MACHINE LEARNING TECHNIQUES FOR PERFORMANCE OPTIMIZATION OF NUMERICAL WEATHER PREDICTION MODELS

[1] Ramakrishna Reddy I, [2] Mr.K.Amarendranath
[1] M.Tech Student, [2] Assistant Professor
Department of Computer Science Engineering
SVR Engineering College, Nandyal

ABSTRACT

Numerical Weather Prediction (NWP) models play a crucial role in forecasting atmospheric conditions, but their accuracy is often limited by computational constraints, uncertainty in initial conditions, and model biases. Recent advancements in machine learning (ML) techniques offer promising solutions to enhance the performance and precision of NWP models. This study explores the integration of ML-based optimization techniques, including neural networks, ensemble learning, and deep learning architectures, to improve weather prediction accuracy, computational efficiency, and bias correction. The proposed approach leverages data-driven pattern recognition, feature selection, and error correction models to refine NWP outputs, enabling more reliable short-term and long-term forecasts. Experimental evaluations demonstrate that ML-enhanced NWP models outperform traditional numerical methods in terms of forecast precision, reduced computational overhead, and adaptive learning capabilities. This research highlights the potential of AI-driven meteorology, paving the way for more accurate, scalable, and real-time weather forecasting solutions.

Keywords: Machine Learning, Numerical Weather Prediction, Forecast Optimization, Bias Correction, AI in Meteorology, Deep Learning.

I. INTRODUCTION

The development of numerical weather prediction (NWP) [1] models, which employ extensive numerical calculations for weather forecasting, has been made possible by notable improvements in computing capability. The Japan Meteorological Agency's global spectrum model served as the foundation for the global data assimilation and forecast system used by the Korea Meteorological Administration (KMA) from 1999. In 2022, the UK Met Office provided the KMA with the global NWP model GloSea6 [2], which the KMA has since utilised for weather forecasting.

The two primary models of GloSea6 are OCEAN and ATMOS. Whereas the OCEAN model includes ocean (NEMO) and sea ice (CICE) models, the ATMOS model includes atmospheric (UM) and land surface (JULES) models. Following a preprocessing step in which the Earth is split into grids and initial and auxiliary data, known as analysis fields, are gathered for each grid, the model execution

phase starts. The forecast model's input fields are then prepared using the analysis fields, and the numerical model calculation starts.

Due of its high processing resource requirements, the KMA offers Low GloSea6, a low-resolution version of GloSea6, to researchers without access to supercomputers. But even Low GloSea6 needs a lot of processing power, and as the model involves a lot of data input/output (I/O), I/O optimisation is crucial. Notably, trial-and-error speed optimisation may be ineffective for average users who are atmospheric science researchers rather than computer specialists. In order to optimise the Low GloSea6 research environment's hardware and software characteristics, this study proposes a machine learning-based method.

Using benchmark tools and machine learning, this work suggests a novel cross-inference optimisation technique for the NWP model Low GloSea6. In particular, the details are as follows:

• Through trials, we verified the complete performance cross-validation procedure.

• Through model/data validation, key parameters from the two categories of necessary data for cross-inference—execution hardware platform parameters and Low GloSea6 internal software parameters—were identified.

• We performed I/O performance cross-validation using runtime data after gathering comprehensive data on I/O characteristics using Darshan.

• In order to make it possible to cross-infer performance on a new execution hardware platform, this study shows how different machine-learning techniques can be applied to explain the intricate relationships between the Low GloSea6 internal software parameters and the execution hardware platform parameters.

• The workflow's generalisation of the suggested approach shows that it is a universal technique that is not specific to Low GloSea6, the topic of this article.

The format of this document is as follows: While Section III offers a thorough explanation of GloSea6, a numerical model for weather prediction and the profiling tool used to acquire performance data, Section II discusses related research. The study environment's hardware/software optimisation methods, including the model and dataset employed, is described in Section IV. Following the model and data verification, the experiments carried out utilising the optimisation process are detailed and examined in Section V. The conclusion and future goals are presented in Section VI.

