# MACHINE LEARNING-DRIVEN OPTIMIZATION OF NUMERICAL WEATHER PREDICTION MODELS

[1]Pinjari Akbar Basha, MCA Student, Department of MCA

[2] CH Sri Lakshmi Prasanna, M.Tech, (Ph.D), Assistant Professor, Department of MCA

[12]Dr KV Subba Reddy Institute of Technology, Dupadu, Kurnool

https://doi.org/10.51470/ijcnwc.2025.v15.i02.pp889-899

**ABSTRACT**

In order to anticipate future weather, weather forecasting mostly employs numerical weather prediction models that incorporate weather observation data, such as temperature and humidity. For weather forecasting, the Korea Meteorological Administration (KMA) has embraced the UK's GloSea6 numerical weather prediction model. Supercomputers are necessary to run these models for research reasons in addition to using them for real-time weather predictions. However, several researchers have encountered challenges while attempting to run the models because to the restricted supercomputer resources. Low GloSea6, a low-resolution model created by the KMA to solve this problem, can be operated on small and medium-sized servers at research institutes; nonetheless, it still consumes a lot of computer resources, particularly in the I/O load. Model I/O optimisation is crucial because I/O load may degrade performance for models with heavy data I/O, yet user trial-and-error optimisation is ineffective. In order to optimise the Low GloSea6 research environment's hardware and software characteristics, this study offers a machine learning-based method. There were two phases in the suggested approach. In order to determine hardware platform parameters and Low GloSea6 internal parameters under different settings, performance data was first gathered using profiling tools. Second, to identify the ideal hardware platform characteristics and Low GloSea6 internal parameters for novel research contexts, a machine learning model was developed using the data gathered. In contrast to the actual parameter combinations, the machine-learning model demonstrated a high degree of accuracy in accurately predicting the ideal parameter combinations in various study situations. With an error rate of only 16% when compared to the actual execution time, the projected model execution time based on the parameter combination in particular produced a noteworthy result. All things considered, this optimisation technique might enhance the functionality of more high-performance computing research applications.

## I. INTRODUCTION

The development of numerical weather prediction (NWP) [1] models, which use extensive numerical calculations for weather forecasting, has been made possible by notable improvements in computer 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 computation 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 whole 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 used, 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

GloSea6 is a large ensemble seasonal forecasting system.

J. Kettleborough, A. A. Scaife, C. Ruth, and P. Davis,

The Met Office's monthly to seasonal ensemble prediction method is called Global Seasonal Forecasting method (GloSea). A major scientific improvement has been made to the current version, GloSea5, which has been in use since 2014. This includes a method for iceberg advection, enhancements to sea-ice physics, and a consideration of convective entrainment in the updated coupled model. Furthermore, we include a more practical approach to land-surface initialisation, starting soil moisture using a forced land model rather than a climatology. Additionally, we use CMIP6 forcing data to substitute three-dimensional, time-varying fields for a) a constant solar forcing and b) zonal mean climatology of ozone concentrations with fluctuating fluxes that reflect the Solar Cycle. We use re-forecasts from 1993 to 2016 to explain GloSea6's scientific performance, using 100 members each season and start dates for the winter and summer. GloSea5 is used as a baseline to compare the results. The majority of skill changes are either positive or neutral, and we also look at how the broader ensemble affects skill. While there is no discernible difference for the 2010 Russian event, GloSea6 more accurately captures the Z500 and near-surface temperature anomalies during the 2003 European heatwave. Furthermore, during December-January-February (DJF), we see reduced sea-surface temperature (SST) biases for both El Nino and La Nina. Additionally, the observed figure is more in line with the SST ensemble mean standard deviation for El Nino during DJF. The NAO skill and GloSea5 are comparable. However, we see that compared to GloSea5, the September sea-ice extent bias is greater. This could result from how melt-ponds and sea-ice drag are handled.

"Setting up HDF5 for Lustre files,"

Q. Koziol, D. Knaak, J. Mainzer, J. Shalf, M. Howison,

Numerous HPC applications employ HDF5, a cross-platform parallel I/O toolkit, because of its flexible hierarchical object-database representation of scientific data. We present our latest efforts to improve the Lustre parallel file system's HDF5 and MPI-IO libraries' performance. To illustrate the resilience of our optimisations across various file system configurations and to confirm our optimisation approach, we chose three distinct HPC programs to reflect the wide variety of I/O needs and evaluated their performance on three separate platforms. In some situations, we show that the combined optimisations increase HDF5 parallel I/O performance by up to 33 times, approaching the underlying file system's possible peak performance. We further show that the performance is scalable up to 40,960-way concurrency.

