https://doi.org/10.56093/JISAS.V80I1.14
Available online at https://isas.live/newsite
JOURNAL OF THE INDIAN SOCIETY OF AGRICULTURAL STATISTICS 80(1) 2026 149–163

Hybrid Deep Learning with Denoising Stacked Autoencoders for Agricultural Price Forecasting

K.N. Singh[1], G. Avinash[1,2], Kamal Sharma[1,2], Rajeev Ranjan Kumar[1], Mrinmoy Ray[3], Achal Lama[1], and S. Vishnu Shankar[4]

[1]ICAR - Indian Agricultural Statistics Research Institute, New Delhi
[2]The Graduate School, ICAR- Indian Agricultural Research Institute, New Delhi
[3]AKMU, ICAR- Indian Agricultural Research Institute, New Delhi
[4]Tamilnadu Agricultural University, Coimbatore

Received 07 April 2026; Revised 17 April 2026; Accepted 20 April 2026

SUMMARY

The complex nature of agricultural price data, characterized by perishability, seasonality, and non-stationarity, often renders conventional forecasting models inadequate. To address these challenges, this study proposes two hybrid learning frameworks: Stacked Autoencoder-based Deep Learning (SAE-DL) and Denoising Stacked Autoencoder-based Deep Learning (DSAE-DL). These models represent a significant departure from existing deep learning (DL) models, such as Multilayer Perceptron (MLP), Recurrent Neural Network (RNN), 1D Convolutional Neural Network (1D CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). Unlike these standard models, which process raw input directly and often struggle with data volatility, our proposed hybrid approach integrates an autoencoder bottleneck to perform hierarchical feature extraction and dimensionality reduction prior to forecasting. The SAE-DL model focuses on capturing the underlying compressed representation of the price series, while the DSAE-DL introduces a stochastic denoising mechanism to specifically reconstruct core features from noisy data and outliers. Empirical analysis using a real-world onion price dataset reveals that the proposed DSAE-DL models, particularly the DSAE-RNN configuration, consistently outperform all other models across metrics, including RMSE, MAE, and MAPE. The Diebold-Mariano (DM) test further confirms that both proposed hybrid frameworks, SAE-DL and DSAE-DL, significantly improve prediction accuracy compared to standalone DL models. These findings highlight the superior robustness of hybrid autoencoder-based architectures in contending with market volatility.

Keywords: Agricultural Price; Deep learning; Machine learning; Forecasting; Time series data.

1. INTRODUCTION

Precise and timely forecasting of agricultural commodity prices is crucial for stakeholders to make informed decisions and plan accordingly. Agricultural commodities, in particular, present a unique set of forecasting challenges not commonly encountered with non-farm goods and services. These challenges arise from the inherent vulnerability of agricultural commodities to a variety of unpredictable events, such as droughts, floods, disease outbreaks and pest infestations. In addition to these, factors such as seasonality, demand fluctuations, climate variability, market imperfections, globalization and speculative trading complicate the forecasting process. The nonlinear and nonstationary characteristics of agricultural price data further add to the complexity of accurately predicting future prices (Xiong *et al.* 2018).

India ranks as the second-largest vegetable producer globally, with a production output of 137.99 million metric tonnes (MMT). This includes a significant contribution from potato production, which reached 22.80 MMT during the 2021-22 period. Despite this prolific output, vegetable growers in India sometimes face severe challenges due to overproduction. Such

Corresponding author: K.N. Singh
E-mail address: drknsingh15@gmail.com

conditions can lead to distress sales, where farmers are compelled to sell their produce at significantly reduced prices or, in extreme cases, resort to burning crops or discarding them. This issue underscores the importance of equipping farmers with adequate storage facilities to weather periods of bearish market conditions, providing guidance on selling strategies during inflationary times and enhancing their understanding of the supply value chain. In addition, offering timely and accurate forecasts is vital for addressing these challenges and stabilizing prices.

The volatility of onion prices, particularly in India, highlights a significant economic challenge within agricultural markets. Essentially worldwide, onions face unpredictable price shifts due to a range of factors, impacting stakeholders across the board. India's primary onion-producing states: Maharashtra, Karnataka, Gujarat, Madhya Pradesh and Bihar are at the forefront of this volatility, grappling with a supply chain highly sensitive to external disruptions such as adverse weather, inadequate storage and fluctuating government policies. These issues are exacerbated by speculative trading and international trade restrictions, further destabilizing prices. This scenario emphasizes the importance of robust forecasting and strategic management to enhance market stability and resilience, benefiting both domestic and global agricultural economies. These situations lead to the development of complexitiesin DL model architectures to tackle the issues. The structure of the paper is as follows: Section 2 outlines the literature review, Section 3 describes the related work and underlying theoretical concepts, Section 4 presents the empirical analysis of the onion price series, and Section 5 offers concluding remarks, key findings, and future research directions. The references are provided at the end of the paper.

2. LITERATURE REVIEW

Forecasting is an indispensable tool for agricultural practitioners, policymakers and stakeholders, highlighting its significance in strategic decision-making. Statistical models, specifically designed for analyzing and forecasting sequential data points over time, form the foundation of forecasting practices. Time series models, leveraging historical price data, are pivotal in uncovering temporal patterns and trends within agricultural commodity prices, thereby facilitating informed decisions regarding future price movements (Sun *et al.* 2023). Recent advancements in data handling techniques have spurred a dynamic evolution in the field of time series forecasting, leading to the development of models that significantly surpass traditional methods in prediction accuracy (Oikonomidis *et al.* 2023). A wide array of time series models has been explored to enhance price forecasting, broadly categorized into statistical models and artificial intelligence (AI)-based models, which include Machine Learning (ML) and Deep Learning (DL) models. The ARIMA model, proposed by Box and Jenkins in the 1970s, has been widely employed in time series analysis, particularly for forecasting financial data. Despite its widespread application, ARIMA's effectiveness is limited in handling nonlinear data, prompting researchers to explore alternative nonlinear time series models such as SETAR and GARCH. These models are capable of capturing nonlinearity but may lack in generalization and require specific data relationships. The traditional statistical models, which assume a predetermined functional form and heavily rely on data distribution assumptions, encounter limitations in capturing complex patterns and nonlinear relationships. This has led to the rise of ML models as an effective alternative for addressing the challenges associated with non-linear datasets (Paul *et al.* 2022). Furthermore, the advent of Deep Learning (DL) has marked a significant milestone in forecasting, with DL models renowned for their ability to learn complex and non-linear relationships autonomously. This capability makes DL models particularly adept at handling large datasets and improving forecasting accuracy, distinguishing them from traditional statistical and ML techniques. However, despite the advancements in DL, challenges persist, particularly in processing long sequences and high-dimensional data efficiently, underscoring the continuous pursuit for more sophisticated models in the domain of agricultural price forecasting (Najafi *et al.* 2022). In this study, we aim to develop a model for the Onion price series by employing variants of Autoencoder DL models. The initial application of conventional DL models, such as Multilayer Perceptron (MLP) (Haykin, 2009), Recurrent Neural Network (RNN), one-dimensional Convolutional Neural Network (1d CNN), Long Short-term Memory (LSTM) and Gated Recurrent Unit (GRU) to the data, sets the foundation for further enhancements. These models, when integrated with

stacked autoencoders, are poised to significantly enhance prediction accuracy. However, despite their capabilities, convolutional deep learning models face challenges in capturing the temporal dynamics essential for accurately modelling time series data. This limitation stems primarily from their inherent design, which excels in identifying spatial hierarchies in data rather than temporal relationships. To address this challenge and further bolster the model's forecasting capabilities, our study explores the integration of Denoising Stacked Autoencoders (DSAE) (Liu *et al.* 2019). The inclusion of a denoising mechanism within the autoencoder structure provides a dual advantage. Firstly, it enhances the model's ability to capture and reconstruct the core features of the input data, even in the presence of noise (outliers). This capability is particularly valuable in handling the intrinsic volatility and noise present in agricultural price series data. Secondly, the process of denoising aids in the robustness of the features learned by the model, making it more adept at forecasting under conditions of data uncertainty and variability. Remarkably, there has been no existing work on the application of these advanced autoencoder models to multivariate time series or price series data until now. The employment of DL models within the bottleneck of the autoencoder structure allows these networks to compress and reconstruct the input data effectively. Moreover, it enables the extraction of deep, complex patterns crucial for forecasting future values in multivariate time series data. This approach is epitomized in the DSAE-DL model, where the induction of additional noise not only challenges the model to recover the original signal but also significantly improves the robustness and generalization of the learned features. This novel application promises to unlock new potentials in the accuracy and reliability of onion price forecasts, providing substantial insights for stakeholders navigating the complexities of the onion market.