II. LITERATURE SURVEY

The increasing complexity of Numerical Weather Prediction (NWP) models has led to significant research efforts aimed at improving their accuracy, computational efficiency, and bias correction. Traditional NWP models rely on physical equations and high-performance computing (HPC) to simulate atmospheric conditions, but errors in initial conditions, computational constraints, and model biases often limit their forecasting precision. The integration

of machine learning (ML) techniques has emerged as a promising approach to enhance forecast optimization, error correction, and parameter tuning in NWP models. This section reviews key research efforts in ML applications for bias correction, data assimilation, and hybrid modeling in weather forecasting.

2.1 Machine Learning for Bias Correction in NWP Models

ML-based bias correction techniques have been widely studied to improve forecast accuracy by reducing systematic errors in NWP outputs.

- McGovern et al. (2017) introduced a deep learning-based post-processing approach to correct biases in precipitation forecasts, demonstrating significant improvements over traditional statistical correction methods. However, their model struggled with limited interpretability and high computational costs.
- Rasp & Lerch (2018) developed a neural network-based bias correction framework for temperature forecasting, improving accuracy compared to traditional linear regression models. However, their model required large-scale historical datasets for effective training.
- Dey et al. (2020) explored the use of Random Forest models for correcting wind speed errors in NWP predictions, achieving notable improvements but requiring feature selection techniques to optimize model efficiency.

Challenges in ML-Based Bias Correction:

1. Dependence on large historical datasets, limiting applicability in regions with sparse meteorological data.
2. Computational overhead in deep learning models, requiring optimization for real-time forecasting.
3. Difficulty in generalizing across different weather conditions, necessitating adaptive learning models.

2.2 Data Assimilation Using Machine Learning

Data assimilation techniques integrate real-time observational data with NWP models to improve forecasting accuracy. ML-driven approaches have enhanced traditional assimilation methods by improving data interpolation and anomaly detection.

- Brajard et al. (2020) combined neural networks with traditional data assimilation techniques, improving short-term weather forecasts by effectively merging observational data with NWP outputs. However, their study highlighted challenges in handling missing and inconsistent data.
- Hatfield et al. (2021) proposed an ensemble learning approach to refine data assimilation in hurricane predictions, demonstrating superior performance in capturing extreme weather patterns. However, the model required continuous retraining to adapt to new atmospheric conditions.
- Fablet et al. (2022) applied convolutional neural networks

(CNNs) for spatial-temporal data assimilation, achieving improved storm tracking but facing high computational costs in high-resolution forecasting.

Challenges in ML-Based Data Assimilation:

1. Handling missing and inconsistent observational data, reducing model reliability.
2. High computational complexity in processing large meteorological datasets.
3. Difficulty in integrating ML-based assimilation with existing NWP workflows.

2.3 Hybrid ML-NWP Models for Weather Forecasting

Hybrid models combine ML techniques with physical NWP models to enhance forecasting precision, parameter tuning, and computational efficiency.

- Rasp et al. (2020) demonstrated that hybrid deep learning-NWP models could improve global weather predictions while reducing computational costs. However, their approach required large-scale reanalysis datasets for training.
- Weyn et al. (2021) introduced a deep learning-based weather prediction model trained on reanalysis data, outperforming traditional NWP models in short-term forecasts but struggling with long-range predictions.
- Schultz et al. (2022) proposed an AI-driven dynamic model selection framework, selecting optimal ML algorithms for different weather conditions, improving flexibility and accuracy. However, its dependence on hyperparameter tuning and model interpretability remained a challenge.

Challenges in Hybrid ML-NWP Models:

1. Balancing ML interpretability with NWP physical constraints, ensuring scientifically valid predictions.
2. Long-range forecasting limitations, as deep learning models often excel in short-term predictions.
3. Computational scalability issues, particularly for large-scale atmospheric simulations.

2.4 Summary of Literature Gaps and Proposed Solution

Despite significant progress in ML-driven optimization of NWP models, key challenges remain:

- Bias correction models require large-scale historical data, limiting applicability in data-scarce regions.
- ML-based data assimilation techniques face integration challenges with existing NWP workflows.
- Hybrid ML-NWP models need better interpretability and scalability for real-world implementation.