"Auto-tuning to control parallel I/O complexity,"

B. Behzad and associates,

We show the usefulness of our auto-tuning method for HDF5 applications' I/O performance optimisation across platforms, applications, and scale. The system finds efficient settings at every tier of the parallel I/O stack by searching a wide range of configurable parameters using a genetic algorithm. The auto-tuning mechanism uses dynamically intercepted HDF5 calls to transparently apply the parameter adjustments. We used three I/O benchmarks (VPIC, VORPAL, and GCRM) that mimic the I/O activity of their respective applications to test our auto-tuning method. We evaluated the system on a variety of HPC systems (Cray XE6, IBM BG/P, and Dell Cluster) with various weak-scaling configurations (128, 2048, and 4096 CPU cores) that produce 30 GB to 1 TB of data. The auto-tuning mechanism consistently found adjustable parameters that significantly enhanced write performance over the system's default configuration. For test settings, we routinely show I/O write speedups ranging from 2× to 100×.

"Using autotuning to optimise the I/O performance of HPC applications,"

Prabhat, M. Snir, S. Byna, and B. Behzad

Optimising theconfigurable parameters throughout the many levels of the I/O software stack is essential to enhancing parallel I/O performance. The wide parameter space and the intricate dynamics of interaction between these factors make it difficult to find the best configuration for various conditions. Prior studies have concentrated on adjusting these parameters using separate algorithms; yet, these methods have major drawbacks, including inconsistent performance outcomes and sluggish convergence rates. This study presents OPRAEL, a regression-based performance modelling and auto-tuning method for parallel

I/O jobs via ensembles. We used two I/O kernels (S3D-I/O, BT-I/O) and one well-known I/O benchmark (IOR) on the Tianhe-II supercomputer to assess the efficacy of this method. We tuned the I/O stack characteristics as efficiently as possible by using our knowledge of predictive modelling. Our testing findings provide a noteworthy 10.2X increase in write performance speedup for the optimisation challenge using a 500x500x500 input and BT-I/O. In the I/O parameter auto-optimization challenge, we also contrasted the possibility of using reinforcement learning search with that of a single search method. According to our findings, OPRAEL performs better than the conventional method, improving write speed for the 128-process IOR optimisation by up to 8.4X. "II accelerator auto-tuning with black-box optimisation,"

The performance and behaviour of High Performance Computing (HPC) programs depend on software and hardware environments, which are often quite adjustable, according to S. Robert, S. Zertal, and G. Goret. It is quite difficult to determine their ideal parametrisation. In most situations, hand-tuning, theoretical modelling, or exhaustive sampling are inappropriate due to the vastness of the parametric space and the non-linear connection between the parameters and the performance that is given. In this research, we offer an auto-tuning loop that does not assume anything about the performance function. It employs black-box optimisation to find the best parametrisation of IO accelerators for a specific HPC application in a short number of iterations. We explain the installation and trial of the chosen accelerator tuning techniques in our HPC setting utilising two Atos-developed IO accelerators after a study of the literature. Since our success criteria extend beyond determining the ideal parameters, we additionally establish a number of indicators to assess the calibre of our optimisation. The findings collected demonstrate that this framework effectively reduces the execution time of two programs when used in tandem with a mixed hardware-software accelerator and a pure software accelerator. In fact, we see potential time savings of 38% and 20% for each accelerator, respectively, as compared to starting the identical application with the default settings.

## III. SYSTEM ANALYSIS AND DESIGN

### 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 verify 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 previous understanding of I/O optimisation, our research makes optimisation simple. Although Howison et al. [3] improved I/O performance by altering the I/O library code, this method requires 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 factors 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 using 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 since 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 whole 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 via model/data validation.

