# A new Approach to Prediction of Memory Leak in High-Performance Computing (HPC) using Message Passing Interface (MPI)

**Iehab Abduljabbar Kamil[1], Mohanad A. Al-Askari[2]**
[1]Department Of Information Systems, College of Computer Sciences and Information Technology, University of Anbar, IRAQ.
[2]Department Of Information Systems, College of Computer Sciences and Information Technology, University of Anbar, IRAQ.

**Article Information**

***Article history:***

Received: 17, 04, 2024
Revised: 24, 05, 2024
Accepted: 15, 06, 2024
Published: 30,06, 2024

***Keywords:***
high-performance computing,
Message Passing Interface,
Memory Leaks,
Random Forests,
Decision Trees,
AdaBoost.

**Abstract**

The study used Message Passing Interface to build a novel approach for forecasting memory leaks in HPC systems. Addressing resilient challenges the introduction focused on HPC and Messages Processing Infrastructure in system communication management. Throughout the article, it was stressed that memory leaks must be found and fixed rapidly to maintain system reliability. A greater inquiry was performed on using machine learning to identify memory leaks in HPC systems. The proposed approach involves collecting and preparing data from MPI-based high-performance computing (HPC) systems, training trained classification models for finding memory leak designs, and evaluating model performance employing appropriate indicators. To predict, the decision tree and Random forest algorithms, support vector machine models, and the AdaBoost underwent analysis. The technique included MPI metrics, feature engineering, and model training. Several algorithms helped diagnose memory deficits, but Random Forest worked best. The data and analysis showed these methods were helpful. The study found that decision tree (DT) as well as random forest (RF) algorithms could effectively diagnose and categories memory deficits with near-perfect accuracy. Random Forest methods are consistently better than baseline methods, with F-scores of 0.97 to 1.0 on the two systems. Although the baselines were finished, their F-scores may range from 0.89 to 0.97. The Random Forest technique comprised several parameters, each of which affected classification accuracy. The Random Forest approach selected the most important attributes using CPU and memory use statistics.



**Corresponding Author:**

**Iehab Abduljabbar Kamil**
Department Of Information Systems College of Computer Sciences and Information Technology, Kadraa city, Baghdad, IRAQ
Email: iehab.a.k@uoanbar.edu.iq

## 1. INTRODUCTION

High-performance computing (HPC) uses robust computer systems to solve complex problems that require much computer processing. These systems employ groupings of desktop equipment or CPUs linked via a network to compute quickly. MPI manages high-performance computing system communication. This component transmits and synchronizes actions or nodes in a distributed computer system [1].

Data transmitting and receiving is this component's goal. MPI allows concurrent parallel programming on many processors or nodes. It simplifies data transfer and synchronization. High-powered computing systems constantly enhance their velocity to extend their capabilities. High-speed computer systems are needed to find vulnerabilities in memory and analyse large datasets. MPI simplifies distributed computing node communication and coordination. In addition, it simplifies concurrent memory leak detection techniques in the supercomputer cluster. Through HPC and MPI, complex software system memory corruptions may be identified and fixed efficiently. This technique helps the targeted population diagnose and treat memory problems quickly, a significant benefit. The HPC MPI-based Prediction method improves memory leak prediction [5]. Novel ways that will enhance system efficiency and scalability were considered throughout development. Anticipation allows preventative measures to prevent memory leaks from affecting system performance.

Message Passing Interface (MPI)-based HPC systems offer specialist debugging and profiling tools. These thoroughly examine MPI process memory use, deployment patterns, and communication behaviour, making memory leak detection and diagnosis easier. Memory breaches could be identified using these tools [2]. These tools help engineers optimize and fix memory leaks by revealing their causes. MPI improves the memory leak diagnosis framework for efficiency and scalability. Adequate interaction and cooperation across distributed computing nodes allow the framework to evolve. This method improves the HPC system's dependability and predictability by reducing storage leaks and operational workload disruptions. Using the MPI, the work aims to forecast HPC memory leaks in a unique way. This method aims to identify memory leaks early to improve system reliability and efficiency. Preventing major disruptions is required to attain that objective. Memory management is crucial in high-performance computing systems to maximise computer resources and minimise system failures. Addressing the gap requires better, more accurate prediction methods tailored to these situations. In these conditions, various methods are being developed [3].

