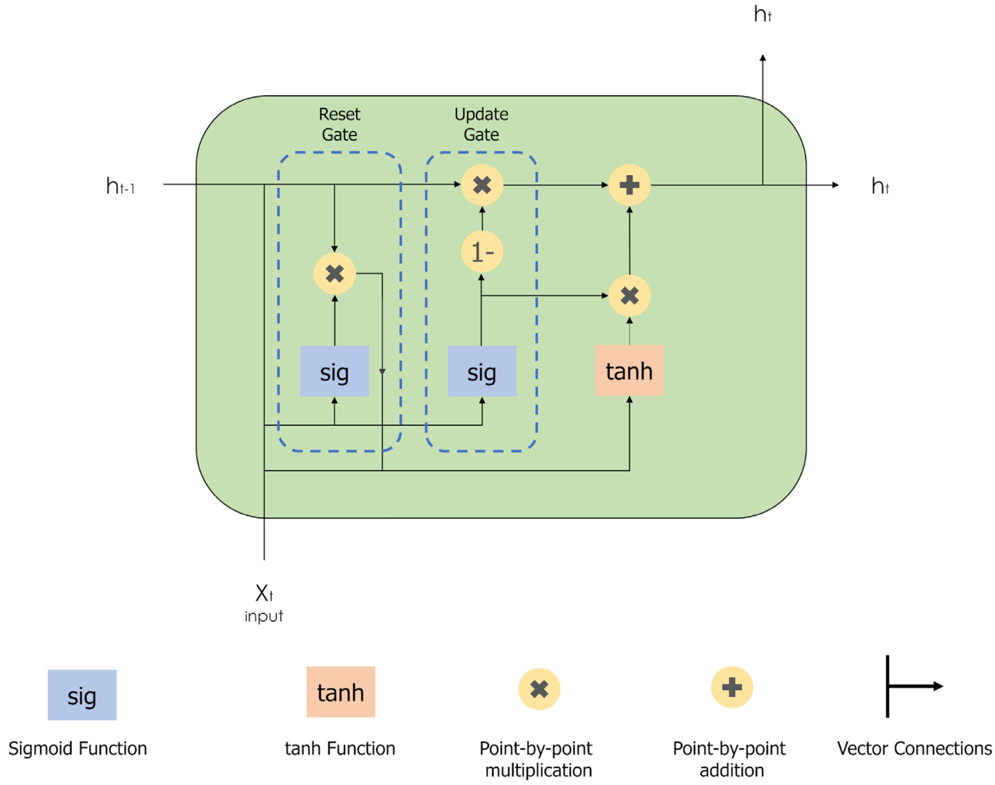

Fig. 4. Gated Recurrent Unit (GRU) model

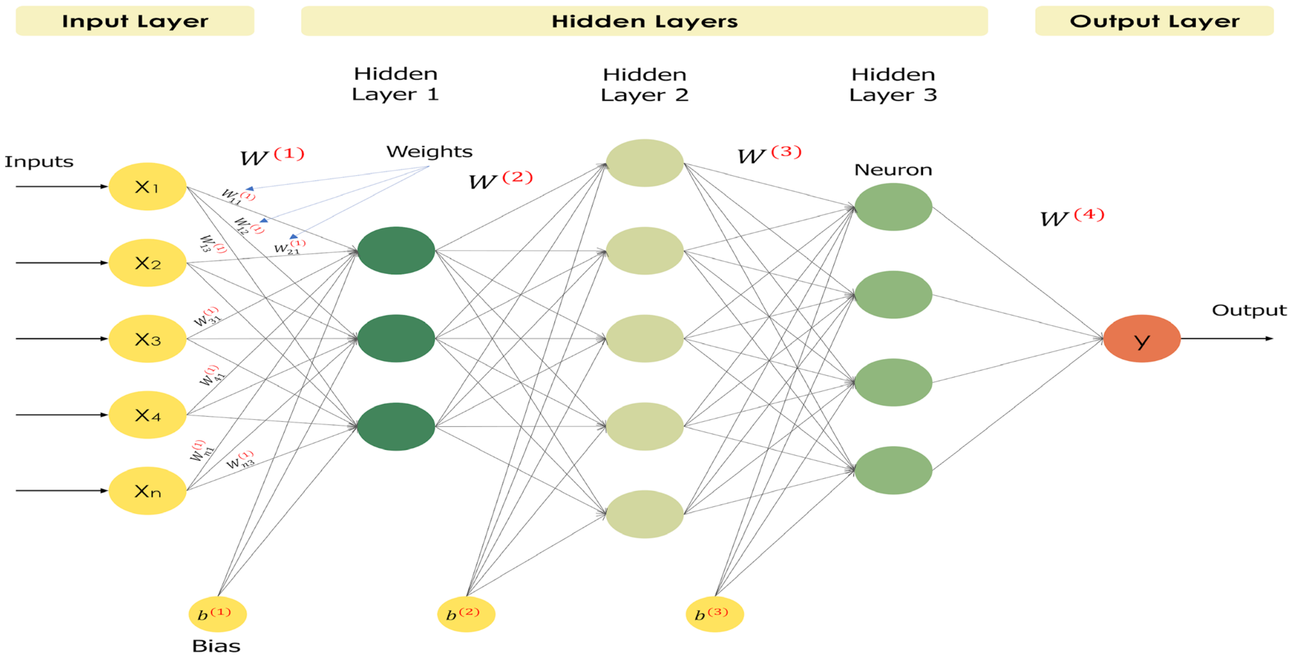

Fig. 5. Multi-Layer Perceptron (MLP) network

Input Layer: A sequence input layer is used to ingest the time series data. It applies normalization techniques to stabilize training and ensure consistent scaling across inputs.

Convolutional Layers: Initial Convolutional Block: The first block consists of a 1D convolutional layer followed by an activation function, layer normalization, and max pooling. It focuses on detecting local patterns and reducing input dimensionality. Subsequent Convolutional Blocks: Two additional convolutional layers refine the feature extraction using larger filters and more complex operations. The final convolutional block applies global average pooling to summarize the feature maps and prepare them for the next layers.

