EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

Constructing an Intelligent Environmental Monitoring and Forecasting System: Fusion of Deep Neural Networks and Gaussian Smoothing

Ruey-Chyi Wu[1,*]

[1]Bachelor Degree Program of Digital Marketing, National Taipei University, Taipei, Taiwan

Abstract

To enhance monitoring of environmental indicators such as temperature, humidity, and carbon dioxide (CO_2) concentration in data centers, this study evaluates various deep neural network (DNN) models and improves their forecast accuracy using Gaussian smoothing. Initially, multiple DNN architectures were assessed. Following these evaluations, the optimal algorithm was selected for each indicator: Convolutional Neural Network (CNN) for temperature, Long Short-Term Memory (LSTM) for humidity, and a hybrid model combining Long Short-Term Memory and Gated Recurrent Unit networks (LSTM-GRU) for CO_2 concentration. These models underwent further refinement through Gaussian smoothing and re-training to enhance their forecasting capabilities. The results demonstrate that Gaussian smoothing significantly enhanced forecast accuracy across all indicators. For instance, R^2 values notably increased: the temperature forecast improved from 0.59925 to 0.98012, humidity from 0.63305 to 0.99628, and CO_2 concentration from 0.71204 to 0.99855. Thus, this study highlights the potential of DNN models in environmental monitoring after Gaussian smoothing, providing precise forecasting tools and real-time monitoring support for informed decision-making in the future.

Keywords: Environmental monitoring Forecast, Gaussian smoothing, CNN, LSTM, GRU, Temperature, Humidity, CO_2 concentration.

Received on 04 03 2024, accepted on 30 06 2024, published on 07 08 2024

doi: 10.4108/eetiot.6519

*Corresponding author. Email: rueychyiwu@gmail.com

1. Introduction

In recent years, deep neural network (DNN) models have made substantial advancements across diverse fields. Accurate forecasting in environmental monitoring is especially critical for enabling proactive decision-making. This study seeks to evaluate and optimize various DNN architectures tailored for forecasting key environmental indicators: temperature, humidity, and carbon dioxide (CO_2) concentration. Through the integration of Gaussian smoothing techniques, the objective is to augment the accuracy and dependability of forecasts, thereby enhancing overall environmental monitoring strategies.

The motivation behind this study arises from the pressing need to enhance the accuracy and reliability of environmental monitoring systems. Traditional methods often face challenges in accurately capturing and forecasting complex environmental dynamics, characterized by nonlinear relationships, sudden environmental changes, and the requirement for real-time responsiveness. Leveraging the capabilities of DNNs to capture time dependencies and nonlinear relationships presents a promising approach to improving environmental forecast accuracy. DNN models such as Convolutional Neural Networks (CNNs), Long Short-Term Memory Networks (LSTMs), and their bidirectional variants each offer unique advantages in capturing short-term fluctuations and long-term trends, providing flexibility in modeling various features of environmental data.

The primary objective of this study is threefold: firstly, to evaluate and compare the performance of various DNN architectures in forecasting environmental monitoring indicators (temperature, humidity, and CO_2 concentration); secondly, to optimize these forecasts using Gaussian smoothing techniques and validate their effectiveness; and thirdly, to deploy the optimized models in existing dashboards and messaging systems.

(i) Evaluation of DNN Models: This study systematically evaluates multiple DNN architectures, including Simple Recurrent Neural Networks (SRNN), Bidirectional Simple Recurrent Neural Networks (BiSRNN), Convolutional Neural Networks (CNN), Long Short-Term Memory Networks (LSTM), Bidirectional Long Short-Term Memory Networks (BiLSTM), Gated Recurrent Units (GRU), Bidirectional Gated Recurrent Units (BiGRU), LSTM-GRU hybrids, and CNN-LSTM-GRU hybrids. The performance of these models in accurately forecasting environmental indicators is assessed.
(ii) Optimization using Gaussian Smoothing: After identifying the optimal model for each environmental indicator, Gaussian smoothing techniques are applied. Gaussian smoothing enhances forecast stability by applying weighted averages to neighboring data points, thereby reducing noise. This step helps mitigate the impact of outliers and abrupt changes in environmental data, ultimately improving the robustness of the models.
(iii) Deployment of Research Outcomes: The optimized models will be deployed in existing dashboards and messaging systems.

By achieving these goals, this study aims to provide a robust framework for environmental monitoring in data centers, ensuring optimal conditions and efficient energy use, ultimately preventing hardware failures and operational issues.

2. Literature Review

This section delves into pertinent literature concerning a variety of topics including deep learning neural networks, cloud computing, and Internet of Things (IoT).

2.1. IoT and Environmental Control Systems with Machine Learning

Ahmed et al. [1] developed an air quality monitoring system based on the Internet of Things (IoT), emphasizing the importance of real-time data collection and predictive analytics. They summarized advancements and application cases related to this technology. Sarun Duangsuwan et al. [2] introduced the use of Low Power Wide Area Network (LPWAN) and Narrowband Internet of Things (NB-IoT) technologies for air pollution monitoring in smart cities in Thailand, demonstrating the potential of these technologies in real-time monitoring. Ting Yang et al. [3] conducted a comprehensive review of recent advancements in smart environmental monitoring systems, focusing particularly on the monitoring of air quality, water pollution, and radiation pollution. The paper extensively analyzed the sensors employed, machine learning techniques, and classification methods used in these systems, and proposed further research directions. Mengda Jia et al. [4] used Extreme Learning Machine (ELM) to predict air pollution concentrations in Hong Kong, demonstrating the effectiveness of machine learning techniques in improving pollution prediction accuracy. The study highlighted ELM's advantages in handling large-scale data and enhancing prediction accuracy. Ullo et al. [5] conducted a comprehensive review of recent advancements in smart environmental monitoring systems, focusing particularly on the monitoring of air quality, water pollution, and radiation pollution. The paper extensively analyzed the sensors utilized, machine learning techniques, and classification methods employed in these systems, and proposed recommendations for further research.

2.2. CNN

CNNs, initially proposed by LeCun et al. for handwritten digit recognition, demonstrated superior performance in image processing tasks [6]. Alex Krizhevsky et al. [7] applied deep CNNs to the ImageNet dataset, significantly improving the accuracy of image classification. Karen Simonyan et al. [8] further increased the depth of CNNs, demonstrating that deeper networks could achieve better performance. Kaiming He et al. [9] introduced the residual learning framework (ResNet), effectively addressing the degradation problem of and advancing the field of deep learning. Additionally, Goodfellow et al.'s comprehensive book comprehensively summarizes the theory and applications of deep learning [10].