This work focuses on memory corruption within HPC platforms that utilize MPI and seeks to forecast them using a unique technique. The paper organised as follows: The background section analyses HPC and the MPI, including the current commercial problems in this field. The method describes the experiment setup in detail. Thus, these findings point the discoveries to the time, most of the problems and possible opportunities indicate the need for further research. The results section evaluates possible solutions regarding the problem applying proper methods. Conclusion part briefly summarises the findings and overall work for future recommendations.

## 2. BACKGROUND

The detection of anomalies and diagnosis in HPC application need to be enhanced with the help of unique and efficient MPI based technique to analyses memory leaks. However, significant and unique methods have caused both enhancement and to achieve the optimum outcome, memory leaks are required to be addressed and identified fast [2]. Utilization of machine learning approaches have been very helpful in addressing and identifying memory leaks and these are very significant. Memory problems may be remedied faster with this automated method. This technique reduces human review and response times, speeding resolution. The versatility and usefulness of the procedure come from its ability to find memory leaks independent of the compute nodes' programs. Concerning its adaptability, it can discover issues under various workloads without prior knowledge of the applications.

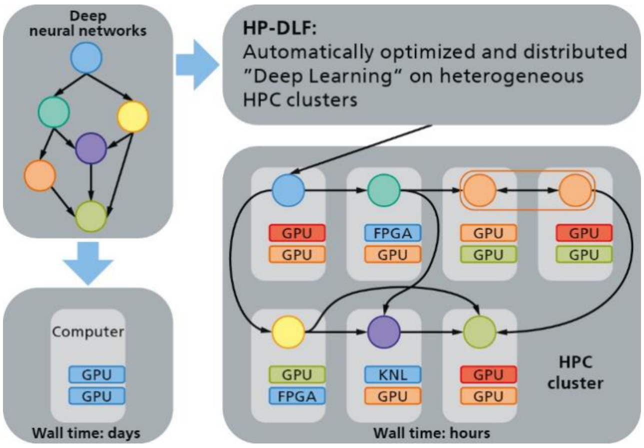


Figure 1. Deep Learning Technology in HPC [19]

Fig. 1 attached above constitutes the application of Deep Learning (DL) technology in HPC. DL analyses massive datasets on HPC using sophisticated neural networks. This improves CPU and main memory performance. Parallel computing spreads computations over several CPU cores, improving operational efficiency. Gradient-based optimization methods are used to train DL models [3]. RAM must be abundant to meet this demand. HPC requires main memory to store and retrieve data during model training and inference. This happens as the main memory efficiently stores data. CPUs provide the processing capacity needed for large-scale neural network activity.
Machine Learning approaches detect memory use abnormalities in MPI-based high-performance computing systems to avoid memory leaks. This technique differs from threshold-based monitoring and qualitative analysis used in other systems. Instead, it analyses MPI program statistics data to identify memory leaks [3]. This process occurs via data collecting. Several MPI programs run during training with various quantities of input and forms. Based on these measurements, statistical features are developed to classify the condition at which the dataset includes memory leak. Machine Learning methods, including k-nearest neighbor's and random forests, are developed on the dataset to identify memory leak trends [4].

Memory measurements are monitored during runtime, and the procedure collects statistical features. In the corresponding process, acquired attributes are added to existing machine learning models. Once a breach in memory is found, these techniques detect the irregularity and predict if the application is leaking memory. This strategy makes responding quickly, reducing resource waste and performance degradation possible. The detection approach is automated, which reduces system administrators' workload and boosts technological efficiency in the system [1]. MPI must be examined to minimize buffer loss in powerful computing devices and improve reliability and efficiency. It is necessary for improvements.

