A Long Short-Term Memory Approach for Weather Forecasting

Chong Zhi Yi
School of computing
Asia Pacific Univeristy of Technology and Innovation
Kuala Lumpur, Malaysia
tp065357@mail.apu.edu.my

Phang Shea Men
School of computing
Asia Pacific Univeristy of Technology and Innovation
Kuala Lumpur, Malaysia
tp061776@mail.apu.edu.my

Lee Jian Xhe
School of computing
Asia Pacific Univeristy of Technology and Innovation
Kuala Lumpur, Malaysia
tp065191@mail.apu.edu.my

Seur Jia Yi
School of computing
Asia Pacific Univeristy of Technology and Innovation
Kuala Lumpur, Malaysia
tp072426@mail.apu.edu.my

Adeline Sneha J
Senior Lecturer/School of Computing
Asia Pacific Univeristy of Technology and Innovation
Kuala Lumpur, Malaysia
adeline.john@apu.edu.my

Kamalanathan Shanmugam
Senior Lecturer/School of Technology
Asia Pacific Univeristy of Technology and Innovation
Kuala Lumpur, Malaysia
kamalanathan@apu.edu.my

Abstract— **Weather forecasting involves using weather data to predict the future weather conditions in a specific location. Understanding the weather is important as it affects various things such as planting crops, running a business, and being prepared for emergencies. Farmers rely on precise weather forecasts to determine the best time for planting, while businesses use them to organize their operations, and communities depend on them to stay secure. This study examines the application of Long Short-Term Memory (LSTM) in forecasting weather. LSTM is a neural network known for effectively interpreting and processing sequential data, like a sequence of climate observations. By adjusting parameters such as batch size, number of epochs, and optimizer algorithm, the accuracy of the predictions changes in the updated results.**

Keywords—LSTM Network, Weather Forecasting, Neural Network, Predictive Modelling, Sequential Data.

I. INTRODUCTION

Weather, as one of the most directly perceivable changes in our daily lives, often captures widespread attention. Whether it's sunny or rainy, the weather directly impacts our activity plans, and in severe cases, it may even involve taking precautionary measures to cope with harsh conditions. However, with the continuous progress of technology, traditional weather forecasting methods are entering an era of innovation, and the emergence of deep learning technology undoubtedly injects new vitality into weather prediction.

In this captivating field, deep learning models such as Long Short-Term Memory Networks (LSTM) are gradually becoming powerful tools for forecasting future weather. Robust deep learning frameworks like LSTM and Keras provide scientists with powerful tools to build, train, and evaluate these models. By constructing, training, and evaluating these models, people can take appropriate measures to deal with unpredictable weather, avoiding being troubled by sudden weather changes in daily life. The goal of this research is to reveal the crucial role of deep learning in weather forecasting. In this research, the LSTM model is trained to analyse and process historical weather parameters, making it capable of predicting future temperature changes based on the weather conditions from the preceding hours.

II. LITERATURE REVIEW

In traditional weather forecasting methods, the indigenous communities use nature to predict the weather and make decisions, especially when it comes to farming and everyday jobs. Radeny et al. (2019) looks at how indigenous people in East Africa predict the weather through observing the sky and nature. In Africa, the rising temperatures are affecting farming and other agricultural practices. In this situation, farmers and herders rely on the knowledge from their cultural heritage to make decisions on agricultural practices. Research done by Tahiluddin et al. (2023) looks at how people in Tawi-Tawi, Philippines utilize traditional weather knowledge due to the absence of modern forecasts. Locals rely on a variety of natural signals such as cloud formation, wind direction, temperature, visibility, celestial positions and animal behaviours to forecast the weather. Similarly, Balehegn et al. (2019) also explores how Afar herders in Ethiopia depend on animals, bugs, birds, and trees to forecast weather and identify changing climate. Despite challenges in accuracy, these traditional methods persist due to the lack of access to modern alternatives.

Dharmasena (2021) studied about how people in Sri Lanka predict bad weather such as droughts, floods, storms, and rain, using old-fashioned ways of observing the nature. This method is emphasized by the locals together with their traditional farming methods to assist them in taking care of the environment and farming. The researcher also suggests that combining what indigenous people know with modern science can help to predict the weather better and manage disasters more effectively. In short, these studies highlight the importance of Indigenous Knowledge in predicting the weather and climate in many different places.