3. METHODOLOGY

In this paper, different DL models for time series forecasting were used, viz., conventional DL models, SAE-DL, followed by DSAE-DL models, were fitted for the price series of onion to capture the volatility. The performance of all the fitted models is compared, and the best-fitted model for the data is selected using the error measures.

3.1 Multilayer Perceptron (MLP)

The proposed models build upon the **Multilayer Perceptron (MLP)**, a feedforward architecture highly effective for modelling complex temporal dynamics in time series forecasting (Liu *et al.* 2022). An MLP consists of an input layer, one or more hidden layers, and an output layer, where interconnected neurons transform information via weighted links.

The fundamental operation of an MLP neuron is defined by:

$$\hat{y} = \sum_{j=0}^{s_0} W_{jk} \cdot \xi\left(\sum_{i=0}^{s_h} W_{ij} \cdot x \right) \tag{1}$$

where $\hat{y}$ denotes the network's output, x represents the input vector, s_0 and s_h are the sizes of the output and hidden layers respectively, and W_{ij} and W_{jk} are the weights connecting the input to the hidden layers and the hidden to the output layers, respectively. The function ξ, defined as:

$$\xi(\alpha, x) = \begin{cases} \alpha \cdot \left(e^x - 1\right), & \text{if } x < 0 \\ x, & \text{otherwise} \end{cases} \tag{2}$$

is known as the Exponential Linear Unit (ELU) when $\alpha = 1$, and it simplifies to the Rectified Linear Unit (ReLU) for $\alpha = 0$. The activation function introduces the non-linearity necessary for the network to map complex, non-linear patterns within time series data.

3.2 One-Dimensional Convolutional Neural Networks (1dCNN)

The 1D-Convolutional Neural Network (1D-CNN) is an effective architecture for time series analysis, utilizing convolutional, sub-sampling (pooling), and fully-connected layers to extract temporal features and perform classification.

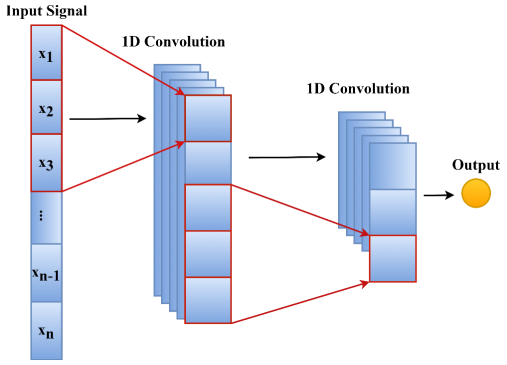

Fig. 1. Architecture of a 1D Convolutional Neural Network

The convolutional layer captures temporal dependencies from adjacent nodes using shared local weights, which reduces memory requirements and enhances pattern identification. The 1D convolution operation is defined as:

$$Z_t = \sigma\left(W * X_t + b\right),$$

where σ denotes the activation function, W the weight matrix, Xt the input at time t, and b the bias. The sub-sampling (pooling) layers apply non-linear down-sampling to reduce temporal dimensionality. This decreases computational load while enabling the model to extract the most salient sequential features. The pooling operation is defined as:

$$Y_t = \text{pooling}\ (Z_t),$$

Pooling performs down-sampling on the convolutional output to reduce dimensionality. The architecture concludes with a fully-connected layer that executes a dense matrix computation, synthesizing extracted temporal features to generate the final output for applications such as financial price forecasting:

$$\hat{Y}_t = W_{\text{output}} \cdot A_t + b_{\text{output}}$$

with W_{output} representing the weight matrix of the fully-connected layer, At the activation from the previous layer, and b_{output} the output bias.

During training, the 1D-CNN minimizes the reconstruction error via backpropagation through time (BPTT), optimizing parameters to capture the underlying temporal dependencies:

$$L_t = \frac{1}{2}\sum_i \left(Y_{t,i} - \hat{Y}_{t,i}\right)^2$$

where L_t represents the loss at time t. The integration of convolutional, sub-sampling, and fully-connected layers allows the 1D-CNN to efficiently extract and leverage temporal patterns for time series analysis.

3.3 Recurrent Neural Networks (RNN)

RNNs utilize recurrent hidden layers to maintain a short-term memory of sequential inputs, making them suitable for time-series analysis. While they support various configurations (e.g., many-to-one, many-to-many) and are trained via Backpropagation Through Time (BPTT), they are hindered by vanishing and exploding gradients. Additionally, their sequential nature limits scalability and increases training time for complex sequences. The basic RNN cell is illustrated in Fig. 2.

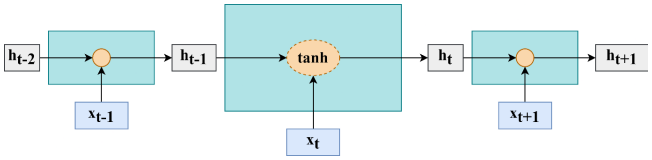

Fig. 2. Architecture of an RNN cell

The workflow involves sequential input processing and hidden state update. At each time step t, the hidden state h_t is computed using the input x_t and the previous hidden state $h_t - 1$, as shown in the following equation:

$$h_t = \sigma\left(W_{ih} \cdot x_t + W_{hh} \cdot h_{t-1} + b_h\right)$$

3.4 Long Short-term Memory Networks (LSTM)

LSTMs enhance the standard RNN architecture by addressing the long-term dependency problem through a gated mechanism. This allows the network to selectively retain or discard historical information over extended sequences using three sigmoid-based gates: the forget gate (f_t), input gate (i_t), and output gate (o_t) by (Hochreiter and Schmidhuber 1997). These gates meticulously regulate the flow of information, deciding what data to retain or discard within the cell state, thereby bolstering the network's capacity to capture long-term dependencies. The dynamics within an LSTM cell encompass a series of operations to modulate its internal state and output: The forget gate (f_t) controls the degree to which information from the previous cell state is kept:

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

Simultaneously, the input gate (i_t) influences the cell state by introducing a new candidate value, calculated as:

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

This candidate value for updating the cell state ($\hat{C}_t$) is derived through a tanh activation function:

$$\hat{C}_t = \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right).$$

Subsequently, the cell state (C_t) is updated by amalgamating the influences of both the forget and input gates:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t .$$

The output gate (o_t) then dictates which components of the cell state are utilized to compute the final output:

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

Lastly, the LSTM cell's output value (ht) is determined by modulating the cell state with the output gate's activation:

$$h_t = o_t \cdot \tanh\left(C_t\right)$$

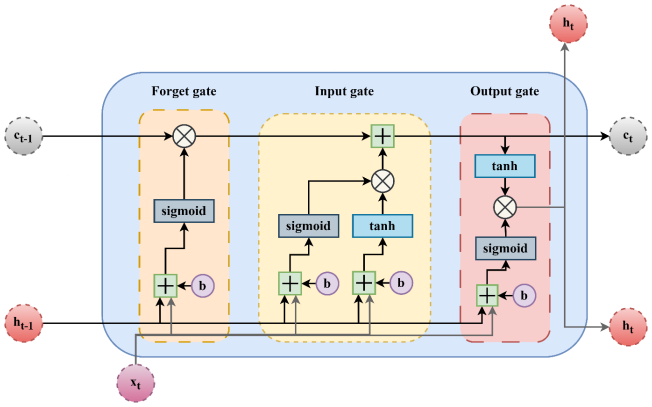

Fig. 3. Architecture of an LSTM cell

Within these equations, W_f, W_i, W_c, and W_o denote the weight matrices allocated for each gate and the cell state update, correspondingly. The bias terms b_f, b_i, b_c, and b_o contribute to the computations of their respective gates. The gates' operations utilize the sigmoid activation function (σ) to manage information flow, while the tanh function normalizes the outputs for the cell state and final output. LSTM architectures often incorporate ReLU activation in fully-connected layers and utilize Mean Square Error (MSE) as the loss function. These configurations facilitate the modelling of temporal dependencies across various sequential data processing tasks.

3.5 Gated Recurrent Units (GRUs)

Introduced by Chung *et al.* 2014, Gated Recurrent Units (GRUs) simplify the LSTM architecture by merging the input and forget gates into a single update gate and combining the hidden and cell states. This streamlined structure reduces the parameter count and improves computational efficiency.

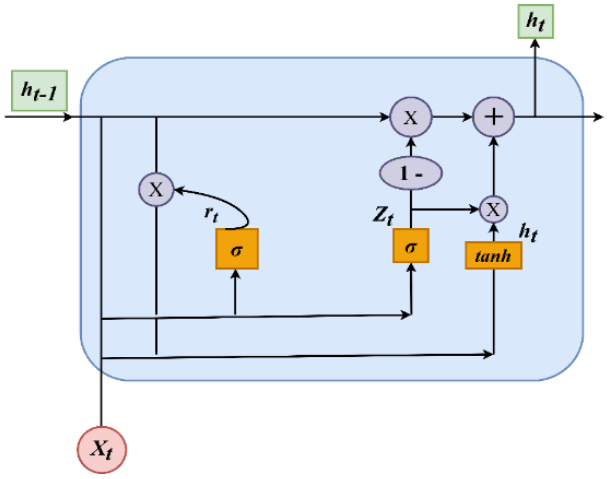

Fig. 4. Architecture of a GRU cell

GRUs are particularly adept at prioritizing recent information over distant past events, which is often more relevant for future predictions. This is accomplished through two key mechanisms: the update gate and the reset gate. The update gate (Z_t), defined by the equation

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

regulates the extent to which the GRU cell updates its activation, or hidden state. Conversely, the reset gate (R_t) is described by

$$R_t = \sigma\left(W_R \cdot \left[h_{t-1}, x_t\right] + b_R\right)$$

determining how much of the past information will be forgotten. These gates collaboratively filter and process information through the cell, ensuring that GRUs can maintain relevant information over time.

The new memory content ($\hat{H}_t$), which is intended to update the hidden state, is computed as follows:

$$\hat{H}_t = \tanh\left(W \cdot \left[R_t \cdot h_{t-1}, x_t\right] + b_z\right).$$

Subsequently, the GRU updates its hidden state (h_t) by blending the old state with the new memory content, influenced by the update gate:

$$h_t = \left(1 - Z_t\right) \cdot h_{t-1} + Z_t \cdot \hat{H}_t .$$

This concise structure, governed by the weight matrices W_z, W_R and W, along with bias vectors b_z and b_R, and regulated by sigmoid activation functions, allows GRU layers to adeptly manipulate recurrent connections and inputs. Consequently, GRUs present a highly efficient and effective model for sequential data processing, capable of learning to prioritize recent over outdated information with fewer parameters than LSTMs.

3.6 Stacked Autoencoders (SAE)

An Autoencoder (AE) is an unsupervised neural network designed for dimensionality reduction and feature learning (Hinton *et al.* 2006). The Stacked Autoencoder (SAE-DL) builds upon this by layering multiple encoders and decoders to extract a hierarchy of features from multivariate time series. This structure addresses the limitations of shallow AEs by capturing deep temporal dependencies that are often lost in less complex architectures. In the SAE-DL framework, the architecture is trained using a greedy layer-wise approach. Each layer is optimized as a standalone autoencoder; once trained, its internal representation (the bottleneck features) serves as the input for the subsequent layer in the stack. This hierarchical structure allows the model to capture deep, complex temporal dependencies by progressively distilling the input data into more abstract feature sets. The encoding and decoding processes in SAE are mathematically expressed as:

$$\text{Encoded}_k = f_{\text{encoder}_k}\left(\text{Encoded}_{k-1}; \theta_{ek}\right),\ k = 1, \ldots, K,$$

where K is the number of stacked layers, Encoded_0 = X is the input time series, and θ_{ek} are the parameters of the k-th encoder.

$$\hat{X} = f_{\text{decoder}_1}\left(f_{\text{decoder}_2}\left(\ldots f_{\text{decoder}_K}\left(\text{Encoded}_K; \theta_{dK}\right)\ldots\right); \theta_{d1}\right)$$

where X^ is the reconstructed output, and θ_{dk} are the parameters of the k-th decoder.

3.7 Denoising Stacked Autoencoders (DSAE)

The Denoising Stacked Autoencoder (DSAE-DL) enhances the SAE-DL architecture by introducing stochastic corruption to the input data during the training phase. This modification forces the model to learn more robust, invariant features that are less susceptible to noise and overfitting. By training the network to reconstruct the original "clean" data from a corrupted version, the DSAE-DL achieves superior generalization performance compared to standard autoencoders (Samal *et al.* 2021).

The architecture follows the hierarchical structure of the SAE-DL, but incorporates a stochastic noise injection step (typically using Gaussian noise) applied to the input time series. Each layer in the stack is trained to minimize the discrepancy between the noisy input and the original undistorted signal. This process ensures that the extracted feature hierarchy prioritizes salient temporal patterns over transient fluctuations. The DSAE model is succinctly described by the following equations:

$$X_{\text{noisy}} = X + \mathcal{N}\left(\mu, \sigma^2\right)$$

$$\text{Encoded}^{\text{noisy}} = f_{\text{encoder}}\left(X_{\text{noisy}}; \theta_e\right)$$

$$\hat{X} = f_{\text{decoder}}\left(\text{Encoded}^{\text{noisy}}; \theta_d\right)$$

where Encodednoisy is the latent representation of the noisy input, θ_e and θ_d are the parameters of the encoder and decoder, respectively. The DSAE's objective is to minimize the reconstruction loss between the original time series X and the reconstructed series $\hat{X}$, often using mean squared error as the loss function. This approach enables the DSAE to robustly forecast future values by leveraging the noise-invariant features learned during training.

3.8 Proposed Methodology for SAE-DL and DSAE-DL Models

Figs. 5 and 6 illustrate the proposed hybrid architectures: Stacked Autoencoders with Deep Learning (SAE-DL) and Denoising Stacked Autoencoders with Deep Learning (DSAE-DL). These models integrate unsupervised feature extraction with supervised sequence modelling to improve multivariate time series forecasting.The methodology follows a structured four-stage process:

1. **Input Processing:** The SAE-DL utilizes the raw multivariate time series. In contrast, the DSAE-DL applies Gaussian noise to the input, creating a corrupted dataset that forces the model to learn noise-invariant patterns.