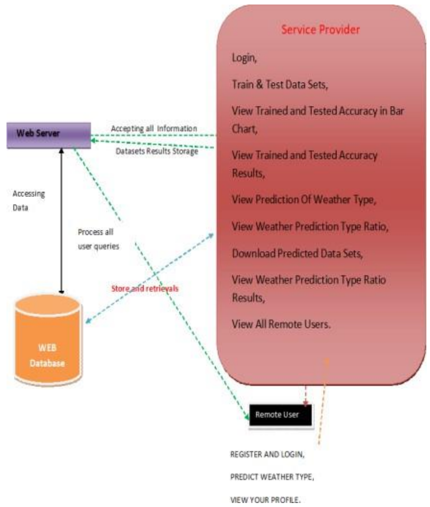
• We performed I/O performance cross-validation using runtime data after gathering comprehensive data on I/O characteristics using Darshan.

- 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.
- Assuming a linear connection between the independent variables, MLR is a technique for forecasting 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.

**SYSTEM ARCHITECTURE**




## IV. IMPLEMENTATION

**Modules**

**Service Provider**

The Service Provider must use a working user name and password to log in to this module. He may do many tasks after successfully logging in, including Train & Test 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.

**ALGORITHMS**

**Gradient boosting**

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

**K-Nearest Neighbors (KNN)**

- Simple, but a very powerful classification algorithm
- Classifies based on a similarity measure
- Non-parametric
- Lazy learning
- Does not “learn” until the test example is given
- Whenever we have a new data to classify, we find its K-nearest neighbors from the training data

**Example**

- Training dataset consists of k-closest examples in feature space
- Feature space means, space with categorization variables (non-metric variables)
- Learning based on instances, and thus also works lazily because instance close to the input vector for test or prediction may take time to occur in the training dataset

**Logistic regression Classifiers**

The relationship between a collection of independent (explanatory) factors and a categorical dependent variable is examined using logistic regression analysis. When the dependent variable simply has two values, like 0 and 1 or Yes and No, the term logistic regression is used. When the dependent variable contains three or more distinct values, such as married, single, divorced, or widowed, the technique is sometimes referred to as multinomial logistic regression. While the dependent variable's data type differs from multiple regression's, the procedure's practical application is comparable.

When it comes to categorical-response variable analysis, logistic regression and discriminant analysis are competitors. Compared to discriminant analysis, many statisticians believe that logistic regression is more flexible and appropriate for modelling the majority of scenarios. This is due to the fact that, unlike discriminant analysis, logistic regression does not presume that the independent variables are regularly distributed.

Both binary and multinomial logistic regression are calculated by this software for both category and numerical independent variables. Along with the regression equation, it provides information on likelihood, deviance, odds ratios, confidence limits, and quality of fit. It does a thorough residual analysis that includes diagnostic residual plots and reports. In order to find the optimal regression model with the fewest independent variables, it might conduct an independent variable subset selection search. It offers ROC curves and confidence intervals on expected values to assist in identifying the optimal classification cutoff point. By automatically identifying rows that are not utilised throughout the study, it enables you to confirm your findings.

**Naïve Bayes**

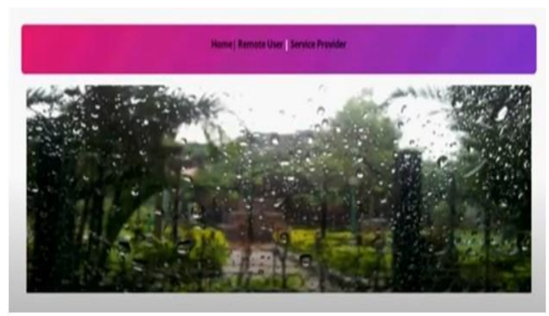
The supervised learning technique known as the "naive bayes approach" is predicated on the straightforward premise that the existence or lack of a certain class characteristic has no bearing on the existence or nonexistence of any other feature.

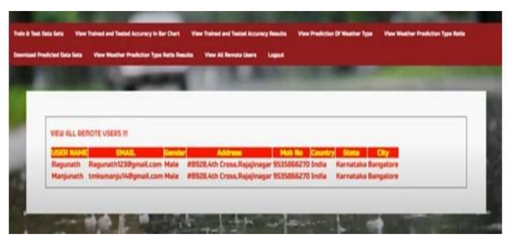
However, it seems sturdy and effective in spite of this. It performs similarly to other methods of guided learning. Numerous explanations have been put forward in the literature. We emphasise a representation bias-based explanation in this lesson. Along with logistic regression, linear discriminant analysis, and linear SVM (support vector machine), the naive bayes classifier is a linear classifier. The technique used to estimate the classifier's parameters (the learning bias) makes a difference.