2.3. LSTM

LSTM were initially proposed by Hochreiter and Schmidhuber to address the challenges faced by traditional RNNs in handling long-term dependencies [11]. Gers et al. [12] further expanded LSTM by introducing the concept of "Forget Gates", enhancing its performance in processing long sequential data. Graves et al. [13] demonstrated LSTM's application in speech recognition, showcasing its robust performance in practical applications. Sutskever et al. [14] introduced the LSTM-based Sequence-to-Sequence (Seq2Seq) learning model, which achieved significant success in tasks such as machine translation. Additionally, Klaus Greff et al. [15] conducted a comprehensive comparison and analysis of various LSTM variants, providing valuable reference material for LSTM research. Kun Wang et al. [16] proposed a hybrid flow model using an improved CNN classification algorithm and key frame extraction mechanism to reduce video data load and network congestion. Experimental results showed its effectiveness in reducing video size while maintaining stable Quality of Experience.

2.4. GRU

Accurate prediction of traffic flow can significantly improve quality of life, with IoT devices capturing real-time data. The GRU model has shown promising results in traffic flow prediction, albeit with complexities in hyperparameter and window optimizations. A recent study introduces a new algorithm aimed at enhancing prediction accuracy and stability, reducing errors by 4.5% [17]. Predicting runoff is crucial for flood prevention. LSTM and GRU models excel in automatically filtering redundant information. Research indicates that both LSTM and GRU models outperform Artificial Neural Networks (ANN), with GRU demonstrating shorter training times [18].

2.5. Gaussian Smoothing

Gaussian Smoothing is a crucial technique in image processing and signal processing used to eliminate noise and enhance main structures. The concept of scale-space filtering, where Gaussian smoothing plays a fundamental role in handling structures of different scales, was first introduced by Andrew P. Witkin [19]. Marr et al. [20] incorporated Gaussian smoothing as a preprocessing step in their edge detection theory to improve edge detection accuracy. Koenderink et al. [21] demonstrated the multiscale representation of image structures, highlighting the critical role of Gaussian smoothing. For achieving full automation in the segmentation process, Jaya Brindha G. et al. [22] proposed the use of Gaussian process-based regression techniques to estimate tuning parameters required for morphological processing. This method was tested on sunflower leaf and ImageCLEF datasets, showing potential for automation in leaf segmentation processes.

Gaussian smoothing is a technique widely used in signal processing and image processing to smooth data and reduce noise. Gaussian smoothing is implemented through a convolution operation, where data I is convolved with a Gaussian kernel G. For the one-dimensional case, the formula is [20][23][24]:

$$I_{smoothed}(x) = (I * G)(x) = \int_{-\infty}^{\infty} I(t)G(x-t)\,dt \quad (1)$$

For the two-dimensional case, applied in image processing, the Gaussian smoothing formula is:

$$I_{smoothed}(x,y) = (I * G)(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(t,s)G(x-t,y-s)\,dt\,ds \quad (2)$$

where the Gaussian kernel G is defined as:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (3)$$

- *G(x)* is the smoothed output
- σ is the standard deviation of the Gaussian distribution

For the two-dimensional case, the Gaussian kernel formula is:

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (4)$$

3. Research Methodology

Here are the steps of the research methodology, as depicted in Figure 1:

(i) Problem Definition: Define the issues that need to be addressed concerning the management of critical environmental monitoring resources.
(ii) Data Collection and Preprocessing: Gather data on temperature, humidity, and carbon dioxide (CO_2) concentration in the environment. Clean and transform the data to prepare it for subsequent analysis and modeling steps.
(iii) Evaluating DNN Models: Assess various DNN architectures to preliminarily select the optimal models for training.
(iv) Initial Selection of Optimal Model Algorithm for Gaussian Smoothing: Based on the optimal performing algorithms identified in the previous step, apply Gaussian smoothing preprocessing as necessary and retrain the models.
(v) Deploying to a Visualization Dashboard and Message Push Notifications: Send the time-series forecast waveforms generated by the trained models to visualization dashboard systems and relevant stakeholders. This serves as proactive measures for decision-making.

These steps outline a structured approach to enhancing environmental monitoring through data-driven methodologies, DNNs, and Gaussian smoothing techniques.

4. System Implementation and Evaluation

The implementation procedure of the system in this study is detailed in the following subsections.

4.1. Problem Definition, Data Collection, and Preprocessing

This study aims to address the following questions:

(i) Which DNN architecture performs optimally in forecasting environmental monitoring indicators?
(ii) How does Gaussian smoothing enhance the forecast accuracy of DNN models?

This study explores key indicators in the environmental control system of data center to forecast future operational dynamics. The data sources include:

(i) Temperature sensor: Monitors the air temperature in the server farm of the data center. It is measured in Celsius to indicate changes in the environmental temperature.
(ii) Humidity sensor: Measures the air humidity in the server farm of the data center. It is measured in percentage to indicate the water vapor content in the air.
(iii) Air quality sensor: Detects the concentration of CO_2 in the office air. It is measured in ppm (parts per million) to indicate the concentration of carbon dioxide in the air.

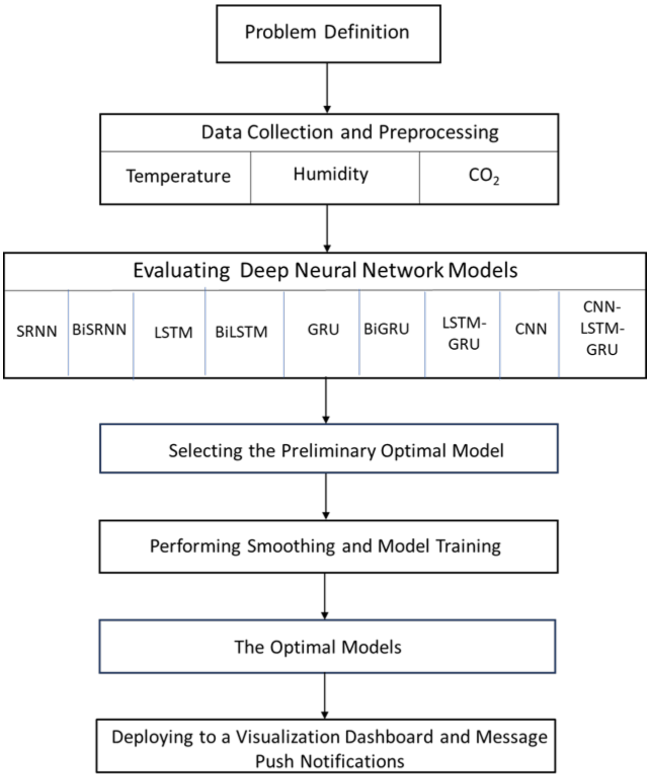


Figure 1. Research Architecture

The normal ranges for temperature, humidity, and CO_2 in a typical data center are generally defined as follows:

(i) Temperature: Ideally maintained between 20 to 25 degrees Celsius to ensure stable and efficient operation of equipment.
(ii) Humidity: Controlled within a relative humidity range of 40% to 60%. Lower than 40% may lead to static electricity issues and equipment damage, while higher than 60% can cause equipment failures and corrosion.
(iii) CO_2 Concentration: Higher concentrations, tpm), can significantly impact the comfort and health of office personnel, although their effect on equipment is relatively minor.