## 3. PROPOSED APPROACH

The proposed work uses machine learning to anticipate HPC memory leaks. The study technique emphasizes MPI use. MPI-based HPC data will be gathered and pre-processed in the project's early phases. Memory use trends will accompany runtime performance assessments. Following that, the supervised classifier or anomaly detection system would be trained to identify memory leak patterns. This would find memory leaks. Feature engineering extracts important features from MPI communication habits, memory allocation, and system resource use. After training, the model will forecast memory leaks both or using prior data. These forecasts are based on the training model. In HPC infrastructure, this method aims to detect and fix memory issues early. This strategy aims to improve system dependability and performance. Precision, recall, accuracy, and the F1 score can be evaluated to determine whether the provided method predictably identifies memory leaks while reducing false positives [5].
Determine whether the recommended approach works. The approach may employ Random Forest, Decision Trees, SVM, or AdaBoost. These methods are possibilities. Random Forest can handle multidimensional data and capture accurate relationships. SVMs are very efficient in binary classification and non-linear data processing. The Adaboost method, which repeatedly merges weak classifiers, may enhance high-performance computer memory leak detection. The system uses MPI runtime and memory consumption data to build a trustworthy classifier. This allows the system to create a robust classifier. In the high-performance computing environment, Adaboost iteratively modifies weights. Prioritizing misclassified cases improves prediction performance. This takes place by weighting wrongly labeled events more. The algorithms are developed utilizing MPI runtime observations and memory usage patterns to accurately identify memory breaches in HPC technology. This would enable memory leak detection.

1. *Random Forest Algorithm Equation:*
   $P(T) = \Sigma\_1\ P(Ti)$
2. *Decision Tree Algorithm Equation:*
   $P(E) = 1 - \Sigma 1\ P(Ei)$
3. *Support Vector Machine (SVM) Equation:*
   $f(x) = 1\alpha; \cdot K(x, xi) + b$
4. *AdaBoost Algorithm Equation:*
   $F(x) = \Sigma\_1\alpha t \cdot ft(x)$

Memory leaks may significantly impact HPC performance and efficiency [6]. Due to memory leakage, memory storage may decrease. New methodologies have revealed errors in memory in HPC systems that employ the MPI. These solutions actively find and rectify memory leaks to minimize system instability and performance loss. MPI can monitor memory utilization and discover unusual patterns that indicate memory leaks [4]. It prevents memory waste. Memory distribution and deallocation patterns across MPI operations can be examined to uncover irregularities that can be tagged for subsequent analysis. Runtime monitoring functions in MPI implementations provide feedback on real-time memory use. Predicting HPC memory leaks using Machine Learning has proved effective [5]. Memory utilization and system behavior data may be used to train these algorithms to identify memory leaks.

## 4. METHOD

MPI, or Message Passing Interface, has been used to establish a unique technique to minimize memory leaks in HPC systems. Abnormalities need to be identified as these indicate memory leaks. It is accomplished via Machine Learning and MPI-specific characteristics. The method begins with MPI metrics collection. These measurements include message sizes, communication patterns, and RAM usage. Skewness, kurtosis, serial correlation, and others are examples. These statistics reveal MPI transmission and memory use trends [7]. In order to enhance model prediction, the approach includes MPI-specific information such as message queue lengths and memory allocation patterns. Historical data is used to train Machine Learning models in the final step. Decision Trees, Ada-Boost model, Random Forests, and Support Vector Classifiers are applied to determine the best memory leak prediction model. The trained model analyses real-time MPI data during the program. Hence, it seeks to find memory leaks by identifying deviations from regular activity [8]. In addition, memory breach systems for computation may be addressed quickly to reduce degradation in performance and resource waste. MPI-based HPC systems may prevent memory leaks with this cutting-edge solution, which improves system reliability and efficiency.