2. **Hierarchical Encoding:** Both architectures pass the input through successive encoder layers. This stage compresses the data into a lower-dimensional latent space, capturing abstract temporal features while discarding redundant information.
3. **Deep Learning Bottleneck:** The core of the architecture integrates a DL model (MLP, RNN, LSTM, or GRU) at the bottleneck layer. These models process the refined latent features to capture long-range temporal dependencies.
4. **Reconstruction and Prediction:** The framework concludes with a decoder phase to reconstruct the input (for feature validation) and a final output layer for the forecasting task.

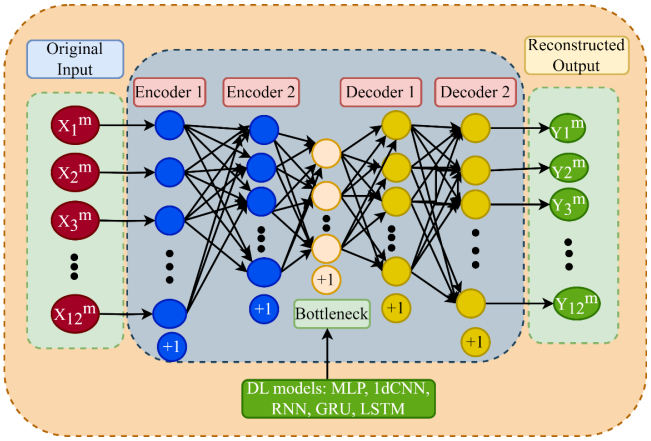

Fig. 5. Architecture of a SAE-DL models

Decoder Layers: The latent space representation is then reconstructed back into the original data dimensionality through multiple decoder layers. The objective is to accurately reconstruct the input time series for the SAE-DL model and the denoised time series for the DSAE-DL model.

Reconstructed Output: The final output is the reconstructed multivariate time series data, which should ideally resemble the original data closely for the SAE-DL model and the denoised version of the original data for the DSAE-DL model.

Training Objective: The training process involves optimizing the model parameters to minimize the loss function, usually the mean squared error, between the original (or denoised for DSAE-DL) and reconstructed time series.

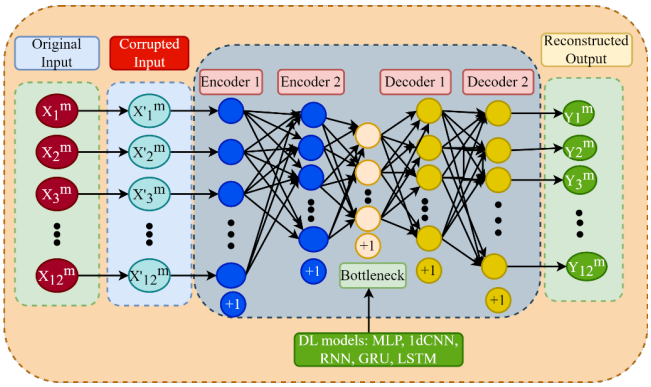

Fig. 6. Architecture of a DSAE-DL models

The use of DL models within the bottleneck of the autoencoder structure allows these networks to not only compress and reconstruct the input data but also to capture deep, complex patterns crucial for forecasting future values in multivariate time series data. This is especially true for the DSAE-DL model, where the additional noiseinduction step serves to improve the robustness of the learned features.

3.9 Grid Search Hyperparameters Selection

Hyperparameters selection plays a critical role in ML/DL, directly impacting the model's complexity, learning rate and ability to generalize. Key hyperparameters, such as batch sizes, the number of hidden layers and the number of units per layer, are determined before training commences, playing a vital role in optimizing the model's learning efficiency. The selection of the number of lags, informed by the Autocorrelation Function (ACF) plot and the Augmented Dickey-Fuller (ADF) test, along with a predetermined stopping criterion of 200 epochs, refines our methodology. We utilize a grid search technique for hyperparameters tuning, methodically traversing a grid to identify the optimal combination that enhances validation performance (Nayak *et al.* 2024). This thorough approach, illustrated in Fig. 7, encompasses initializing hyperparameters, generating combinations, conducting training sessions for each, and pinpointing the most effective set for ultimate training, thus ensuring the model is finely tuned for its intended application (Avinash *et al.* 2024).

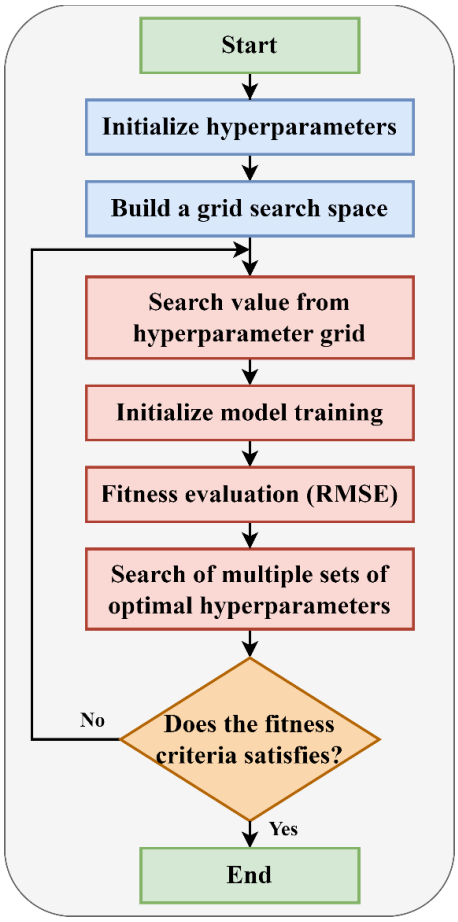

Fig. 7. Grid search hyperparameter tuning

From a mathematical standpoint, this procedure represents an optimization challenge, aiming to pinpoint hyperparameters *h* that elevate validation set performance. Given hyperparameters such as batch size *b* within {8, 16, 32, 64, 128, 256}, number of hidden layers l within {1, 2, 3}, and units per layer *u* within {8, 16, 32, 64, 128, 256, 512}, the hyperparameter space H = B × L × U spans all possible combinations. The objective is to discover $h* = \text{argmin}\ h \in H\ L(Mh)$, necessitating the training and evaluation of 6 × 3 × 7 = 126 distinct models, based on the cardinalities of the sets. The optimal set $h*$ is then employed for exhaustive model training, validation and testing on novel data. The computational experiments were facilitated by Python, utilizing its robust capabilities for training DL models. These experiments were conducted on a computing system equipped with an AMD Ryzen 7 5700U processor and 8 GB of RAM, which was found to be sufficient for the training and evaluation of each deep learning model. The duration for model processing ranged from 25 to 30 minutes per model, leveraging the grid search hyperparameter tuning strategy.

3.10 Model Evaluation

To quantitatively assess the performance of each forecasting model, we employed three widely used accuracy metrics: Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE). RMSE provides a measure of the average magnitude of the forecast errors, emphasizing larger deviations, while MAE quantifies the average absolute difference between predicted and actual values, offering a straightforward interpretation of overall error. MAPE, on the other hand, expresses the forecast error as a percentage of the actual values, facilitating comparison across datasets with different scales. In addition to these metrics, a post hoc analysis was conducted using the Diebold-Mariano (DM) test to identify the model with superior predictive performance. The DM test statistically evaluates whether the differences in forecast accuracy between two competing models are significant, enabling a robust comparison of their predictive capabilities. This combination of error metrics and formal statistical testing ensures a comprehensive and rigorous evaluation of the models, providing greater confidence in the selection of the most reliable forecasting approach.