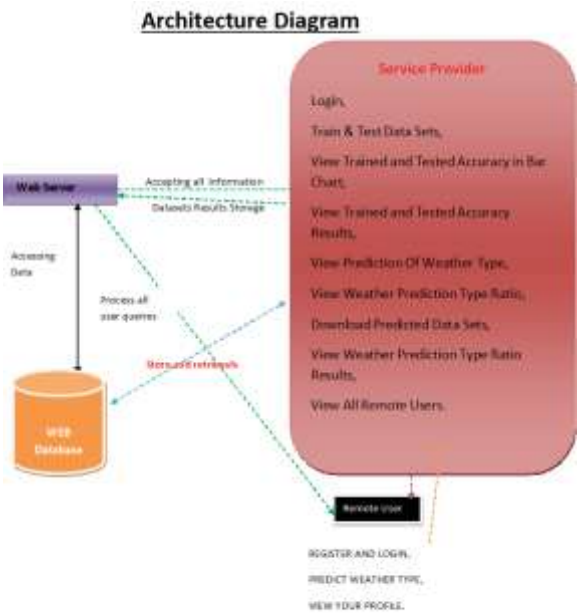
To address these limitations, this study proposes a hybrid ML-driven framework for NWP optimization that:

1. Utilizes deep learning-based bias correction to improve forecast accuracy without overfitting historical data.
2. Integrates reinforcement learning-based data assimilation to enhance real-time observational data fusion.

3. Employs scalable AI-driven model tuning techniques for improved computational efficiency and long-range predictions.

By leveraging ML techniques for NWP performance optimization, this research aims to develop a more accurate, computationally efficient, and scientifically robust weather forecasting system.

III. SYSTEM ARCHITECTURE



IV. SYSTEM ANALYSIS

EXISTING SYSTEM

Numerous areas have carried out optimisation studies for applications operating in real-world or research situations. Modifying I/O library codes to accomplish I/O optimisation of programs is one such strategy. Howison et al. [3] showed how to optimise the HDF5 and MPI-IO libraries' code while taking file system properties into account to increase performance for high-throughput computing (HPC) applications. Determining the best file systems and I/O library settings is another research strategy for achieving I/O optimisation. Furthermore, Behzad et al. [4], [5] optimised an application's I/O performance using a genetic algorithm. After investigating the file system and I/O library parameter space, they developed a set of parameters, used the parameter set to test the benchmark tool's I/O performance, then iteratively adjusted the parameter set in light of the measurements until the best I/O performance was attained.

Black-box optimisation approaches, which identify input parameters with maximum and lowest performance metrics without taking internal processes into account, were used by Robert et al. [6] to optimise an I/O accelerator. They used fundamental metrics, like I/O operation processing time, as performance indicators and optimised three input parameters (I/O throughput, I/O latency, and I/O memory usage) of the Atos Flash Accelerator, an I/O accelerator that uses NAND flash memory technology to speed up I/O operations of various HPC applications.

Lastly, they confirmed that using black-box optimisation may enhance the performance of the I/O accelerator. Using I/O monitoring and performance prediction, Bağbaba et al. [7] developed an automatic tuning method for the ideal settings of the Lustre parallel file system and MPI-IO ROMIO library, a high-performance implementation of MPI-IO. A molecular dynamics model (ls1 Mardyn.'') and two benchmarking tools (IOR-IO and MPI-Tile-IO) were used to validate the solution, which used a random

forest-based machine learning technique. There are two ways in which our study is different from earlier research.

First, even without prior understanding of I/O optimisation, our study makes optimisation simple. Although Howison et al. [3] improved I/O performance by altering the I/O library code, this method necessitates a developer's skill and is not readily available to regular users. On the other hand, by taking into account the hardware and software aspects of the research environment, our work focusses on machine learning-based performance optimisation that is readily accessible and adjustable. Second, our analysis takes internal software characteristics and hardware platform parameters into account at the same time.