However, traditional weather forecasting methods often face limitations in terms of accuracy, adaptability, and real-time monitoring due to the complexity of the atmospheric system (Pu & Kalnay, 2018). In contrast, emerging applications of artificial intelligence (AI), such as Nvidia's FourcastNet, Google DeepMind's GraphCast, and Huawei's Pangu Weather, have brought revolutionary changes to weather forecasting by leveraging machine learning, big data analysis, and pattern recognition (Hickey, 2020). These AI applications demonstrate significant improvements in accuracy and speed compared to traditional methods (Heikkiläarchive, 2023). The use of big data technology further enhances forecasting capabilities, allowing for more detailed and comprehensive modeling of weather systems. Artificial intelligence excels in accuracy, efficiency, adaptability, and real-time monitoring, providing powerful tools for disaster prevention, agriculture, transportation, and other societal aspects influenced by weather forecasts. (Fathi et al.,2021). The development of these new technologies brings more reliable and comprehensive solutions to the field of weather forecasting.

To implement these AI applications, humans have been integrating machine learning (ML) into building and improving weather forecasting models. The benefit of ML is that it takes a more data-driven approach which increases the accuracy of results. The paper by Tiu et al. (2021) has reviewed and concluded that ML algorithms can provide great help in anticipating and responding to dengue outbreaks. An article by Wang et al. (2019) has also showcased how their proposed ML-based method "deep uncertainty quantification" have a much better accuracy when compared to numerical weather prediction in weather forecasting, with a value of 47.76% better. Another article by Bochenek and Ustrnul (2022) has discussed the usage of ML in weather forecasting. The authors have also reviewed supervised and unsupervised ML methods, then provided suggestions for determining the best methods for accurate weather forecasting. In the final article by Bhawsar et al. (2021), it has reviewed various ML and deep learning techniques utilized in weather forecasting, also listed potential issues meteorologists need to face during weather forecasting. In a nutshell, ML techniques provide better accuracy for weather forecasting results, which is an achievement for humanity.

In recent years, deep learning methodologies, notably Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Temporal Convolutional Networks (TCN), have also gained popularity in weather forecasting. Firstly, CNN, which excels in analysing and classifying 2D images, was found instrumental in improving the accuracy and efficiency in severe convection weather (SCW) phenomena as compared to traditional forecasting methods using manual observation (Ng et al., 2023; Zhou et al., 2019; Xiao et al., 2021). In research conducted by Xiao et al. (2021), a deep CNN-based model, MeteCNN was introduced, achieving a 92.68% accuracy in classifying 11 weather phenomena. Zhou et al. (2019) has also implemented a deep 2D CNN algorithm which has displayed up to 178% improvement in SCW phenomena prediction as compared to traditional methods. Next, a study by Cebeci (2019) has showcased LSTM's prowess in short-term weather forecasting. By utilizing multidimensional datasets in this research, LSTM model has achieved the highest average accuracy rate, outperforming other algorithms such as Support Vector Regression (SVR) and Multi-Layer Perceptron (MLP). Similar research conducted by Ren et al. (2021) has also highlighted the superiority of LSTM models in accuracy and timeliness in short-term local weather forecasting.
Hewage et al. (2021) has evaluated LSTM and TCN models using two regressions, namely multi-input single-output (MISO) and multi-input multi-output (MIMO). In this research, MIMO-LSTM model was identified as the optimal model, offering efficient implementation and accurate predictions up to 12 hours. Upon further research conducted by Behera et al. (2020) on the MIMO-TCN model, the TCN-based model achieved high accuracy in local forecasting for up to 9 hours. Therefore, it can be concluded that the integration of CNN, LSTM, and TCN into weather forecasting models can address the challenges in traditional methods and enhance the accuracy and efficiency of forecasting weather.

In summary, both traditional and modern approaches to weather forecasting have been significant in our daily lives. The implementation of artificial intelligence (AI) has revolutionized weather forecasting, offering more lightweighted, accurate, adaptable, and real-time solutions compared to traditional methods. The integration of machine learning (ML) algorithms further enhances forecasting accuracy, through data-driven approaches. Notably, deep learning methodologies, especially LSTM networks, has demonstrated superior performance in capturing complex weather patterns. Therefore, LSTM is suggested to be further investigated to enhance the researchers' understanding of its role in accurate and efficient weather forecasting.