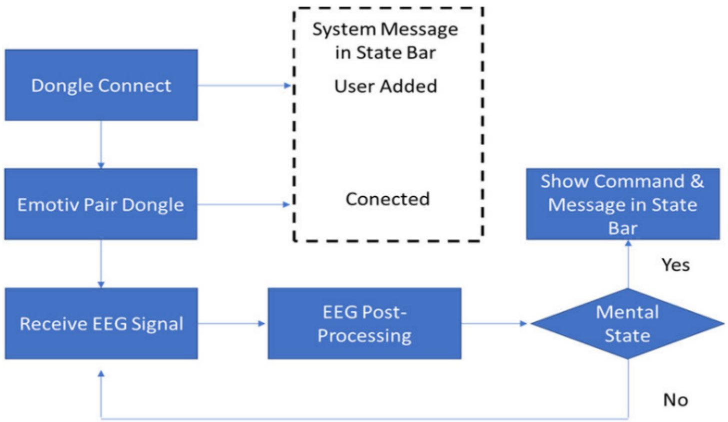


Figure 2. Block Diagram of the Analysis Procedure

The following block diagram uses a multi-step systematic approach to forecast memory leaks in HPC systems that employ the Message Passing Interface. This method is employed to achieve goal achievement. Initially, MPI metrics were required to be gathered containing message dimensions, structures, and RAM operation. In addition to message queue lengths and memory allocation patterns, statistical features are computed. These attributes include statistical traits. Leaked memories can be either unintentional or intentional, depending on the factors considered [9]. Banal or deliberate memory leaks exist. Machine learning models are used for training, testing, and feature engineering. The MPI records are analysed in real time to find abnormalities, and resource use is examined over time. In order to uncover abnormalities, both analyses are conducted. Unusual behaviour patterns may predict memory leaks. This technique detects and fixes memory problems to improve system reliability and efficiency in HPC settings. Fig. 3 demonstrates the periodic steps to conduct the memory leakage in HPC architecture with the help of machine learning algorithms. Data must be collected from MPI-based HPC systems. Additionally, runtime performance must be assessed, focusing on memory use patterns. Data is cleansed and sorted during pre-processing [7].

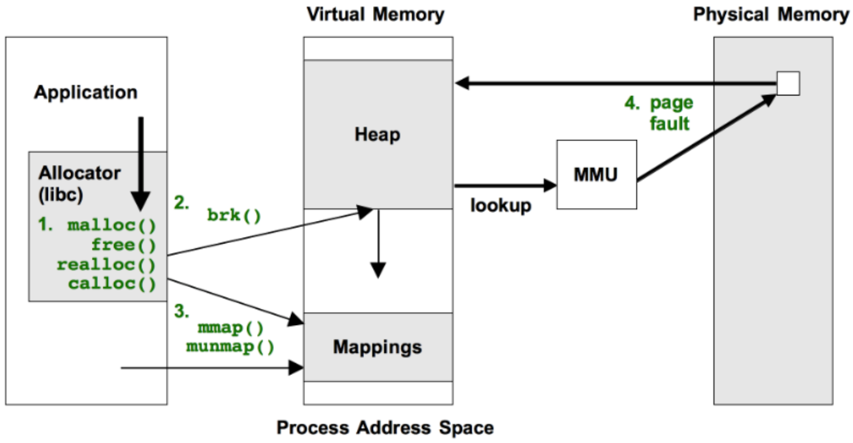


Figure 3. Stages of Performing Memory Leakage Estimation [4]

This step collects data. MPI modes of communication, allocation of memories, and system resource use provide relevant features. Next, machine learning-based prediction technologies are utilised to identify memory leak faults. Random Forest is employed because it can evaluate multidimensional data and properly capture complicated linkages. Clear decision trees are employed because they are easy to understand. Support vector machine models (SVM) have been utilised because they work well in binary classification and non-linear data processing. The Adaboost paradigm for high-performance computing memory leak detection starts with a set of weak classifiers like Decision Trees or SVMs trained on MPI execution data and memory usage characteristics [10]. Subsequently, there is the need for training of Adaboost iteratively through change in weights for cases that were misclassified. These situations must be prioritised. Each poor classifier's performance is correlated with its predecessors' effectiveness. Finally, a powerful ensemble classifier that can detect memory leaks in high-performance computing is necessary. This may be executed by integrating the weighted forecasts of all vulnerable classifiers from various sources. Identifying and fixing memory-related issues quickly is essential to improving the predictability and productivity of this high-performance computing architecture. The F1 score, accuracy, recall, and precision are evaluated to determine whether the technique is effective for discovering memory leaks while minimising false positives [11].
However, memory leaks in HPC systems utilizing MPI may be anticipated using many approaches. Different approaches may be taken using these techniques. Decision Trees split feature space into manageable groups using criteria. The algorithm employs training data to learn about the scenarios. This information helps the system classify events. Decision Trees may employ MPI measurements to find memory leak patterns [9]. Random Forests are a Decision Tree variant that builds several Decision Trees during training and aggregates their predictions. Therefore, Random Forests have several benefits over decision trees.