Recurrent Layers: LSTM Layers: A series of three LSTM layers are employed to capture temporal patterns and long--term dependencies within the sequential data, ensuring the network retains important information over time. BiLSTM Layers: Three BiLSTM layers are used to enhance the model's understanding of context by processing the data in both for-

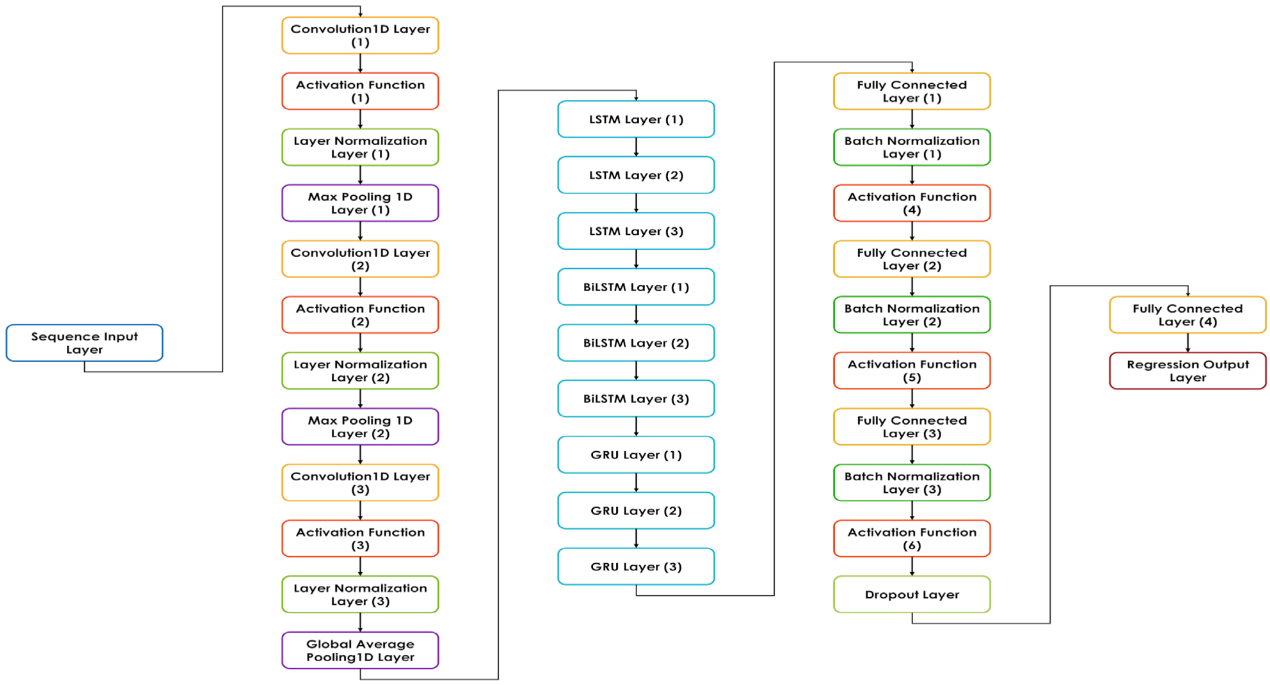

Fig. 6. Layer architecture of the proposed HDNN

ward and backward directions, allowing it to capture dependencies from both past and future time steps. GRU Layers: Three GRU layers provide a more computationally efficient mechanism for handling sequential data, complementing the LSTM and BiLSTM layers.

Fully Connected (MLP) Layers: Three fully connected layers, interspersed with batch normalization, activation functions, and dropout, integrate the features learned from the previous layers. These layers prepare the data for the final prediction.

Output Layer: A final fully connected layer of size one produces the forecasted values. While a regression layer computes the loss during training to guide the model's learning.

1) *Mathematical Foundations*

Convolutional Layer: For a 1D convolutional layer, the output at time step *t*, denoted as *y*[*t*], is computed as:

(1) $$y[t]=\sum_{k=0}^{K-1} W[k]\cdot x[t-k]+b$$

Where: x – input sequence, W – convolution filter, b – bias term, K – filter size.

LSTM Layer: The operations of an LSTM layer can be described by the following equations: Forget gate (2), Input gate (3), Cell state (4), Updated cell state (5), Output gate (6), and Hidden state (7):

(2) $$f_t=\sigma\left(W_f\cdot[h_{t-1},x_t]+b_f\right)$$

(3) $$i_t=\sigma\left(W_i\cdot[h_{t-1},x_t]+b_i\right)$$

(4) $$\tilde{C}_t=\tanh\left(W_C\cdot[h_{t-1},x_t]+b_C\right)$$

(5) $$C_t=f_t\cdot C_{t-1}+i_t\cdot\tilde{C}_t$$

(6) $$o_t=\sigma\left(W_o\cdot[h_{t-1},x_t]+b_o\right)$$

(7) $$h_t=o_t\cdot\tanh(C_t)$$

Where: $[h_{t-1},x_t]$ – concatenation of the previous hidden state and current input. W_f,W_i,W_o,W_C and b_f,b_i,b_o,b_C – weight matrices and biases for each gate (forget, input, output) and cell state. σ – sigmoid activation function, f_t,i_t,o_t – hyperbolic tangent function, $C_t,C_{t-1},\tilde{C}_t$ – forget, input, and output gate activations, respectively. h_t – current, previous, and candidate cell states, respectively. – current hidden state.

BiLSTM Layer: A BiLSTM processes the input sequence in both directions, Forward pass (8), Backward pass (9), and Final hidden state (10):

(8) $$h_t^{\rightarrow}=LSTM\left(x_t,h_{t-1}^{\rightarrow}\right)$$

(9) $$h_t^{\leftarrow}=LSTM\left(x_t,h_{t+1}^{\leftarrow}\right)$$

(10) $$h_t=\left[h_t^{\rightarrow},h_t^{\leftarrow}\right]$$

Where: $h_t^{\rightarrow},h_t^{\leftarrow}$ – hidden state from the forward and backward LSTM. $h_{t-1}^{\rightarrow},h_{t+1}^{\leftarrow}$ – previous and next hidden states for the forward and backward passes, respectively.

GRU Layer: For a GRU layer, the operations are given by the following equations: Update gate (11), Reset gate (12), Candidate hidden state (13), and Final hidden state (14).

(11) $$z_t=\sigma\left(W_z\cdot[h_{t-1},x_t]+b_z\right)$$

(12) $$r_t=\sigma\left(W_r\cdot[h_{t-1},x_t]+b_r\right)$$

(13) $$\tilde{h}_t = \tanh\left(W_h \cdot \left[r_t \cdot h_{t-1}, x_t\right] + b_h\right)$$

(14) $$h_t = \left(1 - z_t\right) \cdot h_{t-1} + z_t \cdot \tilde{h}_t$$

Where: z_t – update gate activation vector, z_t – reset gate activation vector, $\tilde{h}_t$ – candidate hidden state. W_z, W_r, W_h – weight matrices for the update, reset, and candidate hidden states, respectively. b_z, b_r, b_h – corresponding biases.

Fully Connected Layer: For a fully connected (dense) layer, the transformation is described as:

(15) $$y = W \cdot x + b$$

2) Hyperparameter Optimization

The primary objectives of hyperparameter optimization for this hybrid model are: 1. *Select the Optimal Hybrid Architecture:* Dynamically construct a hybrid model that includes only the most beneficial network types, with at least two different types always included. 2. *Determine Optimal Depth:* Identify the best number of layers for each network type (CNN, LSTM, BiLSTM, GRU, and MLP). 3. *Optimize Network-Specific and Training Hyperparameters:* Fine-tune additional hyperparameters specific to each network type as well as general training parameters that affect the entire architecture. This approach minimizes unnecessary complexity, excludes redundant networks, and prioritizes the most impactful layers for the forecasting task.