Alarm lights and messages should be triggered immediately for values above or below the normal range as mentioned above.

4.2. Evaluate DNN Models

This study employs the Python programming language, utilizing libraries including NumPy, Pandas, Matplotlib, SciPy, Scikit-learn, and TensorFlow. The implementation is conducted on the Colab Notebooks cloud platform [25].

This study adopts commonly used regression metrics to evaluate CNNs and RNNs.

(i) Mean Absolute Error (MAE) [26]
- MAE is a metric used to evaluate the accuracy of a model's forecasts. It measures the average absolute difference between the predicted values and the actual values.
- Unlike Root Mean Squared Error, MAE does not square the errors, thus it is not sensitive to outliers.
- A smaller MAE indicates that the model's forecasts are more accurate.

(ii) Root Mean Square Error (RMSE) [27]
- RMSE is a metric used to assess the accuracy of model forecasts by measuring the differences between predicted values and actual values. It is calculated by taking the square root of the average of the squared differences between predicted and actual values.
- RMSE provides a measure of the average size of errors in model forecasts. It is sensitive to outliers.
- A lower RMSE indicates that the model's forecasts are closer to the actual values, reflecting better accuracy.

(iii) Coefficient of determination (R-squared or R^2) [28]
- R^2 is a statistical measure used to assess the goodness of fit of a regression model to the data.
- It ranges from 0 to 1 and represents the proportion of the variance in the dependent variable that is predictable from the independent variables.
- R^2 indicates the model's ability to explain the variability in the data. A value closer to 1 indicates a better fit of the model to the data, whereas a value closer to 0 indicates a poorer fit.

To forecast future server room temperature, humidity, and office CO_2 concentration, we explored algorithms such as SRNN, BiSRNN, CNN, LSTM, BiLSTM, GRU, BiGRU, LSTM-GRU, and CNN-LSTM-GRU during the model selection process. The following presents the loss curves for each training model used to assess fit on the training set, evaluate model overfitting on the validation set, and assess model generalization on the test set. We utilized metrics including MAE, RMSE, and R^2 to compare error values across the training, validation, and test sets. Model performance comparison was based on the majority voting principle.

4.3. Model Selection Process and Criteria

The following steps outline the model selection process before applying Gaussian smoothing:

(i) Model Exploration: Various DNN architectures, such as SRNN, BiSRNN, CNN, LSTM, BiLSTM, GRU, BiGRU, LSTM-GRU, and CNN-LSTM-GRU, were tested to identify the most effective model for forecasting temperature, humidity, and CO_2 concentration.
(ii) Loss Curve Analysis: Analysis of the loss curves of each model on the training, validation, and test sets was conducted to evaluate their performance. This involved monitoring the reduction of loss values with each iteration to ensure the models learned correctly and converged.
(iii) Validation Curve Evaluation: To check for overfitting, we compared the loss curves on the training and validation sets. Through validation curves, we can comprehend how well each model generalizes to unseen data during training.
(iv) Test Set Evaluation: Final evaluation was conducted on the test set to assess the models' generalization ability. This step ensured that the selected models performed well not only on the training and validation sets but also on new, unseen data.
(v) Performance Metrics: MAE, RMSE, and R^2 were used as primary metrics to compare each model's performance across the training, validation, and test sets.
(vi) Majority Voting Principle: Based on these evaluation metrics, we selected the model that consistently and significantly outperformed others across all sets as the optimal forecast model. Specifically, models ranking higher in MAE, RMSE, and R^2 were given priority.

Gaussian smoothing is a commonly used technique in signal processing and image processing to reduce noise and smooth data. Here are the specific technical details of Gaussian smoothing:

(i) Gaussian Kernel: The core of Gaussian smoothing involves generating a smooth kernel function using the Gaussian distribution. This kernel function can be a multidimensional matrix or a one-dimensional vector, characterized by a peak at its center that decreases in amplitude as the distance from the center increases. The standard deviation (σ) of the Gaussian function determines the strength of the smoothing; a larger σ results in a more pronounced smoothing effect.
(ii) Application of Deep Neural Networks (DNN): In certain applications, Gaussian smoothing can be implemented using deep neural networks. By training a deep neural network and applying it to signal or image processing, especially in scenarios requiring nonlinear smoothing, noise can be effectively handled while preserving the essential features of the signal. Deep neural networks typically consist of multiple layers of neurons, optimized using backpropagation to minimize the error between the smoothed signal and the original signal.
(iii) Selection of Kernel Size: When applying Gaussian smoothing, it is crucial to choose an appropriate kernel size (typically indicated by the standard deviation σ of the Gaussian kernel). A kernel that is too small may not effectively smooth out noise in the signal, while a kernel that is too large may blur the details of the signal. The choice of kernel size should be based on the specific requirements of the application and the characteristics of the signal.
(iv) Factors Affecting Smoothing Effectiveness: Besides kernel size, the effectiveness of smoothing is influenced by the characteristics of the signal itself and the level of noise present. In some cases, multiple applications of Gaussian smoothing may be necessary to further improve the smoothing effect, or adjustments to the kernel size may be made based on the particular properties of the signal.

The following steps outline the process of retraining the selected optimal model after applying Gaussian smoothing preprocessing:

(i) Gaussian Smoothing Preprocessing: Apply Gaussian smoothing to the respective training time series data by adjusting the standard deviation (σ) of the Gaussian kernel. This process aims to reduce noise and outliers, thereby enhancing the smoothness of the data.
(ii) Loss Curve Analysis: Analyze the loss curves of each model across the training, validation, and test sets to evaluate their performance. This involves tracking the reduction of loss values with each iteration to ensure correct learning and convergence of the models.
(iii) Validation Curve Evaluation: Compare the loss curves between the training and validation sets to detect potential overfitting. Validation curves provide an understanding of how well each model generalizes to unseen data during training.
(iv) Test Set Evaluation: Perform a final evaluation on the test set to assess the model's ability to generalize. This step ensures that the selected model performs well not only on the training and validation sets but also on new, unseen data.
(v) Performance Metrics: Use MAE, RMSE, and R^2 as primary metrics to demonstrate the performance of each optimal model across the training, validation, and test sets.

4.3.1. Evaluation and Selection of the Optimal Training Model for Temperature

This section outlines the evaluation and selection process to identify the optimal training model for temperature forecasting in the server room. Figure 2 presents a segment of raw temperature data. Table 1 displays evaluation metrics

(MAE, RMSE, and R^2) for the training results of various models before applying Gaussian smoothing.