III. MATERIALS

A. Dataset

This study utilized the meteorological dataset shared by ROHAN LAL KSHETRY on Kaggle. The dataset includes various meteorological information such as temperature, wind speed, and humidity, spanning from the year 2009 to 2016, providing approximately 96,000 time points of meteorological measurement data.

The research objective is to build a model using deep learning techniques to predict future temperature changes. During the data preparation phase, the meteorological data were organized and processed to transform the time series data into a supervised learning problem, enabling the model to use past hours' weather information to predict the temperature at the next time point. Long Short-Term Memory (LSTM) network was chosen as the deep learning architecture, training the model to learn patterns from historical data through 100 training cycles. Finally, the model's performance is evaluated on the test set using the Root Mean Square Error (RMSE) to measure the difference between the actual temperature and the model's predicted temperature.

B. Implementation

To implement this model, Python 3.10 programming language is required. The main libraries used include keras.models for defining the Sequential model in Python, keras.layers for various tools to build neural network layers, sklearn.metrics for evaluating model performance, Matplotlib for creating visualizations, Pandas for data processing and analysis, and Numpy for numerical computations in Python. Additionally, sklearn.preprocessing is used to provide data preprocessing tools.

In this research, the model is implemented using Google Colab. Typically, for running general machine learning tasks, it is recommended to have at least 4GB of memory, and installing Python, libraries, and storing databases may require at least 10GB of available disk space. However, in the case of using Google Colab, there is no need to worry about the hardware requirements of the local machine since the code runs on cloud resources provided by Google. Therefore, the memory and disk space of the local machine of the researchers do not directly impact the operation of the weather forecasting model in Google Colab.

IV. Methods

A. *Preproccesing*

For the input weather forecast dataset, a series of processing measures were employed to ensure consistency and diversity in the training data, while preventing overfitting. Firstly, the data format was standardized to ensure a uniform structure for all weather forecast data, including information such as date, temperature, humidity, wind speed, etc., and to maintain consistency in the model's input data. For potential missing values, appropriate handling methods were applied, such as mean or median imputation, to ensure the integrity and availability of the dataset. When dealing with time series data, extraction of time features, including year, month, day, hour, etc., was performed to assist the model in capturing time-related patterns effectively. Numerical features underwent standardization to scale them to a similar range, avoiding the impact of differences between various features on the model's training.

B. *Model Architecture*

Using the Keras Sequential model, the architecture involved the stacking of LSTM layers, fully connected layers, and an output layer. The LSTM layer comprised 30 units, responsible for handling long-term dependencies in time series. The fully connected layer consisted of 256 nodes and utilized the ReLU activation function, with a Dropout layer to prevent overfitting. The final output layer, designed for regression tasks, contained one node with a linear activation function. The entire model was compiled with mean squared error loss function and the Adam optimizer. This structure enabled the neural network to comprehend patterns in time series for accurate temperature predictions during testing.

C. *Model Training*

The model underwent training using the mean squared error loss function and the Adam optimizer. Throughout the training process, the model adjusted weights and biases through multiple iterations to minimize the loss function, enhancing accuracy in temperature predictions. Training progress was monitored by observing changes in training loss and validation loss through visualizations. Finally, the model was evaluated by calculating the Root Mean Square Error (RMSE) to measure the difference between actual temperature and model-predicted temperature.

D. *Evaluation*

The primary metric for evaluating the model's prediction accuracy on the test set was the Root Mean Square Error (RMSE). RMSE serves as an indicator of the difference between the model's predicted results and the actual observed values. It involves summing the squares of prediction errors, averaging them, and taking the square root to provide a more interpretable measure of error. By computing and outputting RMSE, a clear understanding of the model's accuracy in predicting actual temperatures and the overall level of error between predicted results and real values is obtained.