A. Key Hyperparameters

The optimization process targets a comprehensive set of hyperparameters, grouped into three categories:

Architecture and Depth Hyperparameters: Control the depth and inclusion of each network type: Hp1: CNN depth (ranges from 0 to 3); Hp2: LSTM depth (0 to 3); Hp3: BiLSTM depth (0 to 3); Hp4: GRU depth (0 to 3); Hp5: MLP depth (1 to 3, with Hp5 = 1 indicating only a dense layer is added). The novelty lies in how these hyperparameters not only define the number of layers but also control whether specific networks (CNN, LSTM, BiLSTM, GRU, or MLP) are included in the architecture. For instance, setting Hp1 = 0 excludes CNN layers from the architecture entirely. Similarly, setting Hp5 = 1 excludes MLP layers but ensures a dense layer is added as a final layer.

Network-Specific Hyperparameters: Tailored hyperparameters for each network type: Hp6, Hp7, Hp8 for CNN: Filter size (ranges from 3 to 7), Number of filters (10 to 100), and Pooling size (2, 3). Hp9, Hp10, and Hp11 for LSTM, BiLSTM, and GRU respectively: Number of hidden units (ranges from 10 to 300). Hp12, and Hp13 for MLP: Number of neurons in fully connected layers (ranges from 10 to 300), and Dropout probability (ranges from 0.2 to 0.5). These hyperparameters are excluded if their corresponding network type is not part of the architecture.

Training Hyperparameters: These hyperparameters govern the learning process of the entire network, influencing how the model updates its weights, how quickly it converges, and how well it generalizes: Hp14: Solver (adam, sgdm, rmsprop); Hp15: Activation function (relu, leaky-relu, elu, swish, tanh); Hp16: Mini-batch size (128, 256, 512, 1024); Hp17: Maximum epochs (50, 100, 150, 200); Hp18: Initial learning rate (0.001 to 0.009); Hp19: L2 regularization (0.0001 to 0.0009); Hp20: Momentum (0.1 to 1); Hp21: Gradient decay factor (0 to 0.99); Hp22: Squared gradient decay factor (0 to 0.99). Some training hyperparameters are solver-dependent and are conditionally set based on the selected solver.

B. Optimization Method

The optimization process employs BO with the Expected Improvement Plus (EI+) acquisition function. This approach effectively balances exploration and exploitation, enabling the optimizer to explore promising regions of the search space while efficiently converging on the optimal solution. The optimization process involves: 1. *Define the Search Space:* The search space is defined by the ranges of the hyperparameters, creating a complex landscape of potential architectures. BO is guided by the Root Mean Squared Error (RMSE) objective function, seeking to minimize the model's error. 2. *Constraint Enforcement:* A critical constraint ensures that the architecture includes at least two types of networks. Architectures that violate this constraint (i.e., those that include only one type of network or no networks) are excluded from the search, with the code describing this constraint shown in Fig. 7.

```
if sum([Hp1>0, Hp2>0, Hp3>0, Hp4>0, Hp5>1])<2
    % Skip the trial; architecture is not valid
    rmse = inf ;
end
```

Fig. 7. MATLAB code for enforcing two-network minimum constraint

3. Dynamic Model Construction: For each hyperparameter configuration, a corresponding hybrid neural network is constructed dynamically. Layers are added based on the inclusion criteria defined by Hp1 through Hp5, that if any depth hyperparameter is set to zero, the corresponding network is excluded from the architecture, and network--specific hyperparameters are applied only to included networks. 4. *Evaluation and Objective Function:* Each model is trained and evaluated using RMSE as the objective function. A 5-fold cross-validation is used at each iteration, ensuring robust performance evaluation with 4 folds for training and 1 fold for validation. 5. *Search Strategy and Iterative Refinement:* The optimization runs for a maximum of 30 iterations, progressively refining the hyperparameters. Each iteration improves the optimizer's understanding of the hyperparameter space, gradually converging on configurations that minimize RMSE.

Data Description

To evaluate the hybrid network, two datasets were used from the ISO New England Smart Grid [14], covering the Western/Central Massachusetts (WCMA) and Rhode Island (RI) zones. The data spanned from January 1, 2020, to December 31, 2021, for training and validation, with testing on data from January 1 to January 31, 2022.

1. Data preprocessing steps

Data cleaning and imputation: The data included historical day-ahead and real-time prices, demand, weather, and market clearing prices for WCMA and RI zones. Key variables such as real-time price, demand, temperature, and energy price component were extracted. Missing weekend values were imputed using a 5-hour moving median.

Variables Generation: New variables focused on historical values of electricity and fuel prices were created, including lagged same-hour, average, and peak electricity prices (1, 2, and 7 days prior) and lagged natural gas, fuel oil, and coal prices.

Feature Selection: Variables were selected based on their Pearson correlation with the electricity price [15], and those with a Pearson correlation coefficient (r) below 0.5 were excluded to enhance model performance. The input variables for each dataset used to train the proposed network are:

WCMA dataset (8 variables): V01, Real-time energy price [$/MWh]; V02, Same hour price (previous day) [$/MWh]; V03, Same hour price (one week ago) [$/MWh]; V04, Peak hour price (previous day) [$/MWh]; V05, Peak hour price (two days ago) [$/MWh]; V06, Peak hour price (one week ago) [$/MWh]; V07, Crude oil price [$/Barrel]; V08, Fuel oil price (previous day) [$/Gallon].

RI dataset (9 variables): V01, Real time energy price [$/MWh]; V02: Same hour price (previous day) [$/MWh]; V03, Same hour price (two days ago) [$/MWh]; V04, Same hour price (one week ago) [$/MWh]; V05, Average price (previous day) [$/MWh]; V06, Same hour price (previous day) [$/MWh]; V07, Same hour price (two days ago) [$/MWh]; V08, Natural gas price [$/MBtu]; V09, Natural gas price (previous day) [$/MBtu].

Performance evaluation

In this study, the predictive performance of the model is evaluated using four key metrics: Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), RMSE, and the Coefficient of Determination (R^2). Each metric provides unique insights into the model's accuracy and error characteristics [16]:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{16}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{17}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{18}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \times 100 \tag{19}$$

Where: n – number of observations, y_i – actual value, $\hat{y}_i$ – predicted value, and $\bar{y}$ – mean of actual values.

A good model has low MAPE, MAE, and RMSE, indicating minimal error and strong predictive performance. A high R^2 close to 100% indicates that the model explains a large portion of the variance in the data.

Results and Discussion

The models in this study were implemented in MATLAB 2021b and run on a system with an Intel® Core™ i5–6300U CPU and 8 GB of RAM.

1) Hyperparameter Optimization Results

Table 1 outlines the best hyperparameter configurations for WCMA and RI datasets using an HDNN for ST-EPF, highlighting key differences in their setups.

In terms of architecture, the WCMA model incorporates one layer each of CNN, LSTM, BiLSTM, and GRU, along with two MLP layers, emphasizing the need for thorough spatial and temporal feature extraction, as illustrated in Fig. 8. On the other hand, the RI model adopts a simpler structure with two GRU layers and three MLP layers, reflecting a reduced need for complex architectures, as shown in Fig. 9.