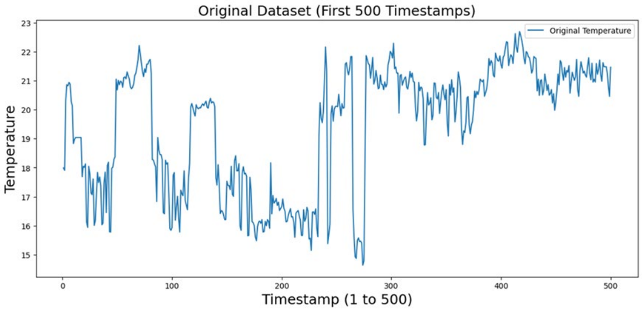
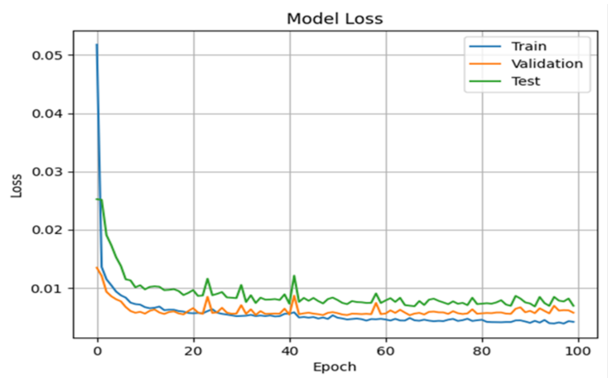


Figure 2. Segment of the temperature data graph from the dataset

Table 1. Temperature model test scores and rankings without Gaussian smoothing

Temperature Models	MAE	RMSE	R^2	Rank
SRNN	0.60472	0.83587	0.57384	4
BiSRNN	0.61693	0.88356	0.52382	9
LSTM	0.57689	0.83744	0.57224	5
BiLSTM	0.60006	0.84830	0.56108	8
GRU	0.58287	0.84599	0.56346	6
BiGRU	0.58444	0.83194	0.57784	2
LSTM-GRU	0.58794	0.84653	0.56290	7
CNN	0.59655	0.81056	0.59925	1
CNN-LSTM-GRU	0.58533	0.83523	0.57450	3

The evaluation results underscored the CNN model's exceptional performance in accurately capturing temperature dynamics. Therefore, the CNN model was chosen as the optimal model for temperature forecasting. Figure 3 illustrates the CNN model's loss curves, validation set performance, and test set generalization.



Training Metrics:
Train MAE: 0.04329352182261853
Train RMSE: 0.0623077775426986
Train R-squared: 0.8946220471756454

Test Metrics:
Test MAE: 0.06118459821917203
Test RMSE: 0.08313481118093623
Test R-squared: 0.5992549023367485

Validation Metrics:
Validation MAE: 0.055805804218133456
Validation RMSE: 0.07563311897974542
Validation R-squared: 0.7582077873388324

Test Metrics in the original scale:
MAE: 0.5965499147956753
RMSE: 0.8105644550743142
R-squared: 0.5992548567922017

Figure 3. CNN training performance without Gaussian smoothing

Figure 4 illustrates the CNN model's forecasts for the next 72 hours of temperature without applying Gaussian smoothing. This graph visually represents how the CNN model predicts future temperature changes in the server room based on its training data. The forecasts are plotted alongside the actual observed values for comparison. Examining this figure allows us to assess the initial performance of the CNN model and identify any discrepancies or areas for improvement before applying Gaussian smoothing to enhance forecast accuracy.

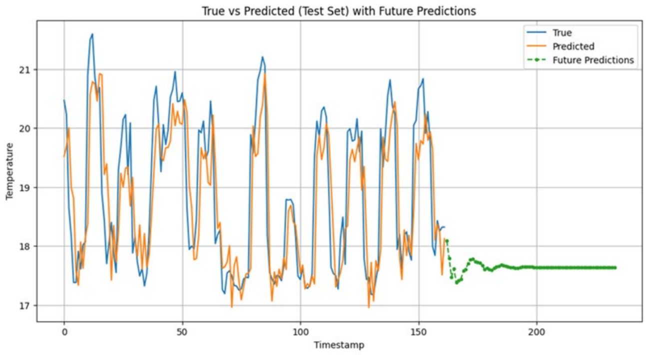


Figure 4. CNN model's future 72-hour forecast without Gaussian smoothing

With the CNN algorithm identified as the optimal model for temperature forecasting, we proceeded to preprocess the raw temperature data using Gaussian smoothing. Figure 5 illustrates the temperature data after Gaussian smoothing. Figure 6 showcases the CNN model's fitting and generalization performance across the training, validation, and test datasets, supported by actual experimental data. Furthermore, Figure 7 presents the forecast results for the next 72 hours.

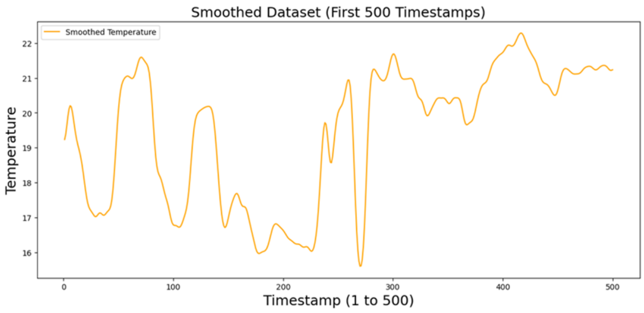


Figure 5. The temperature data after Gaussian smoothing processing

4.3.2. Evaluation and Selection of the Optimal Training Model for Humidity

In this section, the study outlines the process of evaluating and selecting the optimal training model for forecasting humidity in the server room. Figure 8 displays a segment of the raw humidity data. Table 2 lists the evaluation metrics (MAE, RMSE, and R^2) for the training results of various models before applying Gaussian smoothing.