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

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

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$$

where,

y_i : True values of the variable

$\hat{y}_i$: Predicted values of the variable

N : Number of observations in the dataset

The Diebold-Mariano (DM) test formula is given by:

$$\text{DM} = \frac{\bar{d}}{\sqrt{\text{Var}\left(\bar{d}\right)}}$$

where d is the mean difference of the forecast errors from two models, and Var(d) is the variance of these differences.

4. RESULTS AND PERFORMANCE COMPARISON

4.1 Data Description

This study utilizes the weekly Onion price series (in Rs./ Quintal) from January 8, 2006, through June 11, 2023 (obtained from the Agmarknet; https://agmarknet.gov.in) from Lasalgaon market, Maharashtra (India) comprising 910 records (tuples) of numerical data stored in .CSV format. Each record includes three primary attributes: Minimum, Maximum, and Modal prices. Table 1 displays descriptive statistics of the Onion multivariate time series utilized in the study, and Fig. 8 exhibits the time plot containing Minimum, Maximum and Modal price considered under study.

Table 1. Descriptive Statistics of Onion Price Series

Statistics	Minimum Price	Maximum Price	Modal Price
Minimum (in Rs./ Quintal)	216.50	236.00	227.75
Mean (in Rs./ Quintal)	918.80	1374.68	1243.50
Maximum (in Rs./ Quintal)	4411.75	7822.89	6657.77
Standard Deviation (in Rs./ Quintal)	686.85	1049.49	966.87
Coefficient of Variation (%)	74.71	76.30	77.71
Skewness	1.91	2.02	1.94
Kurtosis	4.24	5.37	4.56
Jarque-Bera Test	1243.39**	1716.25**	1361.50**
Shapiro-Wilks Test	0.80**	0.81**	0.81**

Note: ** *Significant at 5% level of significance*

The descriptive statistics, summarized in Table 1, reveal substantial market volatility, with a Coefficient of Variation (CV) exceeding 74% across all categories. The significant disparity between mean and maximum values underscores the market's susceptibility to extreme price escalations. Statistical measures of skewness and kurtosis (K > 3) indicate that the distributions are right-skewed and leptokurtic. The non-normal nature of the series is further validated by the Jarque-Bera and Shapiro-Wilk tests ($p < 0.05$). These complex, heavy-tailed characteristics justify the application of the proposed **SAE-DL** and **DSAE-DL** architectures for effective non-linear feature extraction and denoising.

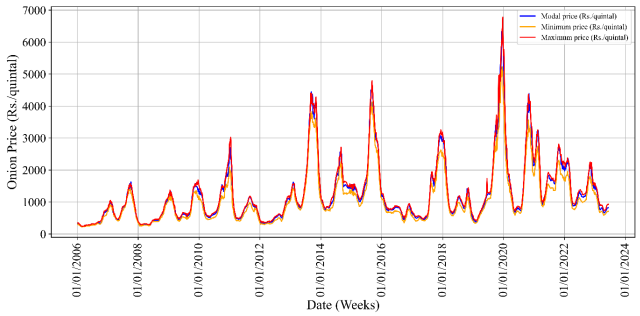

Fig. 8. Time series plot of weekly Onion price series

Our study encompasses 910 observations with three parameters, adhering to a data partitioning ratio of 80%, 10% and 10% for training, validation and testing, respectively. This equates to 728, 91 and 91 data points for each phase. Such an allocation strategy is meticulously designed to fine-tune model performance while adeptly balancing the bias-variance trade-off. It allows for the leveraging of the maximum amount of data for training (80%), thereby reducing bias by enabling the model to learn comprehensive patterns within the Onion market's price series. Concurrently, allocating 10% of the data to both validation and testing phases aims to minimize variance with minimal loss by evaluating model performance on previously unseen data, ensuring the model's predictive reliability. To assess the stationarity and nonlinearity of the series within this study, Augmented Dickey-Fuller (ADF) and Brock-DechertScheinkman (BDS) tests were performed. The ADF test, with its null hypothesis positing nonstationarity or presence of a unit root in the series, confirmed nonstationarity with an ADF value of 1.84 (p-value: 0.35) with 12 fixed lags based on PACF and ACF plots, with the ADF test considered under the study. Conversely, the BDS test's null hypothesis suggests that the series follows a linear pattern. Results obtained with statistics surpassing 755.00 for 0.5, 1.0, 1.5, and 2.0 σ across 2 and 3 embedding dimensions suggest the rejection of this null hypothesis, indicating the non-linear nature of the price series across all categories. Additionally, the Autoregressive Conditional Heteroskedasticity (ARCH) LM test revealed the presence of volatility in all the price series under consideration. Armed with the necessary insights from these statistical tests, our study progresses towards training the Denoising Stacked Autoencoder Deep Learning (DSAE-DL) models. The primary objective is to evaluate the performance

of these models for forecasting agricultural price series and to compare them with conventional Deep Learning (DL) models. By employing the Grid Search Cross-Validation (GSCV) technique, this study aims to illuminate which model exhibits superior performance in price forecasting.

4.2 Data Pre-processing

Prior to implementation, we found that the data contains approximately 7-9% missing values in the daily data. Imputation of missing values was performed using Kalman smoothing imputation method, and later averaged the daily data in weeks. Furthermore, to enhance the effectiveness of training DL models, the data were preprocessed using the 'Min-Max Scaler' transformation to normalise them.

4.3 Model Building

The development of forecasting models involves two main stages: hyperparameter tuning and model training, and forecasting.

4.3.1 Grid search hyperparameter tuning

After confirming the presence of nonstationarity, nonlinearity, and volatility, the models were prepared for training. To enhance model performance, we optimized key hyperparameters, including batch sizes (8, 16, 32, 64, 128, 256), number of hidden layers (1, 2, 3), and number of hidden units/kernel filters (8, 16, 32, 64, 128, 256, 512). The number of lags was fixed at 12 weeks based on the PACF visualization and ADF test results. The number of epochs was determined using an early stopping criterion, with a maximum limit of 200. A learning rate of 0.01, the Adam optimizer, and the LeakyReLU activation function were selected due to their demonstrated effectiveness for the characteristics of our dataset. These hyperparameters were systematically varied across defined ranges, and the optimal combination was identified using grid search cross-validation. The final set of optimal hyperparameters is presented in Table 2 and played a crucial role in achieving superior model performance. Early stopping and loss-curve analysis were essential in optimizing the models for forecasting horizons of up to 12 weeks, effectively minimizing overfitting. The close alignment between training and validation losses indicates that the models successfully captured the underlying data patterns without memorizing noise, which is critical for ensuring generalization to unseen test data. The consistent performance across multiple forecasting periods demonstrates the robustness of the Denoising-based Stacked Autoencoder Deep Learning (DSAE-DL) models in learning complex temporal dependencies. This balanced model configuration enhances predictive accuracy and reliability, offering valuable guidance for future model development and selection in time series forecasting.