V. Algorithm Implementation

The selected algorithm for weather forecasting in this study is Long Short-Term Memory (LSTM), which is a variant of Recurrent Neural Network (RNN) well-suited in analysing time-series data (Cebeci, 2019; Hewage et al, 2021; Ren et al., 2021). In LSTM, each block comprises of three crucial multiplicative units: the input gate, which receives input and determines whether to accept its current input; the forget gate, which allows the LSTM to discard previous memory; and the output gate, which determines what to be transferred and displayed (Ren et al., 2021). These features provide LSTM algorithms with the ability to selectively process current inputs, forget previous states, and decide what information to output, making them effective and efficient in capturing sequences, which is a crucial process in predicting weather.

In this research, an LSTM-based weather forecasting model created by user priyanshu2015 on GitHub is employed. This model forecasts the temperature using 8 types of weather data from the previous three hours. The researchers have run the model using Google Colab and the parameters were identified and tested.

A. *Purpose*

In this research, the main objective is to investigate the application of the LSTM model in forecasting future weather based on the available data. The research aims to investigate the factors contributing to the accuracy of this model in predicting the weather. Historical weather information, like temperature, humidity, wind speed, wind bearings, visibility, and pressure are used to predict what the future weather might be like. This study aims to analyses how the LSTM model was trained and tested using this information to forecast weather.

B. *Parameters*

TABLE I. PARAMETERS

Parameter	Value
Batch size	128
Number of epochs	100
Loss function	MSE
Optimizer	Adam

The parameters used in training this model include the batch size, number of epochs, loss function and optimizer. The initial values of the parameters used in the original source code are shown in TABLE I. In this research, the values will be modified by the researchers to assess how it impacts the model's performance.

VI. Results and Discussion

In this section, the LSTM-based weather forecasting model will be implemented and trained with modified parameters (batch size, number of epochs, optimizer). The training loss, validation loss, time taken to conduct the training and the RMSE score are collected and analysed to compare the parameters and determine the most suitable parameters for an efficient and accurate weather forecasting model.

A. Discussion on Implementation

To set up the model, the primary dataset (weatherHistory.csv) revolving around previous weather conditions, encompassing factors like temperature, humidity, and wind force, was loaded into Google Colab files. This information is used to predict the future weather. In order to make LSTM networks work well in predicting the weather, some important parameters require careful tuning to train the model. This paper focuses on finding suitable batch size, number of epochs, and different optimizers to improve weather predictions while maintaining efficiency by consuming less computer power.

Firstly, batch size is modified to investigate its impact on the model's accuracy and training time. This research paper examines how using different batch sizes (64, 256 and 512) instead of the usual 128, as suggested by GitHub user priyanshu2015, will affect how well the LSTM model works. Using smaller batch size makes the optimization process more exact, but it takes longer to train. While increasing the amount of batch sizes for training can result in a faster process, the accuracy of the outcomes may be compromised (Pramoditha, 2023). Therefore, it is important to determine the optimal batch size that ensures a balance between training time and accuracy.

Secondly, the number of epochs determines the number of times the entire training dataset is processed by the learning algorithm during training (Vinayedula, 2023). This research experiments on different number of epochs, starting with 10, 50, 150, and 100, which is originally implemented in source code. The aim is to assess the influence of the number of iterations the model undergoes on its capacity to comprehend temporal patterns. The right number of epochs help make predictions that are accurate without making the model too simple or too complicated (DeepAI, 2020). Thus, this paper aims to find the ideal number of epochs to train the LSTM model.

Optimization algorithms help to improve how well neural networks work by changing things like weights and learning speed to reduce losses. Using optimization algorithms is very important in making the model work better (Doshi, 2021). This research commences with the "Adam" (Adaptive Moment Estimation) optimizer, which was set by the source code author, priyanshu2015. Adam optimizer is a method often used in deep learning to improve how the model learns. This study aims to understand how different optimization algorithms, such as Stochastic Gradient Descent (SGD) and Root Mean Squared Propagation (RMSprop), impact the LSTM model's learning ability from past data and accuracy in predicting the future.

The performance of this weather forecasts model was assessed using a measure called Root Mean Squared Error (RMSE). The RMSE formula calculates the average difference between the actual and predicted values of the weather and then finds the square root of that average. The accuracy of the model's predictions is computed by calculating the average size of error between the predicted values and actual values. The RMSE calculation is crucial in this research to demonstrate the performance of the LSTM model predicting weather (C3.ai, 2021).