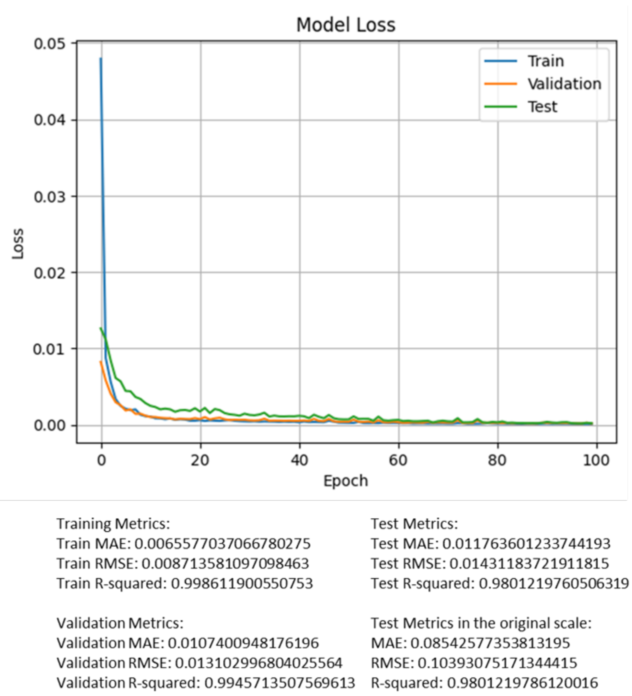


Figure 6. The performance of the CNN model with Gaussian smoothing

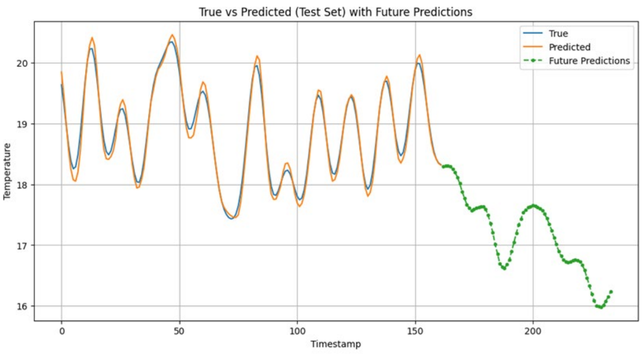


Figure 7. CNN model's forecast for the next 72 hours with Gaussian smoothing

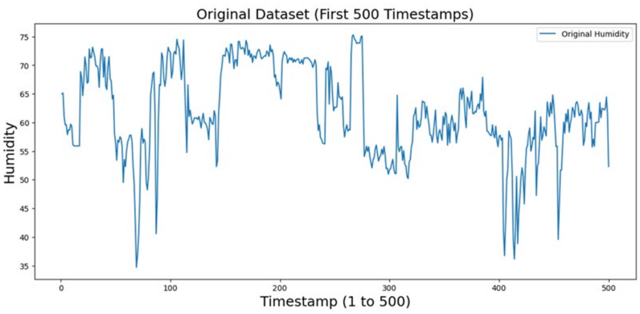


Figure 8. Segment of the humidity data graph from the dataset

In statistics, R^2 is used to assess how well a model fits the observed data. It ranges from 0 to 1, where 0 indicates that the model does not explain any variance in the target variable, and 1 indicates a perfect explanation of all variance. However, under certain conditions, R^2 can be negative. Negative R^2 values typically occur when the model's predictive performance is worse than simply taking the mean of the data. This can happen due to overfitting or using an inappropriate model, indicating extremely poor model performance. While negative R^2 values are uncommon, they can occur under specific circumstances. When evaluating models, it's crucial to recognize this possibility and consider other evaluation metrics to comprehensively assess model performance [29][30]. Therefore, if the R^2 of the model training results is negative, it will be discarded and not included in the ranking order.

Based on the comprehensive comparison of MAE, RMSE, and R^2 from Table 2, it is determined that the LSTM algorithm is the optimal model for temperature forecasting. Figure 9 presents the loss curve and evaluation metrics of the LSTM model trained without Gaussian smoothing. Figure 10 displays the LSTM model's forecasts for the next 72 hours without Gaussian smoothing.

Table 2. Humidity model test scores and rankings without Gaussian smoothing

Humidity Models	MAE	RMSE	R^2	Rank
SRNN	3.79865	5.13890	0.59811	3
BiSRNN	3.88883	5.19861	0.58872	4
LSTM	3.66924	4.95457	0.62666	1
BiLSTM	3.22688	3.73190	-1.42117	--
GRU	3.67954	4.95869	0.62604	2
BiGRU	3.48140	4.00845	-1.79331	--
LSTM-GRU	4.13368	4.76696	-2.95047	--
CNN	5.58827	7.21441	0.20842	5
CNN-LSTM-GRU	7.24145	8.14731	-10.53970	--

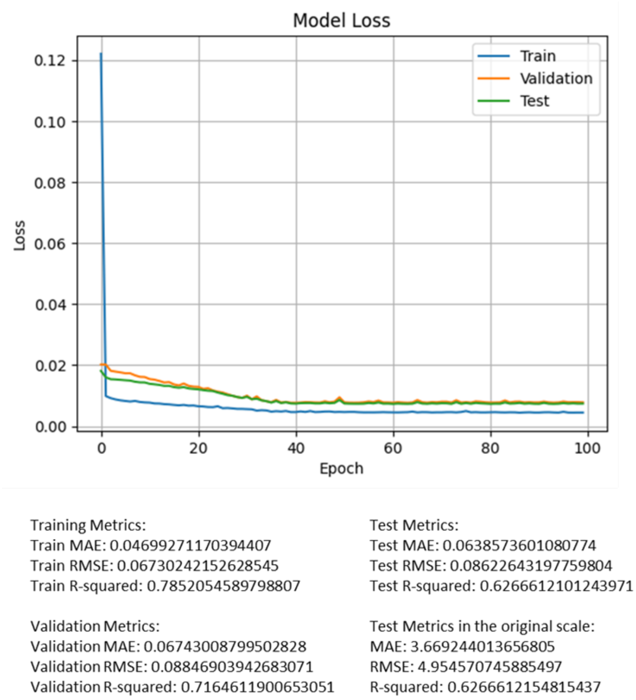


Figure 9. LSTM training performance without Gaussian smoothing

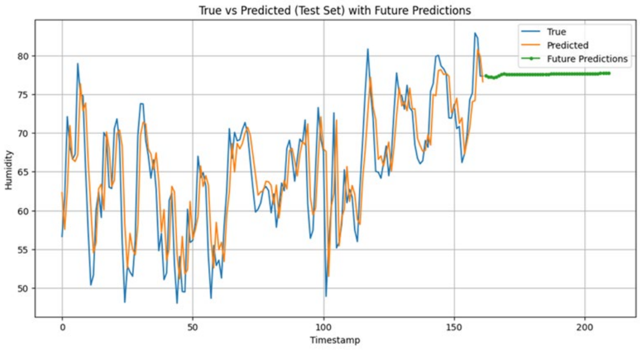


Figure 10. LSTM model's future 72-hour forecast without Gaussian smoothing

Due to the LSTM algorithm being the optimal model for humidity forecasting, a portion of the smoothed preprocessed data is shown in Figure 11. Figure 12 presents the LSTM model's training results, validation results, and test set outcomes, along with the model's loss graph. Figure 13 displays the forecast for the next 72 hours.