Table 2. Optimal hyperparameters obtained by grid search methodology

Models	Batch Size	Hidden Layer	Hidden Units Per Layer
DL Models			
MLP	128	3	128, 64, 32
1dCNN	128	3	128, 64, 32
RNN	64	3	128, 64, 32
LSTM	64	2	128, 64
GRU	64	2	256, 64
SAE-DL Models			
MLP	128	3	64, 32, 16
1dCNN	64	3	256, 128, 64
RNN	128	3	128, 64, 32
LSTM	64	3	128, 64, 32
GRU	64	2	128, 64
DSAE-DL Models			
MLP	64	3	128, 64, 32
1dCNN	64	2	128, 64
RNN	128	2	256, 64
LSTM	64	2	256, 128
GRU	128	2	128, 64

4.3.2 Model training

After identifying the optimal hyperparameters for each deep learning (DL) model, we meticulously trained them, with particular attention paid to preventing overfitting and underfitting. This goal was achieved by closely observing the loss plot curves and employing a stopping criterion method, allowing for the timely halting of the training process. These precautionary measures were instrumental in refining the proposed models, which include the Stacked Autoencoder Deep Learning (SAE-DL) models, comprising SAE-MLP, SAE-1dCNN, SAE-RNN, SAE-LSTM and SAE-GRU, and the Denoising Stacked Autoencoder Deep Learning (DSAE-DL) models, namely DSAE-MLP, DSAE-1dCNN, DSAE-RNN, DSAE-LSTM and DSAE-GRU, in addition to the traditional DL models, MLP, 1dCNN,

RNN, LSTM and GRU. This comprehensive approach ensured that all models were optimally tuned and ready to demonstrate their efficacy on data they had not previously encountered.

Table 3. Performance Metrics of Different Models on Test Data

Types of Models	RMSE	MAE	MAPE
DL Models			
MLP	149.20	111.18	7.97
GRU	174.79	132.94	10.76
1dCNN	179.21	136.16	11.48
RNN	140.87	104.24	8.11
LSTM	145.55	113.46	8.66
SAE-DL Models			
MLP	147.43	109.94	8.35
GRU	151.58	123.08	10.74
1dCNN	138.03	97.06	6.74
RNN	256.44	218.90	22.23
LSTM	152.28	112.81	8.21
DSAE-DL Models			
MLP	**92.02**	**63.89**	**4.43**
GRU	**92.27**	**66.68**	**4.57**
1dCNN	**127.19**	**90.33**	**6.64**
RNN	**79.10**	**58.16**	**4.05**
LSTM	**107.95**	**76.91**	**5.52**

4.3.3 Forecasting performance of different models

The comparative analysis of Deep Learning (DL), Stacked Autoencoder Deep Learning (SAE-DL) and Denoising Stacked Autoencoder Deep Learning (DSAE-DL) models on test data reveals significant differences in performance across the models, as shown in Table 3 and Fig. 9. Conventional DL models show a range of performance, with the RNN model achieving the lowest RMSE of 140.87, indicating its relatively better predictive accuracy among conventional DL models. The 1dCNN model, however, records the highest RMSE of 179.21, suggesting challenges in capturing the data's temporal dynamics effectively. SAE-DL models, incorporating autoencoders for feature extraction, demonstrate varied outcomes. The 1dCNN model stands out with the lowest RMSE of 138.03, surpassing its DL counterpart. This suggests that the autoencoder's feature learning capability may enhance predictive performance. Conversely, the SAERNN model exhibits significantly higher error rates, indicating the feature extraction for the RNN model causes inadequacy in model architecture for the given data. DSAE-DL models consistently outperform both DL and SAE-DL models, with the DSAE-RNN model showing exceptional performance with the lowest RMSE of 79.10. This remarkable improvement underscores the effectiveness of denoising autoencoders in handling noise and extracting more robust features for prediction. The DSAEMLP and DSAE-GRU models also exhibit superior performance, further validating the advantages of the denoising approach. Overall, the DSAE-DL models demonstrate superior predictive accuracy, highlighting their potential for enhancing forecasting tasks. The insights from this analysis suggest that incorporating advanced feature learning and noise reduction techniques can significantly improve model performance.

To further understand the significance of the performance differences between DL, SAE-DL and DSAE-DL models, we conducted a Diebold-Mariano (DM) statistical test. The comprehensive pairwise comparison encompassed 15 models, resulting in 105 unique pairwise combinations. Of these, 39 pairs were identified as not showing significant differences, whereas the remaining 66 pairs demonstrated statistically significant differences, as illustrated in the associated figure. Notably, the test statistic for this analysis yielded a DM value of 1.98. The most significant pairwise difference was observed between the SA-RNN and DSA-GRU models, with a DM value of -7.2, highlighting the superior performance of the DSA-GRU model. Similarly, a significant distinction was noted between the SA-RNN and SAE-RNN models, with a DM value of 6.7 as shown in Fig. 10. These results underscore the effectiveness of the DSAE-RNN model, which not only outperformed all other models compared but also demonstrated the advantage of integrating denoising mechanisms within stacked autoencoders for enhancing forecasting accuracy in highly volatile agricultural commodity price prediction.

5. DISCUSSION

In the realm of agricultural commodity price forecasting, our study embarked on a comprehensive evaluation of Deep Learning (DL), Stacked Autoencoder Deep Learning (SAE-DL) and Denoising Stacked Autoencoder Deep Learning (DSAE-DL) models. Through rigorous comparison, we have unearthed significant insights into the predictive accuracy of