Table 1. Optimized hyperparameters for HDNN on WCMA and RI datasets

Best optimization results	HDNN for WCMA dataset	HDNN for RI dataset
Hp1	1	0
Hp2	1	0
Hp3	1	0
Hp4	1	2
Hp5	2	3
Hp6	4	–
Hp7	61	–
Hp8	–	–
Hp9	113	–
Hp10	202	–
Hp11	92	259
Hp12	109	262
Hp13	0.26286	0.45077
Hp14	adam	sgdm
Hp15	elu	leaky-relu
Hp16	1024	512
Hp17	200	150
Hp18	0.0027461	0.0074038
Hp19	0.00010609	0.00010064
Hp20	–	0.18009
Hp21	0.37502	–
Hp22	0.17148	–
RMSE [$/Mwh]	0.541	0.768
Training time [s]	285	255
Total optimization time [s]	8478	9248

WCMA's network-specific hyperparameters are more complex, with deeper layers and larger filters, to capture patterns in the dataset. The RI model uses GRU and MLP with more hidden units to handle long-term dependencies. Different training hyperparameters reflect the different characteristics of the datasets. WCMA uses the elu activation function, known for handling vanishing gradients in deep networks, while RI's leaky-relu activation helps handle sparse gradients and speeds up convergence. WCMA's adam optimizer supports adaptive learning suited to its complex structure, while RI uses sgdm for stable training with momentum. A larger mini-batch size for WCMA (1024 vs. 512) and a lower learning rate (0.0027461 vs. 0.0074038) highlight the need for stable and gradual optimization in the more complex WCMA model. Minimal L2 regularization is used in both models, with slightly higher values for WCMA to avoid overfitting. Performance

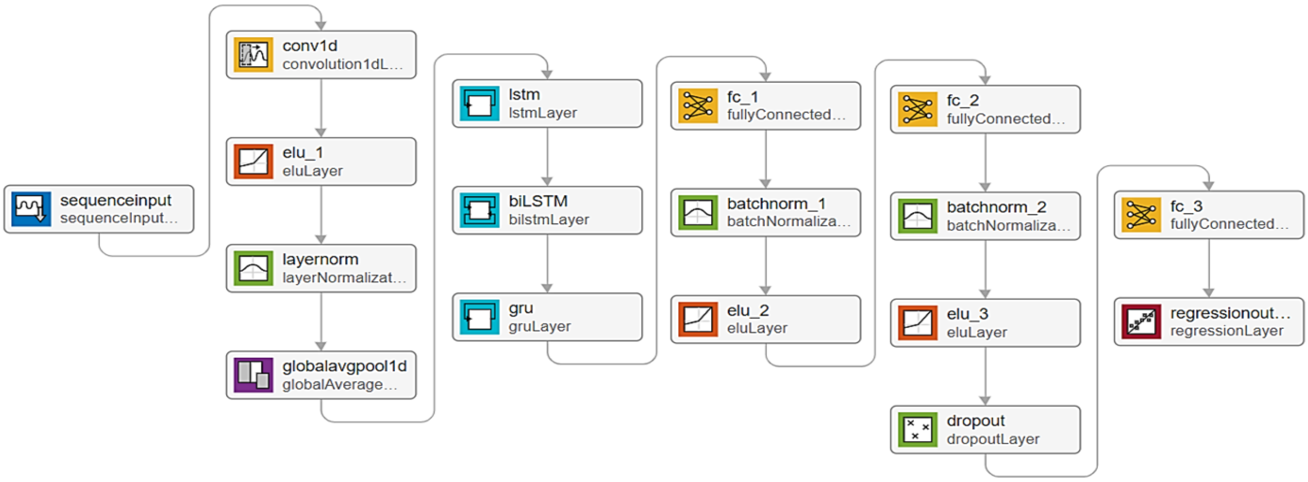

Fig. 8. Optimal model architecture for the WCMA dataset

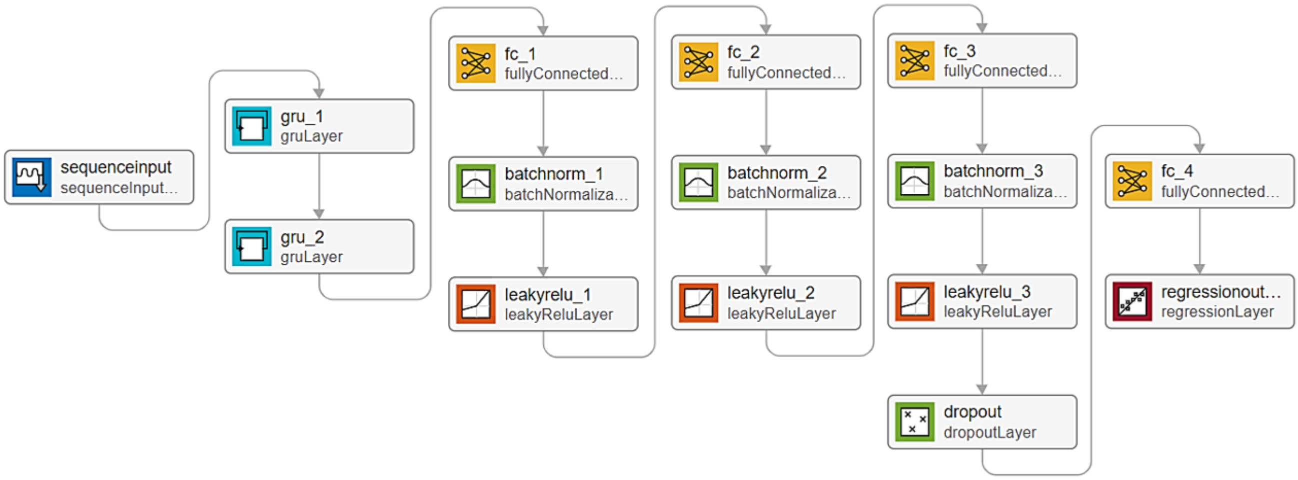

Fig. 9. Optimal model architecture for the RI dataset

metrics show that WCMA outperforms RI with a lower RMSE ($0.541/MWh vs. $0.768/MWh), demonstrating that WCMA's diverse architecture effectively captures complex spatial and temporal dependencies, resulting in superior forecast accuracy. The training convergence of the best models is shown in Fig. 10. and Fig. 11. for WCMA and RI datasets respectively. Despite its greater complexity, WCMA has a slightly shorter total optimization time (8478 seconds vs. 9248 seconds), indicating efficient hyperparameter tuning.

2) Short-Term Electricity Price Forecasting Results

To evaluate the performance of our optimized hybrid deep neural network, we compared it to three categories of models: *Machine Learning Models:* Regression Tree (RT), Random Forest (RF), and Support Vector Regression (SVR), all tuned using BO. *Deep Learning Models:* The proposed hybrid DNN model was compared with its individual network components (CNN, LSTM, BiLSTM, GRU, and MLP), each trained and optimized separately via BO. *Non-Optimized Hybrid Models:* The optimized hybrid model was also compared to three non-optimized versions with different layer architectures: HDNN_1 (1CNN-1LSTM-1BiLSTM-1GRU), HDNN_2 (2CNN-2LSTM-2BiLSTM-2GRU-2MLP), HDNN_3 (3CNN-3LSTM-3BiLSTM-3GRU-3MLP). These non-optimized models used standard hyperparameters from the literature.