(1) Formula for MSE = (1/n) * Σ (predicted_value - actual_value) ^2

where:

n is the total number of data points.

Σ represents the sum over all data points.

(2) Formula for RMSE = sqrt (MSE)

Fig. 1. Formula for MSE.

Figure 1 shows the formula used to calculate the RMSE value. First, the MSE ***(1)***, showing how different the predicted and actual values are, on average, is computed. After that, the square root of MSE will be calculated to get the RMSE ***(2)***. This step is done to ensure the RMSE value has the same units as the original data for clarity. RMSE is a measure used to see how well a model is doing. The smaller RMSE values are, the better the model's performance (C3.ai, 2021).

In simple terms, this research aims to assess the LSTM model's accuracy in predicting weather by experimenting with different parameters and finally, choosing the ideal values or methods best suited for the model.

B. Results

TABLE II. EPOCHS

Epochs	Average Training Loss	Average Validation Loss	RMSE	Time Taken (s)
10	0.0105	0.0049	2.827	43
50	0.0036	0.0020	1.798	84
100	0.0024	0.0014	1.648	259
150	0.0022	0.0013	1.590	432

To assess the impact of different epoch values on model accuracy, 10, 50, 150, and the original number of epochs, 100, were tested using a batch size of 128 and Adam optimization algorithm. The result, as depicted in TABLE II, has shown a noticeable trend where, as the number of epochs increases, the average training loss, validation loss, and RMSE value of the model decrease. This suggests that the model's accuracy improves with a higher number of epochs. This is because, with each epoch, the model updates its weight and bias based on the training data, leading to a reduction in average training and validation loss and an improvement in prediction accuracy (Vinayedula, 2023). Among the experimented values, training the model with 150 epochs resulted in the lowest RMSE score. However, it is also observed that the time taken to train the model at 150 epochs is significantly higher than the others. Additionally, a high number of epochs can also lead to overfitting, causing the model to fail in generalising new data. Therefore, it is important to select a suitable epoch value, which is 100 in this model, to achieve a balance between accuracy and training time and avoid overfitting.

TABLE III. BATCH SIZE

Batch Size	Average Training Loss	Average Validation Loss	RMSE	Time Taken (s)
64	0.0022	0.0021	2.096	444

128	0.0010	0.0007	1.560	251
256	0.0031	0.0015	1.748	157
512	0.0036	0.0021	1.667	144

In this experiment, the impact of different batch sizes (64, 256, 512 and the original batch size 128) on model performance was evaluated for 100 epochs using the Adam optimizer. Observing the results in Table III, smaller batch sizes (64 and 128) showed lower training loss on the training set, possibly due to more frequent parameter updates, whereas larger batch sizes (256 and 512) exhibited poorer performance on training loss, indicating that the model struggled to adapt to the training data, leading to training instability. On the validation set, smaller batch sizes demonstrated lower validation loss, suggesting better generalization ability of the model, while larger batch sizes had a decreased generalization performance, possibly due to overfitting on the training data. A smaller batch size (128) has also exhibited relatively lower RMSE, indicating more accurate predictions on the test data as compared to larger batch sizes. Additionally, larger batch sizes allow for more parallel computation, resulting in shorter training times for the same number of epochs (Sabrepc, 2023). In summary, the choice of different batch sizes has a significant impact on model performance and training efficiency. Through a comprehensive analysis of RMSE, training loss, and validation loss, Batch Size 128 consistently performed relatively well, demonstrating lower RMSE and achieving a balance between training loss and validation loss, making it a potentially good compromise in this model.

TABLE IV. OPTIMIZER

Optimizer	Average Training Loss	Average Validation Loss	RMSE	Time Taken (s)
Adam	0.0036	0.0024	1.993	120
SGD	0.0134	0.0113	5.061	136
RMSprop	0.0042	0.0023	2.070	145

To investigate the performance of the model when using different optimizers, two additional optimizers Stochastic Gradient Descent (SGD) and Root Mean Squared Propagation (RMSprop) are chosen on top of the original Adam optimizer while maintaining the original number of epochs (100) and batch size (128). The results are listed in Table IV. In terms of the average training loss, it is found out that SGD obtained the least satisfactory result out of the three, while Adam had the lowest loss with RMSprop following close behind. As for average validation loss, RMSprop had a slight advantage over Adam, while SGD still fell behind. According to the RMSE values obtained, it is safe to assume that SGD is not a suitable optimizer for this model with a high value of 5.061. When comparing the time taken to train using Adam and RMSprop optimizer, Adam is recorded to be 25 seconds quicker. According to an online article written by Agarwal (2023), it is stated that Adam optimizer has a much faster converge rate than other optimizers, which allows to reach its performance much sooner for quicker training times. Hence, it can be concluded that the original optimizer, Adam, is the superior choice for training and testing for this model.