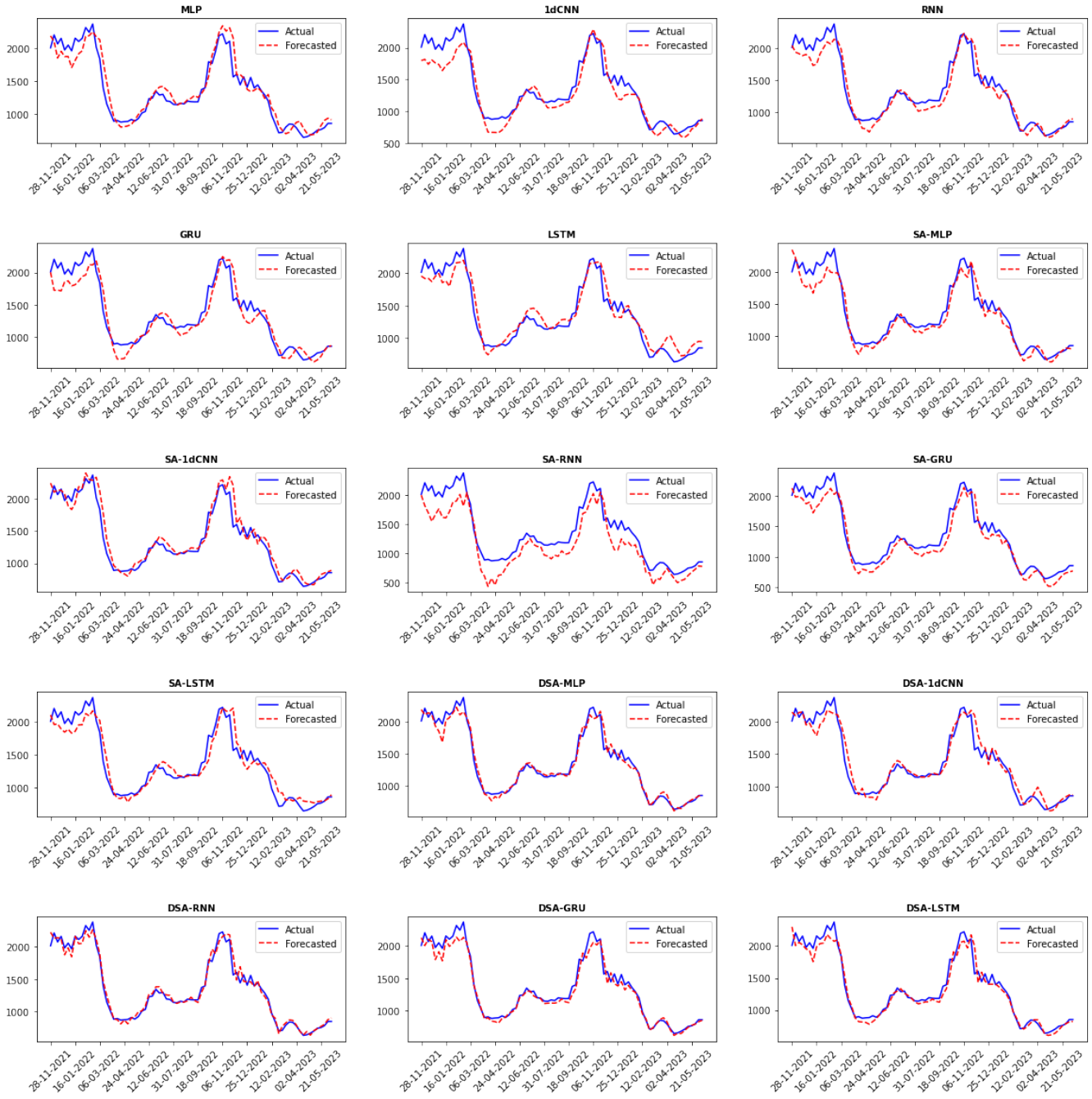

Fig. 9. Actual vs. forecasted plots on test data

these models, quantified by the percentage decrease in Root Mean Square Error (RMSE). Our results revealed that DSAE-DL models substantially 22 outperform their DL counterparts across various architectures. Specifically, the DSAEMLP model demonstrated a 38.32% improvement over the conventional MLP DL model, indicating a significant enhancement in accuracy. Similarly, the DSAE-GRU and DSAE-RNN models showcased remarkable accuracy improvements of 47.21% and 43.85%, respectively, over their DL counterparts. The DSAE-1dCNN and DSAELSTM models also exhibited notable improvements, with increases of 29.03% and 25.83%, underscoring the efficacy of denoising autoencoders in enhancing model performance. In comparison to the SAE-DL models, the improvements were mixed. The SAE-MLP model showed a modest improvement of 1.19% over its DL counterpart, while the SAE-GRU and SAE-

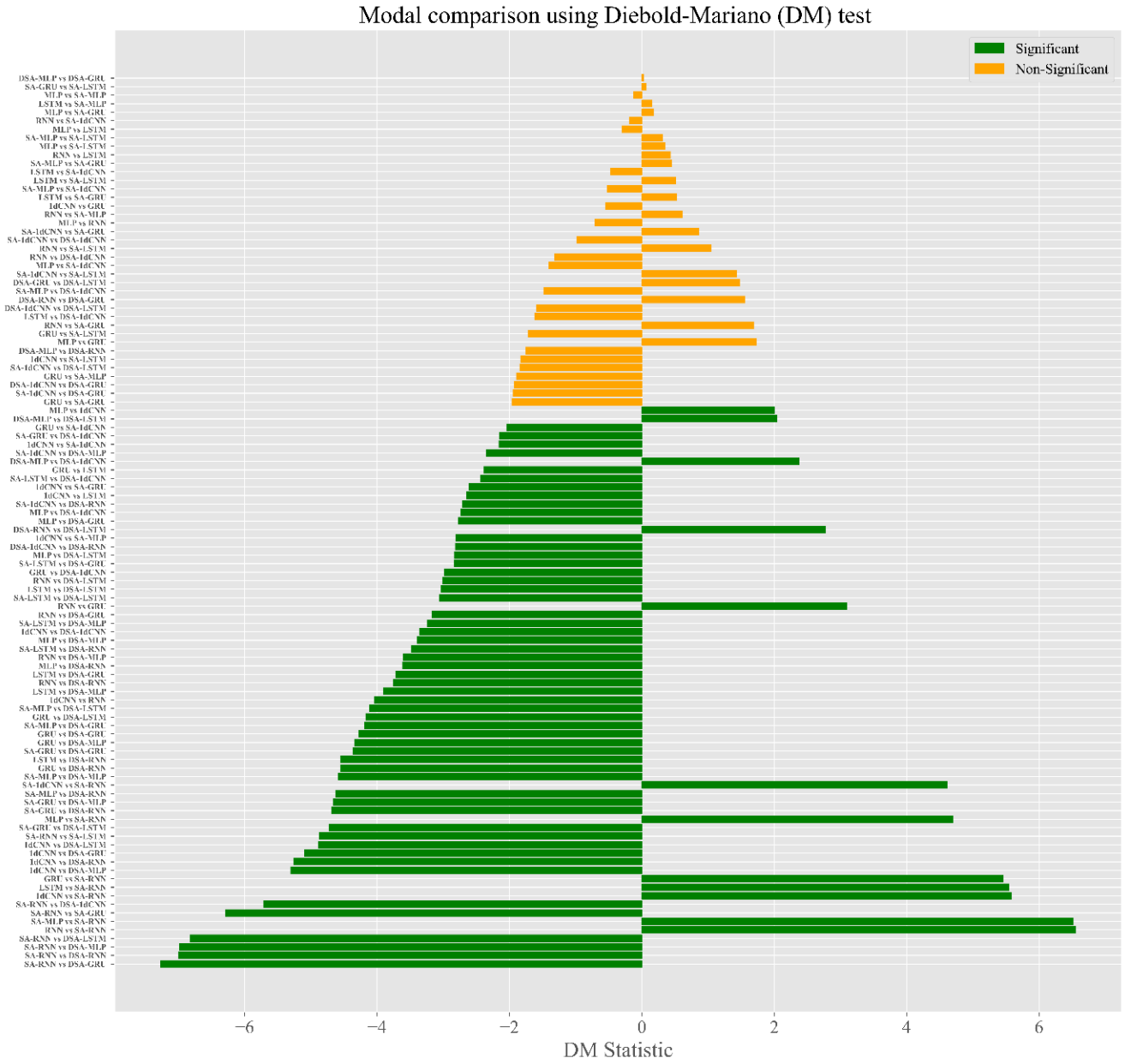

Fig. 10. DM test on Onion price series forecast

1dCNN models displayed more significant accuracy enhancements of 13.28% and 22.98%. However, the SAE-RNN model experienced a surprising deterioration in performance, with an 82.04% increase in RMSE, highlighting potential model suitability issues. Conversely, the SAE-LSTM model slightly underperformed compared to the DL LSTM model, indicating a minor decrease in performance. The transition from SAE-DL to DSAE-DL models marked a pivotal improvement in forecasting accuracy. The DSAE-MLP model, for instance, improved upon the SAE-MLP model by 37.58%, while the DSAE-GRU model outpaced the SAEGRU model by 39.13%. Notably, the DSAE-RNN model achieved the most significant accuracy leap, with a 69.15% improvement over the SAE-RNN model, showcasing the profound impact of denoising techniques. The DSAE-1dCNN and DSAE-LSTM models further solidified the superiority of DSAE-DL models with accuracy improvements of 7.85% and 29.11%, respectively, over their SAE-DL counterparts.