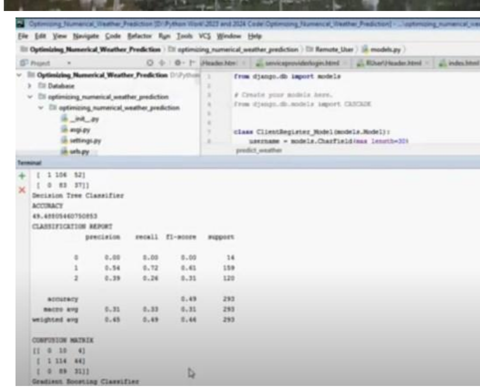
Although the Naive Bayes classifier is commonly used in research, practitioners who want to get findings that are useful do not utilise it as often. On the one hand, the researchers discovered that it is very simple to build and apply, that estimating its parameters is simple, that learning occurs quickly even on extremely big datasets, and that, when compared to other methods, its accuracy is rather excellent. The end users, however, do not comprehend the value of such a strategy and do not get a model that is simple to read and implement.

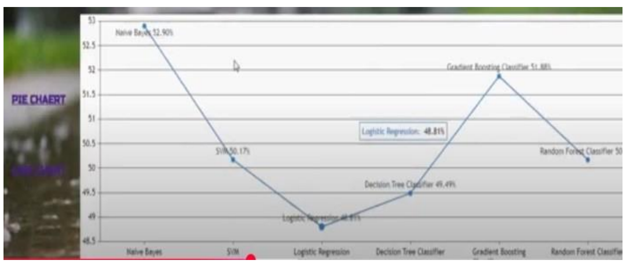
As a consequence, we display the learning process's outcomes in a fresh way. Both the deployment and comprehension of the classifier are simplified. We discuss several theoretical facets of the naive bayes classifier in the first section of this lesson. Next, we use Tanagra to apply the method on a dataset. We contrast the outcomes (the model's parameters) with those from other linear techniques including logistic regression, linear discriminant analysis, and linear support vector machines. We see that the outcomes are quite reliable. This helps to explain why the strategy performs well when compared to others. We employ a variety of tools (Weka 3.6.0, R 2.9.2, Knime 2.1.1, Orange 2.0b, and RapidMiner 4.6.0) on the same dataset in the second section. Above all, we make an effort to comprehend the outcomes.