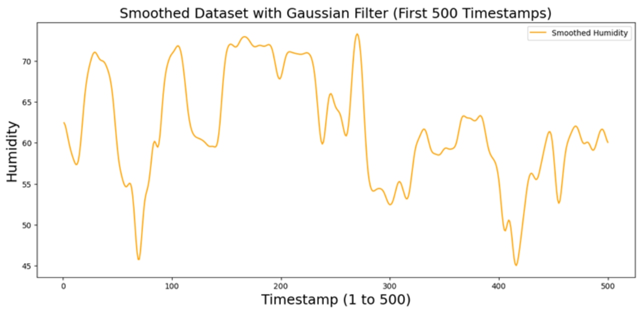


Figure 11. The humidity data after Gaussian smoothing processing

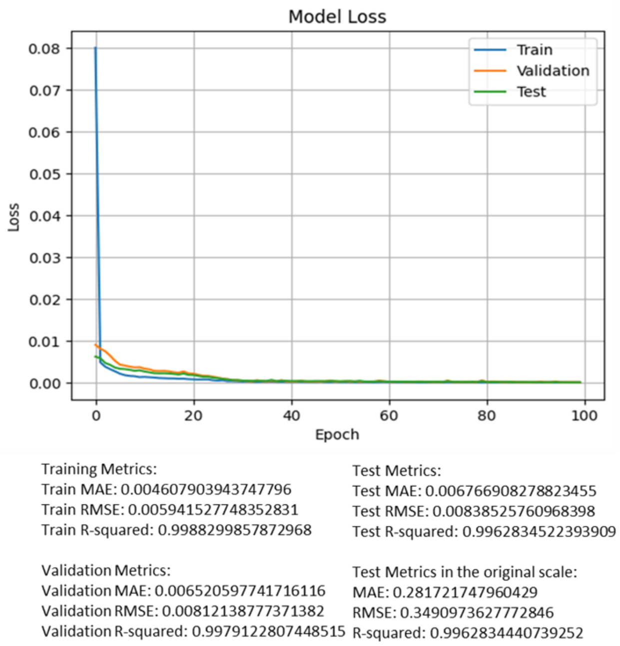


Figure 12. The performance of the LSTM model with Gaussian smoothing

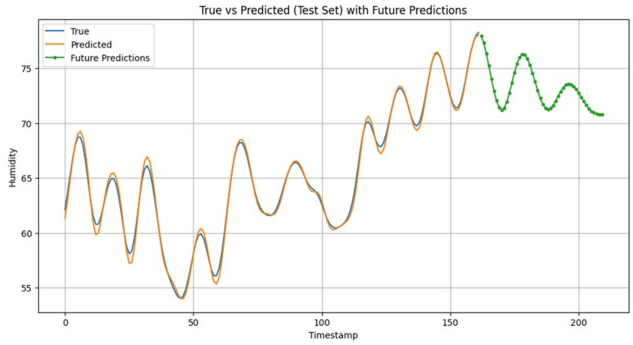


Figure 13. LSTM model's future 72-hour forecast with Gaussian smoothing

4.3.3. Evaluation and Selection of the Optimal Training Model for CO_2 Concentration

Figure 14 displays a portion of the raw CO_2 concentration data. Table 3 lists the evaluation metrics (MAE, RMSE, and R^2) for the training results of each model, showing their performance before undergoing Gaussian smoothing.

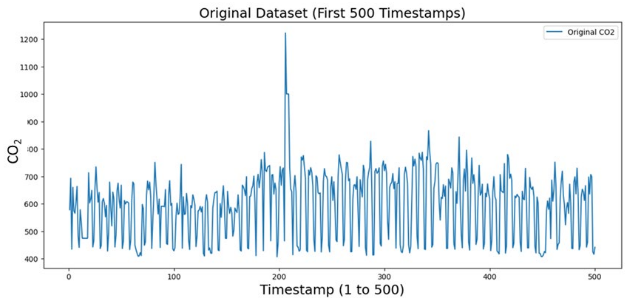


Figure 14. Segment of the CO_2 concentration data graph from the dataset

Table 3. CO_2 model test scores and rankings without Gaussian smoothing

CO_2 Models	MAE	RMSE	R^2	Rank
SRNN	91.14253	129.48040	0.55257	9
BiSRNN	87.09826	113.81699	0.58386	7
LSTM	82.50246	108.72610	0.68451	2
BiLSTM	88.85571	119.68841	0.61768	8
GRU	80.52616	109.93705	0.67744	3
BiGRU	86.32930	116.62307	0.63701	6
LSTM-GRU	76.59286	103.87359	0.71204	1
CNN	87.04390	113.81699	0.65472	5
CNN-LSTM-GRU	79.52342	112.31937	0.66331	4

After synthesizing the MAE, RMSE, and R^2 from Table 3 to compare the training results of each model, the LSTM-GRU algorithm emerged as the optimal model for forecasting CO_2 concentration. Figure 15 displays the loss curve and evaluation metrics of the LSTM-GRU model training results without Gaussian smoothing. Figure 16 illustrates the forecast graph for the next 72 hours by the LSTM-GRU model without Gaussian smoothing.