A. WCMA Dataset

Table 2 shows that the optimized HDNN (1CNN-1LSTM-1BiLSTM-1GRU-2MLP) significantly outperforms all other models, achieving the lowest error rates: MAPE of 0.683%, MAE of $0.204/MWh and RMSE of $0.240/MWh, with the highest R^2 of 99.8%. This indicates excellent predictive accuracy and reliability. Compared to both individual ML/DL models and non-optimized hybrid variants, the optimized HDNN excels at capturing complex patterns, demonstrating superior tuning and architectural design.

Fig. 12. compares the actual (pink dashed line) and forecasted electricity prices of the proposed HDNN model (blue solid line) with ML models (RT, RF, SVR) for the WCMA dataset. The HDNN closely follows the actual price curve, with much less deviation than the ML models, which struggle especially during price peaks and troughs. Fig. 13. shows the actual vs. forecasted prices of the proposed HDNN against each DL model (CNN, LSTM, BiLSTM, GRU, MLP) for the WCMA dataset. The optimized HDNN provides forecasts that closely match the actual prices and outperform the individual DL models, especially in capturing sudden price changes. Fig. 14. compares the proposed HDNN with non-optimized HDNN variants (HDNN_1, HDNN_2, HDNN_3) for the WCMA dataset. The proposed HDNN provides the most accurate forecasts, while the non-

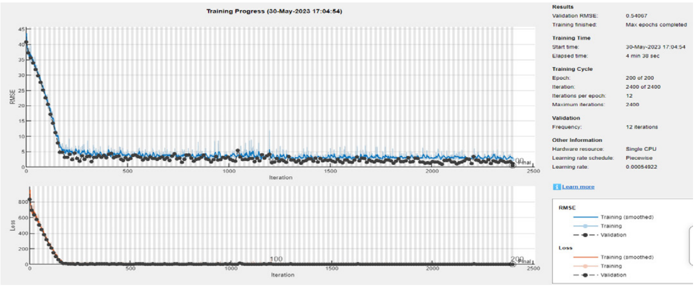

Fig. 10. Training convergence of the WCMA model

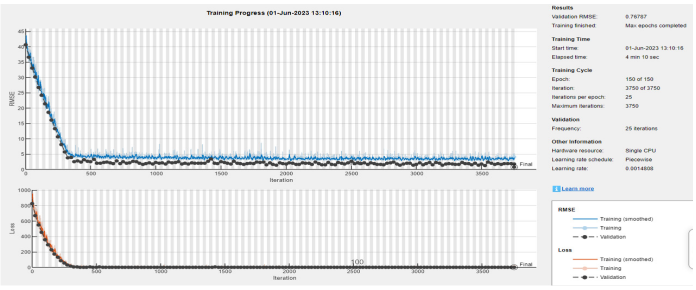

Fig. 11. Training convergence of the RI model

Our novel optimization strategy dynamically adapts the hybrid architecture to the specific characteristics of each dataset. For the WCMA dataset, the strategy selects a more complex architecture that incorporates different neural network layers to capture the intricate data patterns, resulting in improved performance. For the RI dataset, the strategy selects a simpler architecture that reflects the less complex data relationships while still ensuring an efficient and effective model. This adaptive approach ensures that the model's architecture is best suited to the needs of the dataset, improving overall predictive accuracy.

-optimized versions show larger errors, especially during volatile price movements, highlighting the impact of the optimization process.

Table 2. Forecasting performance comparison for WCMA dataset

Model	MAPE [%]	MAE [$/MWh]	RMSE [$/MWh]	R^2 [%]
RT	1.265	0.386	0.462	99.2
RF	4.513	1.147	1.609	89.8
SVR	1.339	0.382	0.438	99.2
LSTM	4.191	1.193	1.219	94.1
BiLSTM	1.238	0.341	0.362	99.5
GRU	4.683	1.330	1.347	92.9
CNN	3.564	1.028	1.031	95.8
MLP	1.667	0.487	0.489	99.1
HDNN_1	1.939	0.577	0.684	68.4
HDNN_2	2.829	0.798	0.877	97
HDNN_3	1.542	0.462	0.587	98.6
Optimized HDNN	**0.683**	**0.204**	**0.240**	**99.8**

B. RI Dataset

Table 3 indicates that the optimized HDNN (2GRU-3MLP) for the RI dataset shows excellent performance, with the lowest MAPE (0.668%), MAE ($0.210/MWh), and RMSE ($0.263/MWh), along with a high R² of 99.7%. This highlights the excellent accuracy and efficiency of the model compared

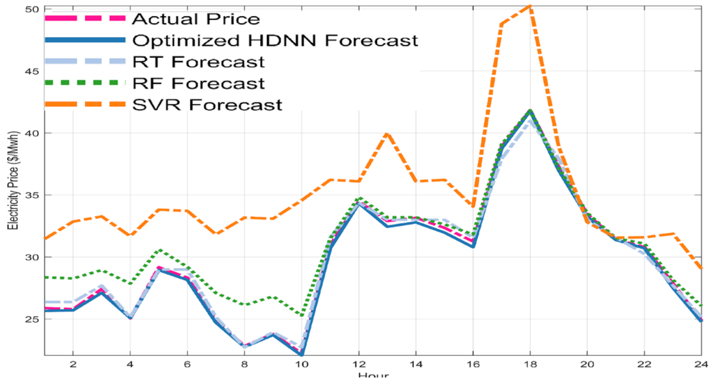

Fig. 12. Forecast accuracy of HDNN vs. ML models on WCMA dataset

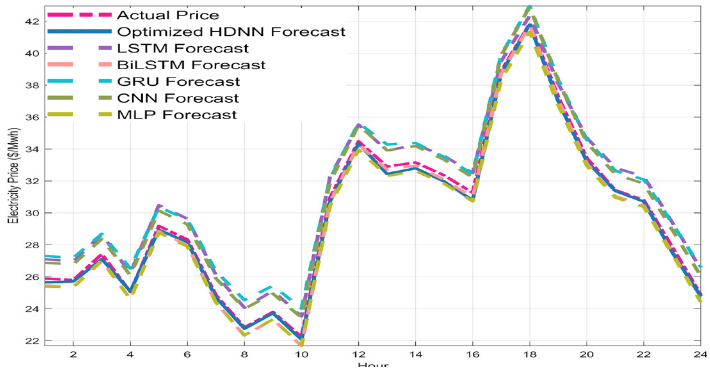

Fig. 13. Forecast accuracy of HDNN vs. individual DL models on WCMA dataset

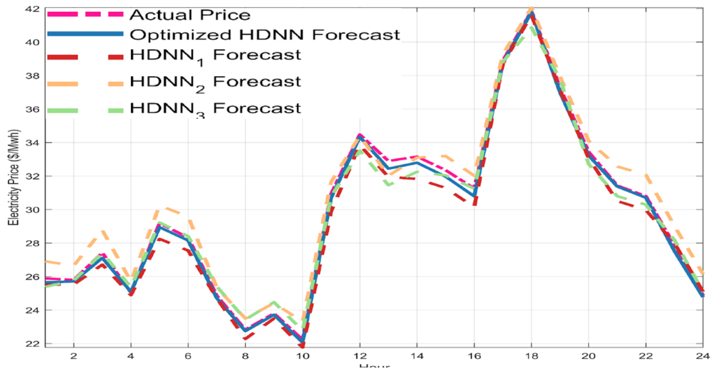

Fig. 14. Forecast accuracy of optimized vs. non-optimized HDNN on WCMA dataset

to all other ML and DL models, as well as the non-optimized hybrid versions. The results confirm that the optimized HDNN effectively captures the complex relationships in the data, making it the most reliable forecasting tool.