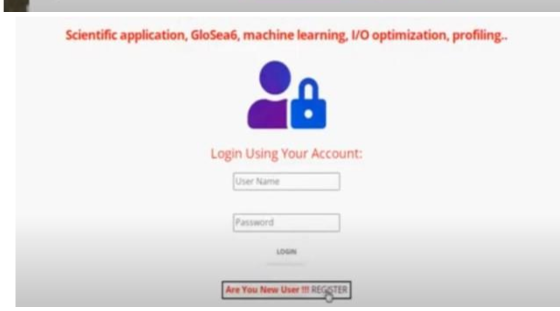
## V. SCREEN SHOTS

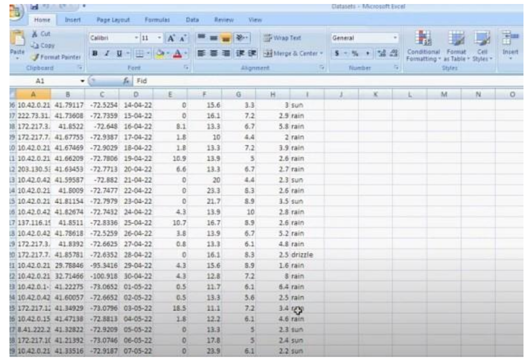


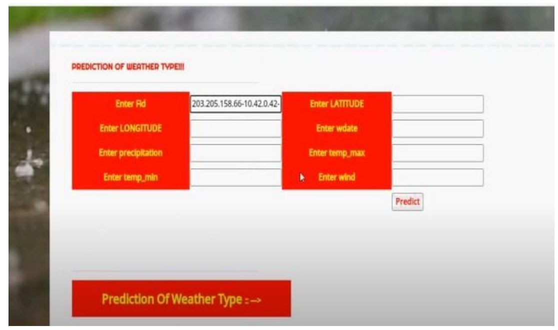



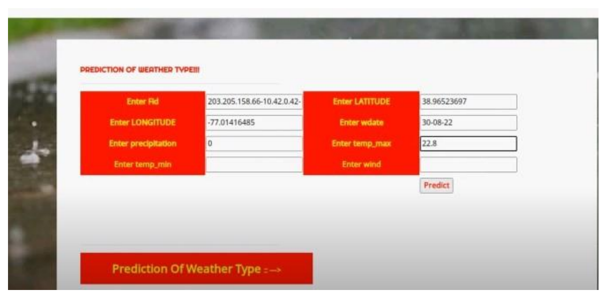


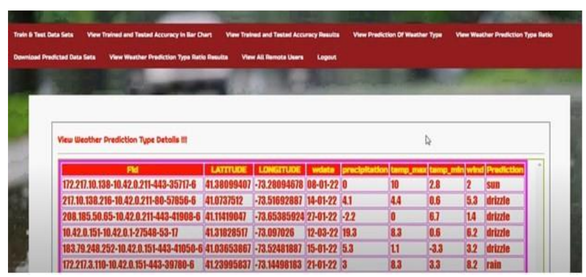



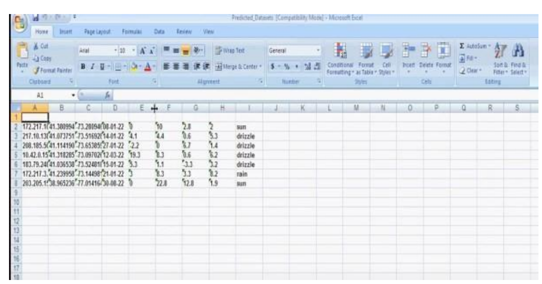







## VI. CONCLUSION

This research provided a machine learning-based method for optimising the hardware/software parameters of scientific applications. A dataset including the hardware platform parameters, the application's internal parameters, and performance data based on the combination of these two factors was created using the scientific weather forecasting program Low GloSea6 as the objective. The dataset was validated prior to the machine-learning model being applied, and the LOOCV approach was used to guarantee the validity of the regression model generated with inadequate data. Using the trained machine-learning model in a fresh research setting, the ideal hardware platform parameters and matching Low GloSea6 internal parameters were discovered, and these values matched the real parameter combinations. Specifically, the parameter combination-based projected execution time had a 16% error rate when compared to the actual execution time, indicating a significant outcome in execution time prediction. The performance of additional HPC scientific applications may be enhanced by using the suggested optimisation technique. Quantum chemistry computations, molecular dynamics (MD) simulations, and computational fluid dynamics (CFD) simulations are among the other methods in addition to weather and climate modelling. Our optimisation approach will speed up the manual performance optimisation procedure that scientists who operate such HPC research applications used to acquire from supercomputing centre workers in order to optimise their programs.

In terms of data, two study paths are described. The first step is to increase the total quantity of data. Some hardware platform data were left out of this investigation, which made it more difficult to forecast execution time accurately. Thus, gathering more I/O performance metrics and hardware/software factors might enhance model performance. Second, it would be advantageous to put into practice the benchmark-based cross-inference optimisation technique suggested in the original algorithm of this work. This would speed up data collecting and make it possible to gather parameter values that were not gathered in this research using other parameters, increasing the model's performance improvement and variety of applications.

**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, ‘‘24/7characterization 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.

[16] *R Package*. Accessed: Apr. 26, 2023. [Online]. Available: https://cran.rproject. org/src/contrib/Archive/

[17] R. Genuer, J.-M. Poggi, and C. Tuleau, ‘‘Random forests: Some methodological insights,’’ 2008, *arXiv:0811.3619*.

[18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning* (Springer Series in Statistics). New York, NY, USA: Springer, 2009, doi: 10.1007/978-0-387-84858-7.

[19] *Package RandomForest*. Accessed: Apr. 26, 2023. [Online]. Available: https://cran.r-project.org/web/packages/randomForest/random Forest.pdf

[20] *PackageGBM*. Accessed: Apr. 26, 2023. [Online].Available: https://cran.rproject. org/web/packages/gbm/gbm.pdf