VII. CONCLUSIONS

In conclusion, this paper has investigated some essential parameters to find out the best choices that could help the LSTM model reach its best performance when predicting the weather. Based on the experiments conducted, it is found that the original parameters selected were already the most suitable for its model, which were 100 epochs, batch size of 128 and the Adam optimizer. There are some parameters that performed better than others in certain aspects, but other conditions should also be considered. As an example, although the epoch values of 150 had a better performance than 100 epoch values for the model, it took a longer time to complete. It should be noted that time taken is an important factor for real-time forecasting. Therefore, we may conclude that the current LSTM model has excellent performance, but further research should be made to help enhance its accuracy for weather prediction.

ACKNOWLEDGMENT

We would like to extend our sincere appreciation to priyanshu2015 for sharing their weather forecasting code on GitHub, which has greatly facilitated our research. We would also like to extend our gratitude to the authors of the journal papers, whether referenced or not, for their comprehensive work that significantly enhanced our knowledge in this field. Lastly, the utmost thanks to every team member for their collaborative efforts in writing this research paper, which has contributed to the successful completion of this project.

REFERENCES

Agarwal, R. (2023, September 13). Complete guide to the ADAM Optimization Algorithm. Built In. https://builtin.com/machine-learning/adam-optimization

Balehegn, M., Balehey, S., Fu, C., & Liang, W. (2019). Indigenous weather and climate forecasting knowledge among Afar pastoralists of north eastern Ethiopia: Role in adaptation to weather and climate variability. Pastoralism: Research, Policy and Practice, 9(8). https://doi.org/10.1186/s13570-019-0143-y

Bhawsar, M., Tewari, V., & Khare, P. (2021). A Survey of Weather Forecasting based on Machine Learning and Deep Learning Techniques. International Journal of Emerging Trends in Engineering Research, 9(7), 988–993. https://doi.org/10.30534/ijeter/2021/24972021

Bochenek, B., & Ustrnul, Z. (2022). Machine Learning in Weather Prediction and Climate Analyses—Applications and Perspectives. Atmosphere, 13(2), 180. https://doi.org/10.3390/atmos13020180

C3.ai. (2021, September 28). Root Mean square Error (RMSE). C3 AI. https://c3.ai/glossary/data-science/root-mean-square-error-rmse/#:~:text=Root%20mean%20square%20error%20or,true%20values%20using%20Euclidean%20distance.

Cebeci, Y. E. (2019). A recurrent neural network model for weather forecasting. 2019 4th International Conference on Computer Science and Engineering (UBMK). https://doi.org/10.1109/ubmk.2019.8907196

DeepAI. (2020, June 25). Epoch. DeepAI. https://deepai.org/machine-learning-glossary-and-terms/epoch#:~:text=In%20the%20context%20of%20machine,dataset%20through%20the%20learning%20algorithm.

Dharmasena, P. B. (2021). Indigenous Weather Forecasting Methods in Sri Lanka. Sri Lanka Journal of Agrarian Studies. Special Issue: Agricultural Heritage of Sri Lanka 2021.

Doshi, S. (2021, December 7). Various optimization algorithms for training neural network. Medium. https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6#:~:text=Optimizers%20are%20algorithms%20or%20methods,help%20to%20get%20results%20faster

Fathi, M., Haghi Kashani, M., Jameii, S. M., & Mahdipour, E. (2021). Big Data Analytics in weather forecasting: A systematic review. Archives of Computational Methods in Engineering.

Heikkiläarchive, M. (2023, July 11). Weather forecasting is having an AI moment. Retrieved from MIT Technology Review.