Figures 15, 16, and 17 compare actual electricity prices (pink dashed line) with predictions from the proposed HDNN (blue solid line) and other models using the RI dataset. In Fig. 15., the optimized HDNN more closely matches actual prices and outperforms traditional ML models. Fig. 16. highlights the superior performance of the optimized HDNN over individual DL models (e.g., CNN, LSTM, GRU), demonstrating the benefit of combining architectures. Fig. 17. shows that the proposed HDNN consistently tracks actual prices more accurately, proving the effectiveness of its optimized configuration over non-optimized variants.

Table 3. Forecasting performance comparison for RI dataset

Model	MAPE [%]	MAE [$/MWh]	RMSE [$/MWh]	R^2 [%]
RT	1.150	0.359	0.301	99.2
RF	2.210	0.633	0.801	97.4
SVR	4.349	1.270	1.276	93.3
LSTM	1.059	0.295	0.312	99.6
BiLSTM	2.041	0.573	0.593	98.6
GRU	1.285	0.385	0.432	99.2
CNN	0.924	0.290	0.355	99.5
MLP	1.313	0.390	0.410	99.3
HDNN_1	3.588	1.014	1.057	95.4
HDNN_2	2.414	0.659	0.729	97.8
HDNN_3	1.535	0.455	0.543	98.8
Optimized HDNN	**0.668**	**0.210**	**0.263**	**99.7**

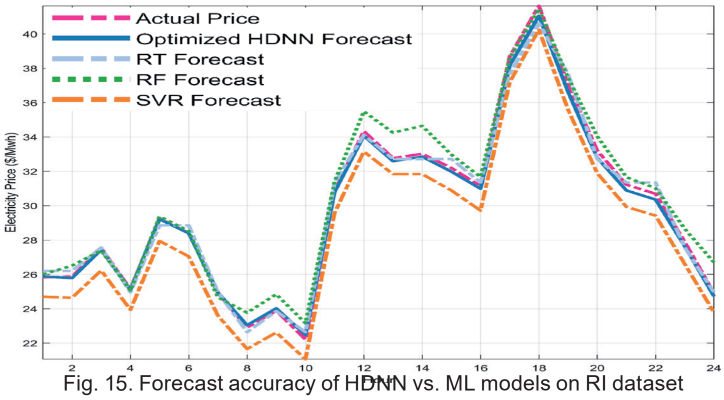

Fig. 15. Forecast accuracy of HDNN vs. ML models on RI dataset

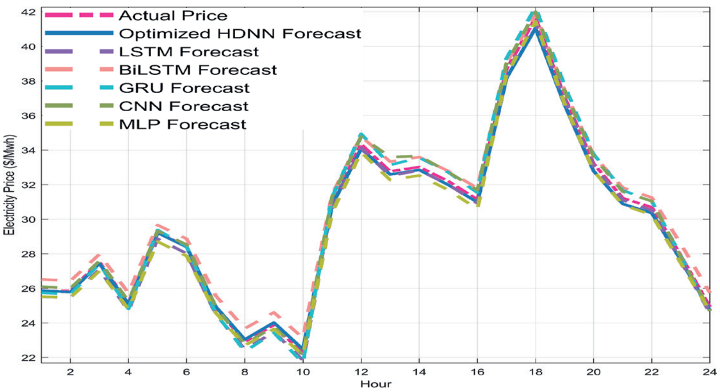

Fig. 16. Forecast accuracy of HDNN vs. individual DL models on RI dataset

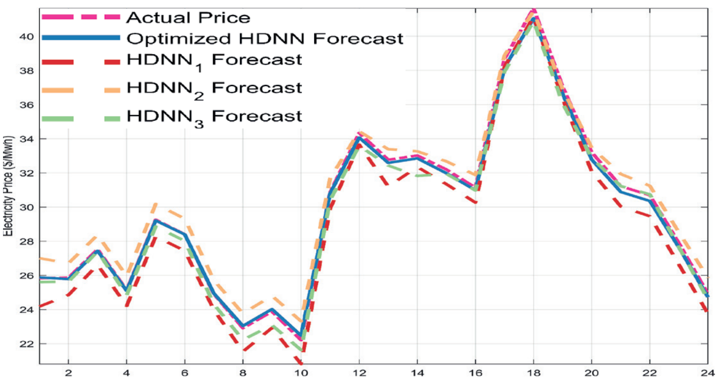

Fig. 17. Forecast accuracy of optimized vs. non-optimized HDNN on RI dataset

To evaluate the generalization capabilities of the proposed HDNN in accurately predicting electricity prices beyond the typical 24-hour window, we extended the evaluation to 31 days. The HDNN was tested against the benchmark models, each predicting electricity prices for the next 24 hours over 31 consecutive days for the two datasets (WCMA and RI). Forecasting errors (residuals) were calculated for all models, and residual plots were generated to evaluate model performance over the test period. The results show that the HDNN consistently outperforms other models across both datasets, demonstrating its superior forecasting accuracy.

Fig. 18. shows the residuals (prediction errors) of the proposed Hybrid Deep Neural Network (HDNN) and comparison models over the test period for the WCMA dataset. The residuals of the HDNN are smaller and more uniformly distributed around zero, indicating a lower prediction error compared to the other models, which exhibit larger and more erratic deviations. This consistent and minimal error suggests that the HDNN effectively captures the underlying patterns of the electricity price data, resulting in more accurate forecasts over the 31-day horizon, while other models exhibit less accuracy.

Fig. 19. illustrates the residuals (prediction errors) for the RI dataset, where the proposed HDNN exhibits tightly clustered residuals around zero, indicating lower error compared to other models that exhibit larger fluctuations. As with the WCMA dataset, the HDNN consistently outperforms the comparison models, exhibiting smaller and more stable residuals, underscoring its ability to maintain predictive accuracy and generalize effectively across different datasets.