## 5. RESULTS AND DISCUSSION

In this section, it is explained the results of research and at the same time is given
The paper examines the ability of models such as decision trees, random forests, a SVC, and AdaBoost to detect MPI-based HPC memory leaks. These methods are implemented to find irregularities in high-performance systems for computing, particularly focusing on memory leaks [11]. The study found that window size greatly affects anomaly detection performance. By increasing the window size, more lengthy series patterns are possibly recorded, but anomalies are detected later. Expanding the window causes this delay. Narrower window widths failed to capture time series behaviour. The study suggests using a 45-second window for properly recognising target abnormalities with a moderate delay. Feature selection is stressed to reduce labour. The framework has obtained an acceptance rate of up to 44% for link clogs by employing a feature selection technique that uncovers relevant attributes for target anomaly identification [12]. A comparison of classifiers shows that random forest performs best. This is proved by the random forest's high accuracy and low false alarm rate. Random forest can spot anomalies when other classifiers can't. It's advantageous because it can detect irregularities and ignore data noise. Additionally, AdaBoost as well as SVC are highlighted, all of which possesses specific qualities that make it useful for anomaly identification.

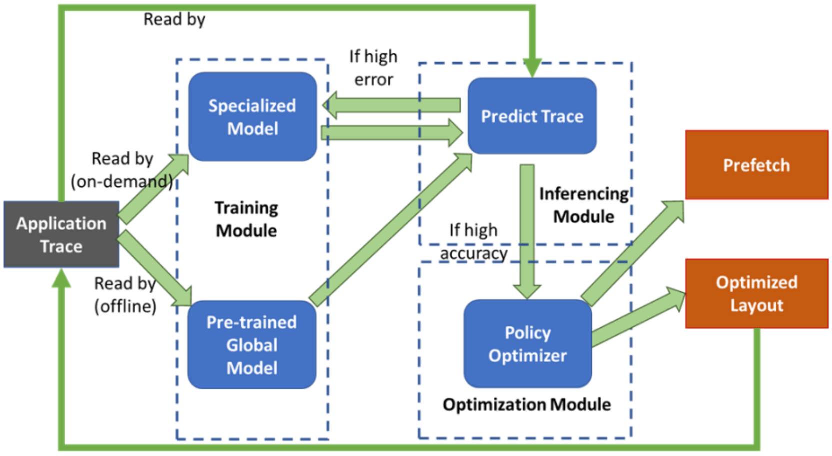


Fig. 4. Memory Error Prediction with ML Algorithms [9]

The above fig. 4, depicts the design of ML models of random forest, decision tree and adaboost to detect memory breaches. It found that Decision Tree (DT) and Random Forest (RF) technologies detect and categorize memory abnormalities with near-perfect accuracy. True, regardless of context, Random Forest methods consistently outperform baseline methods, with F-scores of 0.97 to 1.0 on the two platforms. This is true when comparing Random Forest to baseline approaches. However, baselines only attain F-scores between 0.89 and 0.97. Numerous studies have demonstrated that feature selection significantly affects algorithm performance [16]. The Random Forest method uses varied attributes, which affects classification accuracy. This is because Random Forest employs various properties. Random Forest algorithm analyses CPU and memory use information to choose the most essential attributes.