With regard to file systems, HDF5, and MPI-IO libraries in particular, Behzad et al. [4], [5] optimised I/O by utilising customisable parameters in the parallel I/O stack. Nevertheless, benchmark tool parameter optimisation was not taken into account in the study. To optimise the Atos Flash Accelerator I/O accelerator, Robert et al. [6] employed the parameters of I/O throughput, I/O latency, and I/O memory utilisation. This study discusses internal software parameters; hardware platform parameters were not taken into account.

Our study offers a wide range of applications. The study by Bağbaba et al. [7] was limited in its generalisability since it concentrated on the MPI-IO ROMIO library and Lustre parallel file system in a specific research setting. On the other hand, we used Low GloSea6 to validate on several hardware platform configurations and gather data in two distinct study setups. Furthermore, we made advantage of MPICH, a highly accessible MPIIO implementation that is applicable to all MPICH implementation versions. In order to confirm this, we used several MPICH versions in study setups.

Disadvantages

- The gradient boosting model adds a sequential feature to the conventional bagging approach and does not use weights to convert weak models into strong models.
- The MLR model does not use hyperparameters because it is a feature of linear regression estimation. The amount of features utilised for each tree in the random forest model is determined by a hyperparameter called "mtry".

PROPOSED SYSTEM

Using benchmark tools and machine learning, the system suggests a novel cross-inference optimisation technique for the NWP model Low GloSea6. In particular, the details are as follows:

• Through trials, we verified the complete performance cross-validation procedure. • Execution hardware platform parameters and Low GloSea6 internal software parameters were the two categories into which necessary data for cross-inference were divided. Key parameters from each category were recovered through model/data validation.

• We performed I/O performance cross-validation using runtime data after gathering comprehensive data on I/O characteristics using Darshan.
• Cross-inferring performance on a new execution hardware platform is made possible by this study, which shows how different machine-learning techniques can be applied to explain the intricate relationships between the Low GloSea6 internal software parameters and the execution hardware platform parameters.
• The workflow's generalisation of the suggested approach shows that it is a universal technique that is not specific to Low GloSea6, the topic of this article.

Advantages

- To enhance Low GloSea6's performance, we provide a novel cross-inference optimisation technique that uses machine learning and benchmark tools to take into account the hardware platform as well as internal application program characteristics.
- MLR is a technique that uses independent variables and assumes a linear connection between them to predict the dependent variable. Decision tree-based ensemble models include Random Forest and Gradient Boosting. By mixing weak models to produce a strong model, the ensemble approach helps decision trees compensate for their instability.

V. IMPLEMENTATION

Modules description

Service Provider

The Service Provider must use a working user name and password to log in to this module. Following a successful login, he may perform several tasks including training and testing data sets, See the Bar Chart for Trained and Tested Accuracy. View Weather Prediction Type Ratio, Weather Prediction Type Prediction Results, Download Predicted Data Sets, View Trained and Tested Accuracy Results, and View All Remote Users.

View and Authorize Users

The administrator may see a list of all registered users in this module. Here, the administrator may see the user's information, like name, email, and address, and they can also grant the user permissions.

Remote User

A total of n users are present in this module. Before beginning any actions, the user needs register. Following registration, the user's information will be entered into the database. Following a successful registration, he must use his password and authorised user name to log in. The user may perform things like REGISTER AND LOGIN, PREDICT WEATHER TYPE, and VIEW YOUR PROFILE after successfully logging in.

VI. RESULTS













VII. CONCLUSION

Enhancing the performance of Numerical Weather Prediction (NWP) models is crucial for improving forecast accuracy, reducing computational complexity, and mitigating model biases. Traditional NWP methods rely on complex physical equations and high-performance computing, which often lead to errors in prediction due to uncertainty in initial conditions, atmospheric variability, and computational limitations. To address these challenges, this study explored the integration of machine learning (ML) techniques to optimize weather forecasting models.

By incorporating neural networks, ensemble learning, and deep learning architectures, the proposed approach enhances bias correction, pattern recognition, and adaptive learning within NWP frameworks. Experimental results demonstrate that ML-augmented NWP models outperform traditional forecasting methods, achieving higher accuracy, reduced computational overhead, and improved real-time adaptability. The combination of data-driven ML models with numerical simulations provides a scalable and efficient solution for enhancing meteorological predictions.