Conclusion

In this study, we introduced a dynamic HDNN designed for ST-EPF. By combining the strengths of several advanced DL models such as CNN, LSTM, BiLSTM, GRU, and MLP, our model adapts to different datasets and optimizes its performance using BO. We tested this approach with real-world data from ISO New England's WCMA and RI zones, where the HDNN outperformed traditional ML models, standalone DL architectures, and even other unoptimized hybrid models. Our findings clearly show that: 1. The hybrid model improves forecasting accuracy by effectively capturing both spatial patterns and temporal changes in electricity prices. 2. Fine--tuning both the architecture and the training process through dynamic optimization is key to achieving the best results, regardless of the dataset. 3. The HDNN proved to be versatile, performing well in both single-day and multi-day forecasting, while keeping errors low.

These results suggest that our adaptive and optimized HDNN could be a game changer for electricity market participants, helping them make more accurate and reliable price predictions. Its flexibility and strong performance under different scenarios make it well suited for a variety of electricity markets, and the same approach could be extended to other time series forecasting problems. Future research should address additional challenges in smart grid systems, such as integrating renewable energy generation, optimizing electric vehicle charging control, and improving energy storage management. Addressing these issues will enhance the applicability and computational efficiency of the model, resulting in a more robust and effective approach.

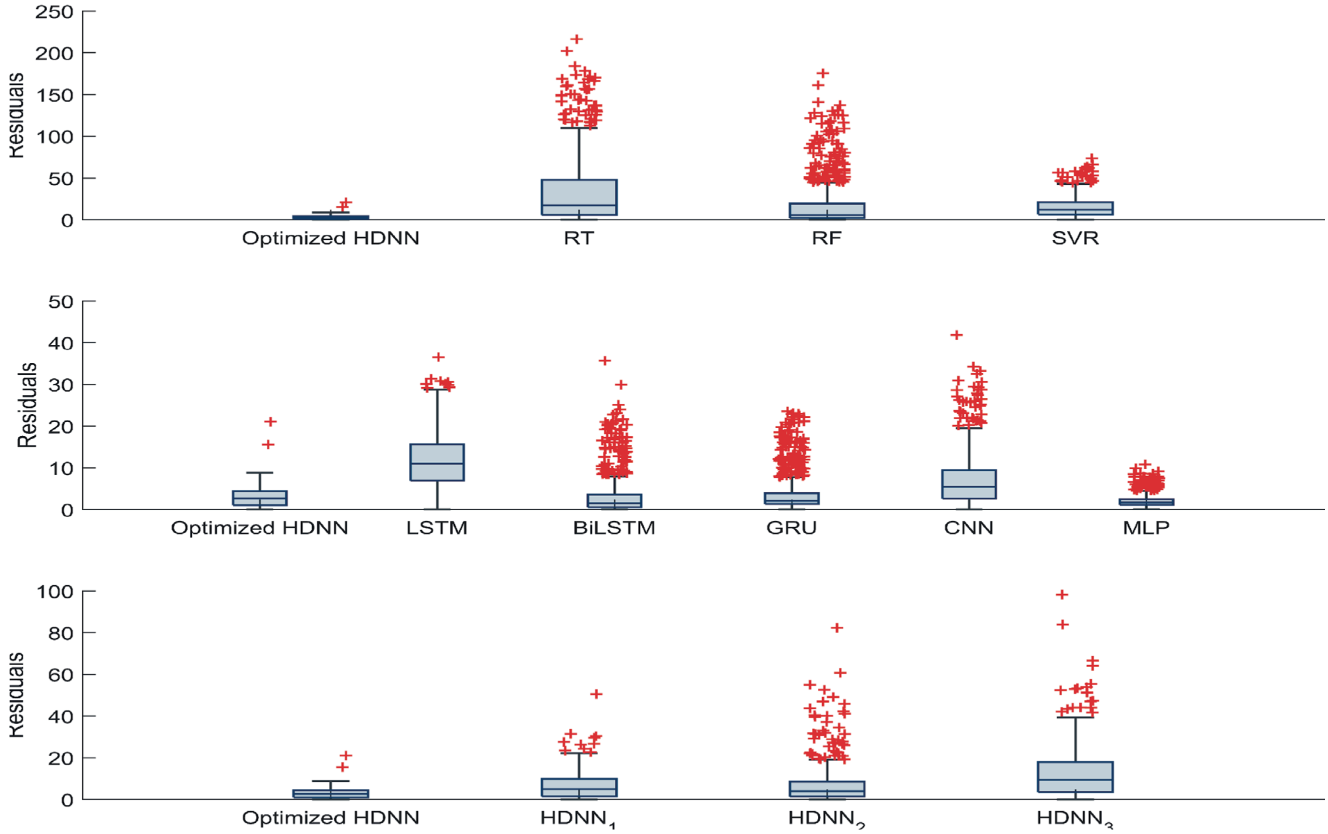

Fig. 18. Residual distribution comparison for the WCMA dataset

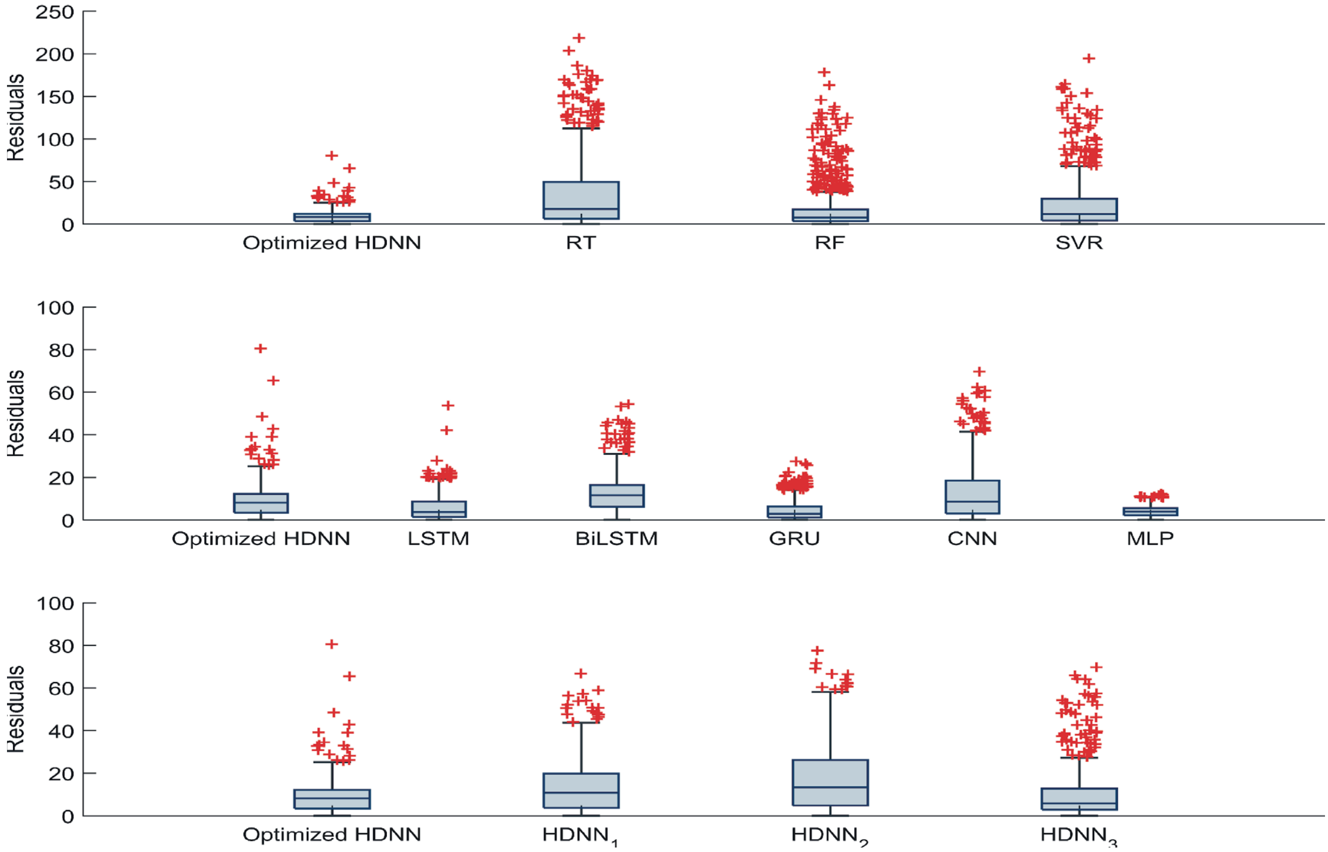

Fig. 19. Residual distribution comparison for the RI dataset

***Authors**: Abderrahim Bakir, PhD Candidate, IRECOM Laboratory, Department of Electrotechnics, Djillali Liabes University, Sidi Bel Abbes, Algeria, E-mail: abderrahim.bakir@univ-sba.dz; Dr. Abdelkader Rami, Associate professor, IRECOM Laboratory, Department of Electrotechnics, Djillali Liabes University, Sidi Bel Abbes, Algeria, E-mail: abc_rim20052003@yahoo.fr; Dr. Riyadh Bouddou, Associate professor, Department of Electrical Engineering, Institute of Technology, University Centre of Naama, Algeria, IEEE Member Algeria Section; E-mail: bouddou@cuniv-naama.dz*

REFERENCES

[1] Pourdaryaei, Alireza ; Mohammadi, Mohammad ; Mubarak, Hamza ; Abdellatif, Abdallah ; Karimi, Mazaher ; Gryazina, Elena ; Terzija, Vladimir: A new framework for electricity price forecasting via multi-head self-attention and CNN-based techniques in the competitive electricity market. In: *Expert Systems with Applications* Bd. 235, Pergamon (2024), S. 121207

[2] Li, Baozhu ; Khayatnezhad, Majid: Modified artificial neural network based on developed snake optimization algorithm for short-term price prediction. In: *Heliyon* Bd. 10, Elsevier Ltd (2024), Nr. 5

[3] Bouddou, Riyadh ; Benhamida, Farid ; Haba, Mekki ; Belgacem, Moussa ; Meziane, Mohammed Ainine: Simulated annealing algorithm for dynamic economic dispatch problem in the electricity market incorporating wind energy. In: *Ingenierie des Systemes d'Information* Bd. 25, International Information and Engineering Technology Association (2020), Nr. 6, S. 719–727