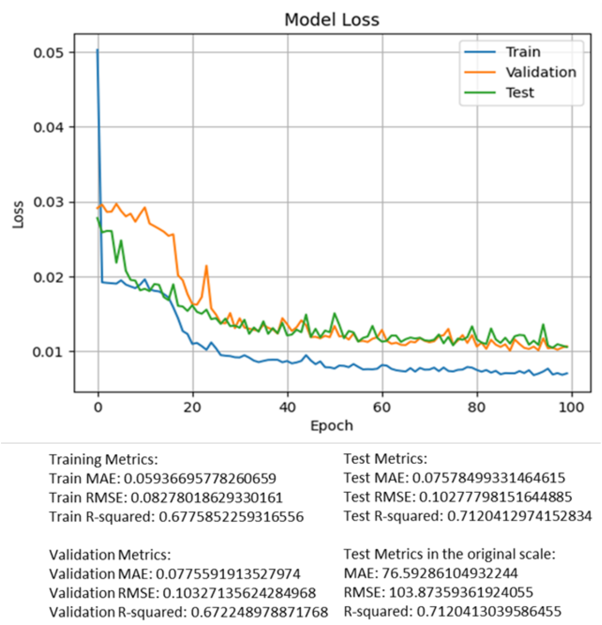


Figure 15. LSTM-GRU training performance without Gaussian smoothing

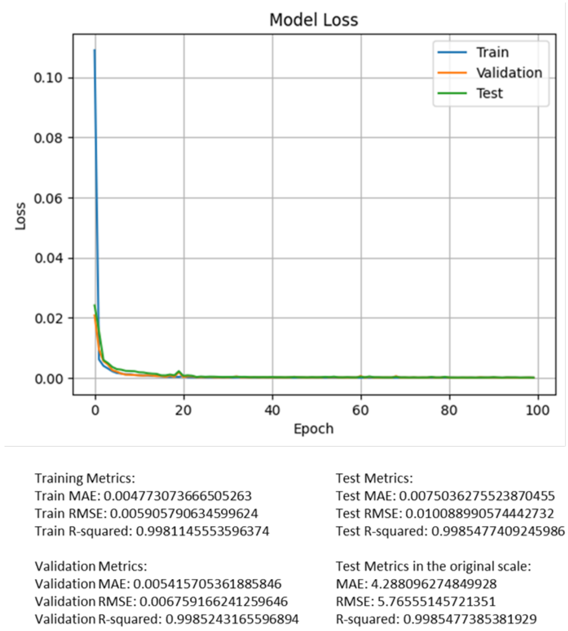


Figure 18. The performance of the LSTM-GRU model with Gaussian smoothing

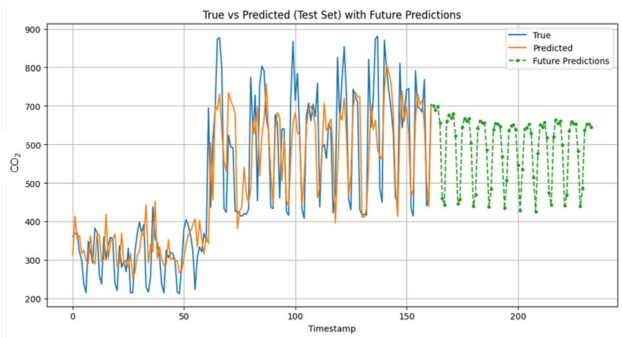


Figure 16. LSTM-GRU model's future 72-hour forecast without Gaussian smoothing

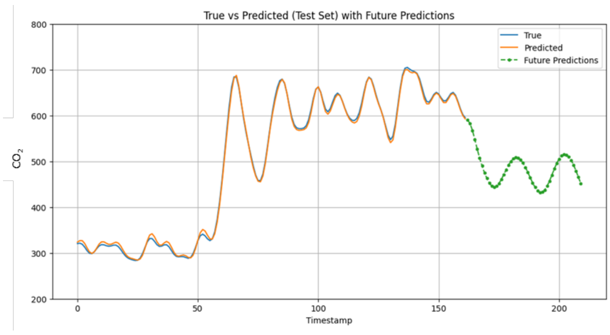


Figure 19. LSTM-GRU model's future 72-hour forecast with Gaussian smoothing

Since the LSTM-GRU hybrid algorithm is the optimal model for CO_2 concentration forecasting, Figure 17 shows the partially smoothed humidity data after preprocessing. Figure 18 presents the training results, validation results, and test set outcomes of the LSTM-GRU hybrid model, accompanied by the model's loss graph. Figure 19 displays the forecast for the next 72 hours.

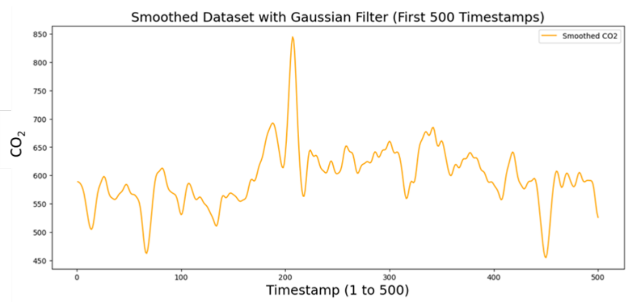


Figure 17. The CO_2 concentration data after Gaussian smoothing processing

4.3.4. Summary and Comparison of Implementation Results

After the above implementations, significant performance differences were observed in environmental monitoring indicators with and without smoothing. For instance, regarding R^2, Temperature improved from 0.59925 to 0.98012, Humidity increased from 0.63305 to 0.99628, and CO_2 concentration enhanced from 0.71204 to 0.99855. Detailed evaluation metrics (MAE, RMSE, and R^2) are summarized in Table 4.

4.4. Deploying to a Visualization Dashboard and Message Push Notifications

Deploying the trained and smoothed models to the dashboard significantly enhances the intuitive visualization of temperature, humidity, and CO_2 concentration forecasts. By integrating forecast charts into the dashboard, users can easily access and analyze these visualizations, facilitating more precise decision-making. This visualization not only enhances the vivid presentation of data but also promotes user understanding of environmental changes.

Additionally, the system possesses the capability to predict anomalies, enabling timely identification of potential risks and early risk detection. This capability is crucial for operational management, as early identification of issues can mitigate potential losses and failures. Real-time messaging software swiftly communicates forecasted information to relevant stakeholders, facilitating proactive response measures and ensuring stable system operation.

Figure 20 illustrates the integration of the trained models from this study with existing threshold settings and an alarm interface. This integration allows the models to effectively monitor environmental indicators and automatically trigger alarms when data exceeds preset thresholds, thereby enhancing system responsiveness. The models can be transmitted to visualization dashboards or messaging systems as needed, such as Line, SMS, and email, ensuring rapid communication of information to relevant personnel.

This deployment approach not only improves data accessibility and visualization effectiveness but also enhances overall operational efficiency, making the decision-making process more flexible and efficient. With real-time data feedback, managers can stay informed about environmental conditions and make adjustments proactively when necessary, further improving the precision and timeliness of operational management.

Table 4. Summary of optimal models after Gaussian smoothing

Key Metrics	Models		MAE	RMSE	R^2
Temperature	CNN	Before smoothing	0.59655	0.81056	0.59925
		After smoothing	0.08543	0.10393	0.98012
Humidity	LSTM	Before smoothing	3.63015	4.91202	0.63305
		After smoothing	0.28172	0.34910	0.99628
CO_2	LSTM-GRU	Before smoothing	76.59286	103.87359	0.71204
		After smoothing	4.28810	5.76556	0.99855

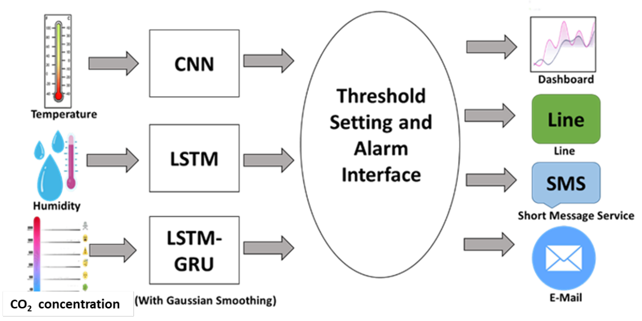


Figure 20. Deployment to a Visualization Dashboard and Push Notification Systems

5. Conclusion and Recommendations

This study successfully demonstrates the effectiveness of DNN models in enhancing the forecasting accuracy of key environmental indicators—temperature, humidity, and CO_2 concentration—through the integration of Gaussian smoothing techniques. The findings reveal significant improvements in forecast performance.

The contributions of this research are threefold:

(i) Model Evaluation and Selection: By systematically assessing various DNN architectures, this study identifies the most effective models for each environmental indicator, thus providing a valuable reference for future research in environmental forecasting.
(ii) Enhanced Forecasting through Gaussian Smoothing: The application of Gaussian smoothing not only enhances the stability of forecasts but also significantly reduces the noise in the data, leading to more reliable predictions.
(iii) Improved Operational Decision-Making: Integrating optimized models into existing dashboards and messaging systems ensures the application of research findings in real-world scenarios, thereby improving operational decision-making in data centers.