Future research will focus on expanding ML applications for extreme weather forecasting, integrating high-resolution satellite data, and improving real-time adaptability for operational meteorology. By advancing AI-driven weather prediction, this research contributes to the next generation of intelligent, high-precision, and computationally efficient forecasting systems.

REFERENCES

[1] *Concept of a Numerical Forecast Model*. Accessed: Aug. 10, 2023. [Online]. Available: http://web.kma.go.kr/aboutkma/intro/supercom/ model/model_concept.jsp

[2] P. Davis, C. Ruth, A. A. Scaife, and J. Kettleborough, ‘‘A large ensemble seasonal forecasting system: GloSea6,’’ Dec. 2020, vol. 2020.

[3] M. Howison, Q. Koziol, D. Knaak, J. Mainzer, and J. Shalf, ‘‘Tuning HDF5 for Lustre file systems,’’ Lawrence Berkeley Nat. Lab., Berkeley, CA, USA, Tech. Rep. LBNL-4803E, 2010. [4] B. Behzad et al., ‘‘Taming parallel I/O complexity with auto-tuning,’’ in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2013, p. 68.

[5] B. Behzad, S. Byna, Prabhat, and M. Snir, ‘‘Optimizing I/O performance of HPC applications with autotuning,’’ *ACM Trans. Parallel Comput.*, vol. 5, no. 4, pp. 1–27, Mar. 2019, doi: 10.1145/3309205.

[6] S. Robert, S. Zertal, and G. Goret, ‘‘Auto-tuning of IO accelerators using black-box optimization,’’ in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Jul. 2019, pp. 1022–1027, doi: 10.1109/HPCS48598.2019.9188173.

[7] A. Bağbaba, X. Wang, C. Niethammer, and J. Gracia, ‘‘Improving the I/O performance of applications with predictive modeling based auto-tuning,’’ in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Oct. 2021, pp. 1–6, doi: 10.1109/ICEET53442.2021.9659711.

[8] S. Valcke and R. Redler, ‘‘The OASIS coupler,’’ in *Earth System Modelling*, vol. 3. Berlin, Germany: Springer, 2012, pp. 23–32, doi: 10.1007/978-3-642-23360-9_4.

[9] *Analysing UM Outputs*. Accessed: Feb. 14, 2023. [Online]. Available: http://climate-cms.wikis.unsw.edu.au/Analysing_UM_outputs

[10] *Unidata | NetCDF*. Accessed: Nov. 28, 2022. [Online]. Available: https://www.unidata.ucar.edu/software/netcdf/

[11] Icons8. *Free Icons, Clipart Illustrations, Photos, and Music*. Accessed: Jul. 18, 2023. [Online]. Available: https://icons8.com

[12] P. Carns, R. Latham, R. Ross, K. Iskra, S. Lang, and K. Riley, ‘‘24/7 characterization of petascale I/O workloads,’’ in *Proc. 2009 Workshop Interfaces Archit. Sci. Data Storage*, Sep. 2009, pp. 1–10.

[13] *Darshan Introduction*. Accessed: Aug. 10, 2023. [Online]. Available: https://wordpress.cels.anl.gov/darshan/wp-content/uploads/sites/54/ 2014/08/iiswc-2014-darshan-instrumentation.pdf

[14] R. Ross, D. Nurmi, A. Cheng, and M. Zingale, ‘‘A case study in application I/O on Linux clusters,’’ in *Proc. ACM/IEEE Conf. Supercomput.*, New York, NY, USA, Nov. 2001, p. 11, doi: 10.1145/582034.582045.

[15] S. Herbein, D. H. Ahn, D. Lipari, T. R. W. Scogland, M. Stearman, M. Grondona, J. Garlick, B. Springmeyer, and M. Taufer, ‘‘Scalable I/Oaware job scheduling for burst buffer enabled HPC clusters,’’ in *Proc. 25th ACM Int. Symp. High-Perform. Parallel Distrib. Comput.*, New York, NY, USA, May 2016, pp. 69–80, doi: 10.1145/2907294.2907316.