[4] Tan, Yong Qiang ; Shen, Yan Xia ; Yu, Xin Yan ; Lu, Xin: Day-ahead electricity price forecasting employing a novel hybrid frame of deep learning methods: A case study in NSW, Australia. In: *Electric Power Systems Research* Bd. 220, Elsevier (2023), S. 109300

[5] Xiong, Xiaoping ; Qing, Guohua: A hybrid day-ahead electricity price forecasting framework based on time series. In: *Energy* Bd. 264, Pergamon (2023), S. 126099

[6] Shao, Zhen ; Zheng, Qingru ; Liu, Chen ; Gao, Shuangyan ; Wang, Gang ; Chu, Yan: A feature extraction- and ranking-based framework for electricity spot price forecasting using a hybrid deep neural network. In: *Electric Power Systems Research* Bd. 200, Elsevier (2021), S. 107453

[7] Ghimire, Sujan ; Deo, Ravinesh C. ; Casillas-Pérez, David ; Sharma, Ekta ; Salcedo-Sanz, Sancho ; Barua, Prabal Datta ; Rajendra Acharya, U.: Half-hourly electricity price prediction with a hybrid convolution neural network-random vector functional link deep learning approach. In: *Applied Energy* Bd. 374, Elsevier (2024), S. 123920

[8] Wang, Keke ; Yu, Min ; Niu, Dongxiao ; Liang, Yi ; Peng, Sha ; Xu, Xiaomin: Short-term electricity price forecasting based on similarity day screening, two-layer decomposition technique and Bi-LSTM neural network. In: *Applied Soft Computing* Bd. 136, Elsevier (2023), S. 110018

[9] Rashid, Chughatta Khalid ; Saaqib, Haroon Shaikh: Short-Term Electricity Price Forecasting Using EEMD and GRU-NN. In: *International Journal of Advanced Natural Sciences and Engineering Researches* Bd. 7 (2023), Nr. 4, S. 420–427

[10] Shao, Zhen ; Yang, Yudie ; Zheng, Qingru ; Zhou, Kaile ; Liu, Chen ; Yang, Shanlin: A pattern classification methodology for interval forecasts of short-term electricity prices based on hybrid deep neural networks: A comparative analysis. In: *Applied Energy* Bd. 327, Elsevier (2022), S. 120115

[11] Mubarak, Hamza ; Abdellatif, Abdallah ; Ahmad, Shameem ; Zohurul Islam, Mohammad ; Muyeen, S. M. ; Abdul Mannan, Mohammad ; Kamwa, Innocent: Day-Ahead electricity price forecasting using a CNN-BiLSTM model in conjunction with autoregressive modeling and hyperparameter optimization. In: *International Journal of Electrical Power and Energy Systems* Bd. 161, Elsevier (2024), S. 110206

[12] Shejul, Kunal ; Harikrishnan, R. ; Gupta, Harshita: The improved integrated Exponential Smoothing based CNN-LSTM algorithm to forecast the day ahead electricity price. In: *MethodsX* Bd. 13, Elsevier (2024), S. 102923

[13] Kim, Jaedong ; Oh, Seunghwan ; Kim, Heesoo ; Choi, Woosung: Tutorial on time series prediction using 1D-CNN and BiLSTM: A case example of peak electricity demand and system marginal price prediction. In: *Engineering Applications of Artificial Intelligence* Bd. 126, Pergamon (2023), S. 106817

[14] *ISO New England*. URL https://www.iso-ne.com/. – abgerufen am 2024-09-24

[15] Kitsatoglou, Alkiviadis ; Georgopoulos, Giannis ; Papadopoulos, Panagiotis ; Antonopoulos, Herodotus: An ensemble approach for enhanced Day-Ahead price forecasting in electricity markets. In: *Expert Systems with Applications* Bd. 256, Pergamon (2024), S. 124971

[16] Bâra, Adela ; Oprea, Simona Vasilica: Devising single in-out long short-term memory univariate models for predicting the electricity price on the day-ahead markets. In: *Connection Science* Bd. 36, Taylor & Francis (2024), Nr. 1