To further build upon this work, the following suggestions are proposed:

(i) Extended Model Testing: Future studies should explore additional DNN architectures and hybrid models to identify even more robust forecasting solutions for other environmental indicators or in different operational contexts.
(ii) Integration of Additional Data Sources: Incorporating external datasets, such as weather forecasts or historical environmental data, could enhance model robustness and prediction accuracy further.
(iii) Expanding Real-time Monitoring Systems: Continuous development of real-time monitoring systems that utilize these optimized models can significantly enhance operational efficiency and decision-making processes in data centers and other critical environments.
(iv) The potential of combining other smoothing techniques: Future research could explore combinations of different smoothing techniques to identify solutions that best suit varying environmental conditions and data characteristics, thereby further optimizing the performance and stability of environmental monitoring systems.

Acknowledgements.

Thank you to the editing team and reviewers for their valuable suggestions, which have contributed to improving the quality of this paper.

References

[1] Ahmed, Md Mohiuddin, Suraiya Banu, and Bijan Paul. Real-time air quality monitoring system for Bangladesh's perspective based on Internet of Things. In: 2017 3rd

International conference on electrical information and communication technology (EICT). IEEE; 2017. p. 1-5. doi:10.1109/EICT.2017.8275161
[2] Sarun Duangsuwan, Aekarong Takarn, Rachan Nujankaew, Punyawi Jamjareegulgarn. A Study of Air Pollution Smart Sensors LPWAN via NB-IoT for Thailand Smart Cities 4.0. 2018 10th International Conference on Knowledge and Smart Technology (KST); Chiang Mai, Thailand. 2018. P.206-209. doi:10.1109/KST.2018.8426195
[3] Ting Yang, Mattia Gentile, Ching-Fen Shen, Orcid, Chao-Min Cheng. Combining point-of-care diagnostics and internet of medical things (IoMT) to combat the COVID-19 pandemic. Diagnostics, 2020;10(4): 224. doi:10.3390/diagnostics10040224
[4] Mengda Jia, Ali Komeily, Yueren Wang, Ravi S. Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications. Automation in Construction, 2019; 101: 111-126. doi: 10.1016/j.autcon.2019.01.023
[5] Ullo, Silvia Liberata; SINHA, Ganesh Ram. Advances n smart environment monitoring systems using IoT and sensors. Sensors, 2020;20.11: 3113. doi:10.3390/s20113113
[6] Y. Lecun, L. Bottou, Y. Bengio an, P. Haffne. Gradient-based learning applied to document recognition, in Proceedings of the IEEE; Nov. 1998; 86(11):2278-2324. doi:10.1109/5.7267199891
[7] Alex Krizhevsky, Ilya Sutskever, G E Hinton. ImageNet classification with deep convolutional neural networks. Communications of the ACM. 2017; 60(6): 84-90. doi:10.1145/3065386
[8] Karen Simonyan, Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014. doi: 10.48550/arXiv.1409.1556
[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 770-778. https://openaccess.thecvf.com/content_cvpr_2016/2 papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf
[10] Goodfellow, Ian, Yoshua Bengio, Aaron Courville. Deep learning. MIT press; 2016.
[11] Hochreiter, Sepp, and Jürgen Schmidhuber. Long short- term memory. Neural computation. 1997; 9(8):1735-1780. doi:10.1162/neco.1997.9.8.1735
[12] Gers, Felix A., Jürgen Schmidhuber, Fred Cummins. Learning to forget: Continual prediction with LSTM. Neural computation, 2000; 12(10): 2451-2471. doi:10.1162/089976600300015015
[13] Graves, Alex, Abdel-rahman Mohamed, Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing; IEEE; 2013. p. 6645-6649. doi:10.1109/ICASSP.2013.6638947
[14] Sutskever, Ilya, Oriol Vinyals, Quoc V. Le. Sequence to sequence learning with neural networks. Advances in neural information processing systems, 2014; 27. https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f274 72c5d894ec1c3c743d2-Paper.pdf
[15] Klaus Greff, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, Jürgen Schmidhuber. LSTM: A Search Space Odyssey. in IEEE Transactions on Neural Networks and Learning Systems. 2017; 28(10):2222-2232. doi:10.1109/TNNLS.2016.2582924
[16] Kun Wang, Jun Mi, Chenhan Xu, Qingquan Zhu, Lei Shu, Der-Jiunn Deng. Real-Time Load Reduction in Multimedia Big Data for Mobile Internet. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM). 2016; 12(5s): 1-20. doi:10.1145/2990473
[17] Basharat Hussain, Muhammad Khalil Afzal, Shafiq Ahmad, Almetwally M. Mostafa. Intelligent Traffic Flow Prediction Using Optimized GRU Model. in IEEE Access. 2021; 9:100736-100746. doi:10.1109/ACCESS.2021.3097141
[18] Shuai Gao, Yuefei Huang, Shuo Zhang, Jingcheng Han, Guangqian Wang, Meixin Zhang, Qingsheng Lin. Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. Journal of Hydrology. 2020; 589: 125188. doi: 0.1016/j.jhydrol.2020.125188
[19] Andrew P. Witkin. IJCAI'83: Proceedings of the Eighth international joint conference on Artificial intelligence. Volume 2. 1983. P.1019–1022
[20] Marr, David, and Ellen Hildreth. Theory of edge detection. Proceedings of the Royal Society of London. Series B. Biological Sciences; 1980. P.187-217. doi: 10.1098/rspb.1980.0020
[21] Koenderink, Jan J. The structure of images. Biol. Cybern. 1984; 50:363–370. doi:10.1007/BF00336961
[22] Jaya Brindha G., Gopi E.S. An hierarchical approach for automatic segmentation of leaf images with similar background using kernel smoothing based Gaussian process regression. Ecological Informatics. 2021; 63: 101323. doi: 0.1016/j.ecoinf.2021.101323
[23] J. Canny. A Computational Approach to Edge Detection. in IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986; PAMI-8(6): 679-698. doi: 10.1109/TPAMI.1986.4767851
[24] Rafael Corsino González, Richard E Woods. Digital Image Processing. 3rd Edition. United States: Prentice-Hall, Inc.; 2006.
[25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Mathieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12: 2825-2830. https://arxiv.org/abs/1201.0490
[26] Yellow Zhisheng. Statistical Foundations of Machine Learning: Core Technologies Behind Deep Learning: Flag Technology Publishing; 2021.
[27] Sebastian Raschka. Python Machine Learning: Unlock deeper insights into Machine Leaning with this vital guide to cutting-edge predictive analytics. Packt Publishing Ltd; 2015. 169-198.
[28] Trevor Hastie, Robert Tibshirani, Jerome Friedman. The Elements of Statistical Learning (2nd ed.). Springer; 2013.
[29] Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining. Introduction to linear regression analysis. John Wiley & Sons; 2021.
[30] Trevor Hastie, Robert Tibshirani, Jerome Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media; 2013.