Hewage, P., Behera, A., Trovati, M., Pereira, E., Ghahremani, M., Palmieri, F., & Liu, Y. (2020). Temporal Convolutional Neural (TCN) network for an effective weather forecasting using time-series data from the local weather station. Soft Computing, 24(21), 16453–16482. https://doi.org/10.1007/s00500-020-04954-0

Hewage, P., Trovati, M., Pereira, E., & Behera, A. (2020). Deep learning-based effective fine-grained weather forecasting model. Pattern Analysis and Applications, 24(1), 343–366. https://doi.org/10.1007/s10044-020-00898-1

Hickey, H. (2020, December 15). AI model shows promise to generate faster, more accurate weather forecasts. Retrieved from Phys.org: https://phys.org/news/2020-12-ai-faster-accurate-weather.html

Ng, Y. R., Abdul Salam, Z. A., How, Y. H., Mohammed, O., & Cheong, Y. K. (2023). Convolutional Neural Network for Fruit Image Classification. Journal of Applied Technology and Innovation, 7(1). https://dif7uuh3zqcps.cloudfront.net/wp-content/uploads/sites/11/2022/12/14090157/Volume7_Issue1_Paper5_2023.pdf

Pu, Z., & Kalnay, E. (2018). Numerical Weather Prediction Basics: Models, numerical methods, and data assimilationHandbook of Hydrometeorological Ensemble Forecasting. Handbook of Hydrometeorological Ensemble Forecasting.

Radeny, M., Desalegn, A., Mubiru, D., Kyazze, F., Mahoo, H., Recha, J., Kimeli, P., & Solomon, D. (2019). Indigenous knowledge for seasonal weather and climate forecasting across East Africa. Climatic Change, 156, 509-526. https://doi.org/10.1007/s10584-019-02476-9

Reilly, J. (2024, January 4). Using machine learning for accurate weather forecasts in 2023. Akkio. https://www.akkio.com/post/weather-prediction-using-machine-learning#:~:text=While%20traditional%20weather%20forecasting%20methods,data%20to%20predict%20future%20conditions.

Ren, X., Li, X., Ren, K., Song, J., Xu, Z., Deng, K., & Wang, X. (2021). Deep learning-based Weather prediction: A survey. Big Data Research, 23, 100178. https://doi.org/10.1016/j.bdr.2020.100178

Sabrepc. (2023, February 10). Epochs, Batch Size, Iterations - How are They Important to Training AI and Deep Learning Models. Retrieved from sabrepc: https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations.

Schultz, M. G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L. H., Mozaffari, A., & Stadtler, S. (2021). Can deep learning beat numerical weather prediction? Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379(2194), 20200097. https://doi.org/10.1098/rsta.2020.0097

Tahiluddin, A. B., Ullang Jr., A. H., Jali, A. M., Ajik, J. O., Ebbah, J. H., & Jamil, W. M. (2023). Traditional Weather Forecasting: Practices of Fishers, Sailors and Old Folks in Tawi-Tawi, Philippines. Acta Natura et Scientia, 4(1), 10-16.

Tiu, K. S., Reka Ramachandran, C., & Meskaran, F. (2021). Dengue disease prediction using machine learning algorithms: A review. jati.apu.edu.my. https://dif7uuh3zqcps.cloudfront.net/wp-content/uploads/sites/11/2021/10/07183920/Volume5_Issue4_Paper6_2021.pdf

Vinayedula. (2023, July 31). How does epoch affect accuracy in deep learning model?. GeeksforGeeks. https://www.geeksforgeeks.org/how-does-epoch-affect-accuracy-in-deep-learning-model/

Wang, B., Lu, J., Yan, Z., Luo, H., Li, T., Zheng, Y., & Zhang, G. (2019). Deep Uncertainty Quantification. KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. https://doi.org/10.1145/3292500.3330704

Xiao, H., Zhang, F., Shen, Z., Wu, K., & Zhang, J. (2021). Classification of weather phenomenon from images by using deep convolutional neural network. Earth and Space Science, 8(5). https://doi.org/10.1029/2020ea001604

Zhou, K., Zheng, Y., Li, B., Dong, W., & Zhang, X. (2019). Forecasting different types of convective weather: A deep learning approach. Journal of Meteorological Research, 33(5), 797–809. https://doi.org/10.1007/s13351-019-8162-6