In terms of MAE comparison for DL versus SAE-DL models, modest improvements were noted in MAE for the MLP and GRU models, at 1.12% and 7.42% respectively, indicating a slight edge in predictive precision. The 1dCNN model exhibited a notable improvement of 28.72%, while the RNN model's performance deteriorated significantly, raising concerns about its compatibility with SAE techniques. Conversely, transitioning from DL to DSAE-DL models marked substantial advancements, with the DSAE-MLP and DSAE-GRU models achieving improvements of 42.53% and 49.84%. This trend underscores the denoising autoencoders' role in significantly refining predictive accuracy. Similarly, MAPE improvements echoed the transformative impact of DSAE-DL models. While the SAE-DL models provided mixed results, with the 1dCNN model showing a 41.29% improvement and the RNN model displaying a pronounced decline, the DSAE-DL models consistently outperformed their counterparts. The DSAE-MLP model, for instance, showed a 44.42% improvement over its DL counterpart, illustrating the profound benefits of integrating denoising mechanisms for more accurate and consistent predictions. The comparative analysis between SAE-DL and DSAE-DL models further highlighted the latter's superior performance. Notably, the DSAE-RNN model achieved an exceptional 73.43% improvement in MAE and an 81.78% improvement in MAPE when compared to the SAE-RNN model, marking the highest percentage increase across all models. These findings emphasize the denoising stacked autoencoders' capability to effectively handle noise and extract relevant features, thereby enhancing the models' forecasting accuracy.

In addition, SAEs, by leveraging unsupervised learning, can effectively learn representations and extract features from unlabeled data, making them highly valuable 23 for pre-training or initial feature learning in complex tasks. Furthermore, conventional DL models often require extensive tuning and can be prone to overfitting, especially when dealing with limited or highly dimensional data (Soui *et al.* 2020). In contrast, the layer-wise training approach of SAEs provides a more structured and potentially less prone to overfitting methodology, facilitating a more efficient learning process. Whereas, the unique advantages of denoising stacked autoencoders in enhancing forecasting accuracy and model robustness against noise, compared to traditional SAE and DL models, similar study reported in other domains by these studies (Wang *et al.* 2017). The denoising process in DSAEs helps in distinguishing the signal from the noise, thus enabling the model to learn more robust features. This attribute is particularly beneficial in the context of forecasting highly volatile agricultural commodity prices, where the data can often be contaminated with outliers or non-systematic fluctuations. In contrast, while SAEs can capture complex hierarchies of features, they lack the explicit mechanism to deal with noise, potentially leading to overfitting (Vincent *et al.* 2008). Similarly, conventional DL models, though powerful in feature extraction and representation learning, may struggle with the generalization of noisy data, impacting their forecasting accuracy and reliability.

In summary, these findings illuminate the significant accuracy enhancements achievable through the adoption of DSAE-DL models in agricultural price forecasting. By offering a robust solution to the challenges of noise and feature extraction in time series data, these models stand out as a promising direction for future research and practical application. The results advocate for the continued exploration of advanced neural network architectures, particularly those incorporating denoising techniques, to push the boundaries of predictive performance in agriculture and beyond to tackle the complexities of non-linear time series data.

6. CONCLUSION

This study significantly advances the field of agricultural commodity price forecasting, particularly relevant in agrarian economies. By comparing traditional Deep Learning (DL) models with advanced Stacked Autoencoder (SAE) and Denoising Stacked Autoencoder Deep Learning (DSAE-DL) models, our research demonstrates the pronounced superiority of DSAE-DL models in addressing the complex, non-linear nature of time series data. The integration of denoising autoencoders within deep learning architectures not only mitigates the impact of noise but also enhances feature extraction capabilities, leading to marked improvements in forecasting accuracy. Our findings reveal that, although DL models establish a robust baseline, the addition of SAE and, notably, DSAE layers

significantly elevates performance by offering more refined noise filtration and feature recognition. This comparative analysis not only highlights the benefits of incorporating advanced neural network architectures in agricultural forecasting but also sets a new standard for predictive performance in the field. The exploration underscores the transformative potential of DSAE-DL and SAEDL models, proposing a robust framework for future research and application in not just agriculture but also in sectors demanding high accuracy in time series forecasting, such as finance, energy and healthcare.

REFERENCES

Avinash, G., V., R., Ray, M., Paul, R.K., Godara, S., Nayak, G.H.H., Kumar, R.R., Manjunatha, B., Dahiya, S., Iquebal, M.A. (2024). Hidden Markov guided Deep Learning models for forecasting highly volatile agricultural commodity prices. *Applied Soft Computing*, 111557 https://doi.org/10.1016/j.asoc.2024. 111557

Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling https://doi.org/10. 48550/ARXIV.1412.3555

Haykin, s. (2009). Neural Networks and Learning Machines, 3rd edn. Pearson Education India.

Hinton, G.E., Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. *Science* 313(5786), 504-507 https:// doi.org/10.1126/science. 1127647

Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation* 9(8), 1735-1780 https://doi.org/10.1162/ neco.1997.9.8.1735

Liu, P., Zheng, P., Chen, Z. (2019). Deep Learning with Stacked Denoising Auto-Encoder for Short-Term Electric Load Forecasting. *Energies* 12(12), 2445 https: //doi.org/10.3390/ en12122445

Liu, Y., Zhao, C., Huang, Y. (2022). A Combined Model for Multivariate Time Series Forecasting Based on MLP-Feedforward Attention-LSTM. *IEEE Access* 10, 88644-88654 https://doi.org/10.1109/ ACCESS.2022.3192430

Najafi, A., Homaee, O., Jasinski, M., Golshan, M., Leonowicz, Z. (2022). Application of Extreme Learning Machine-Autoencoder to Medium Term Electricity Price Forecasting. In: 2022 IEEE International Conference on Environment and Electrical Engineering and 2022 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), pp. 1-4. IEEE, Prague, Czech Republic. https://doi.org/10.1109/EEEIC/ ICPSEurope54979.2022.9854736

Nayak, G.H.H., Alam, W., Singh, K.N., Avinash, G., Ray, M., Kumar, R.R. (2024). Modelling monthly rainfall of India through transformer-based deep learning architecture. *Modeling Earth Systems and Environment* https://doi.org/ 10.1007/s40808-023-01944-7

Oikonomidis, A., Catal, C., Kassahun, A. (2023). Deep learning for crop yield prediction: A systematic literature review. *New Zealand Journal of Crop and Horticultural Science* 51(1), 1-26 https://doi.org/10.1080/01140671.2022.2032213

Paul, R.K., Yeasin, Md., Kumar, P., Kumar, P., Balasubramanian, M., Roy, H.S., Paul, A.K., Gupta, A. (2022). Machine learning techniques for forecasting agricultural prices: A case of brinjal in Odisha, India. *PLOS ONE* 17(7), 0270553 https://doi.org/10.1371/ journal.pone.0270553

Samal, K.K.R., Babu, K.S., Das, S.K. (2021). Temporal convolutional denoising autoencoder network for air pollution prediction with missing values. *Urban Climate* 38, 100872 https://doi.org/10.1016/j.uclim.2021.100872

Soui, M., Smiti, S., Mkaouer, M.W., Ejbali, R. (2020). Bankruptcy Prediction Using Stacked Auto-Encoders. *Applied Artificial Intelligence* 34(1), 80-100 https://doi.org/10.1080/08839514.2019.1691849

Sun, F., Meng, X., Zhang, Y., Wang, Y., Jiang, H., Liu, P. (2023). Agricultural Product 26 Price Forecasting Methods: A Review. *Agriculture* 13(9), 1671 https:// doi.org/10.3390/ agriculture13091671

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning - ICML '08, pp. 1096-1103. ACM Press, Helsinki, Finland. https://doi.org/10.1145/1390156.1390294

Wang, L., Zhang, Z., Chen, J. (2017). Short-Term Electricity Price Forecasting With Stacked Denoising Autoencoders. *IEEE Transactions on Power Systems* 32(4), 2673-2681 https://doi.org/10.1109/TPWRS.2016.2628873

Xiong, T., Li, C., Bao, Y. (2018). Seasonal forecasting of agricultural commodity price using a hybrid STL and ELM method: Evidence from the vegetable market in China. *Neurocomputing* 275, 2831-2844 https://doi.org/10.1016/j. neucom.2017.11.053