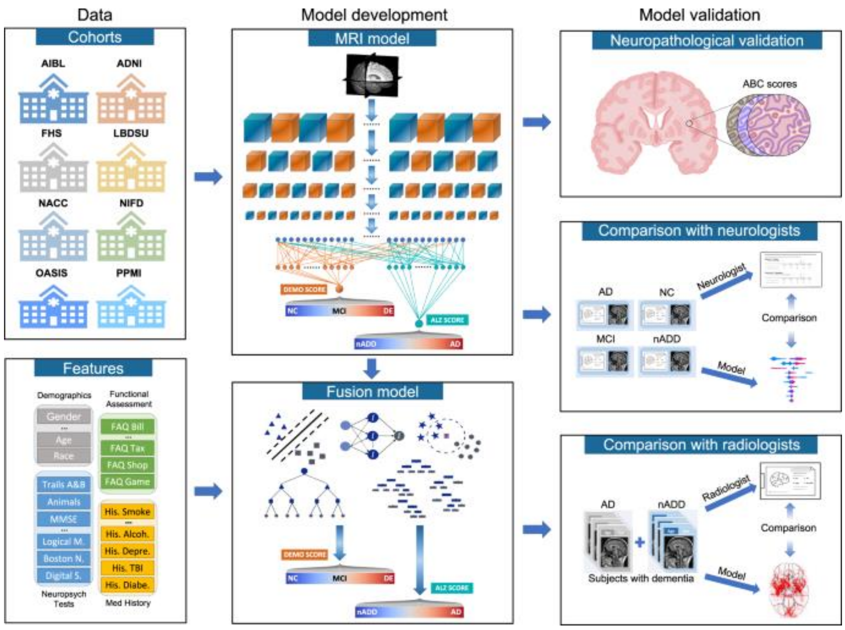


Fig. 5. Memory Abnormality Prediction Employing Machine Learning [6]

The above figure exhibits the implementation models of SVC, adaboost, random forest and decision tree by comparing each other with two bar charts. Adaboost detects aberrations with low false alarm rates and a 1.9% miss rate. The support vector machine classification technique is successful in categorising data into two different groups and managing complex data using non-linear data management methods. This makes identifying memory leaks in HPC systems that employ Message Passing Interfaces fast. It also examines how confidence affects false alarms and ignored anomalies. This emphasises the necessity to balance the two measures to guarantee accurate identification and reduce false awakenings to get the desired outcomes [15].

The study also examines the framework's capacity to tolerate unexpected input configurations, unfamiliar applications, and exceptionally high anomaly intensity. This shows the framework's robustness. The technique can identify problems even in fresh situations, indicating that it might be applied in the real world. The design provides appropriate computational speed, with random forest taking the least amount of time per window that moves. Leakage forecasting leveraging MPI-based methods in HPC environments presents several challenges and future research prospects. Choosing and adding many features for Machine Learning model training is challenging [16]. The relevance and calibre of MPI assessments and platform operational data elements are crucial for assessing the forecasting model's effectiveness. Regarding these results, it can be concluded that extended research on detailed choice of feature approaches will contribute to improvement of the prognosis of different HPC applications.

Developing prediction models that work in various computational systems and surroundings is a huge issue. Models trained on particular facts may need help generalizing to speculative abnormalities or a broad range of input values and applications. Future tests may emphasize robust, adaptable approaches that can manage a variety of high-performance computer configurations and abnormalities. Another issue is distinguishing faults with varied response intensities and symptoms. Depending on anomalies, models may behave in several ways. If this happens, favourable or unfavourable outcomes may have been misinterpreted. Anomaly detection mechanisms must be researched to increase their sensitivity and specificity to identify minute anomalies and reduce false positives [17].

The computational cost of feature generation and model training is a problem in oversized computationally intensive circumstances. In order to satisfy the targets, execution must balance forecast accuracy and computation quickness. Parallel processing and optimized techniques may be examined in the next research. This split increases real-time prediction for heavier workloads and reduces unnecessary computations. Appropriate data storage and management are essential for handling HPC systems' massive MPI metrics and system behaviour data. Prospective analysis may concentrate on improving data compression and storage methods.

Cross-disciplinary collaboration between Machine Learning experts, HPC specialists, system architects, and domain-particular app builders is needed to solve the complex memory leak prediction problems in HPC systems [18]. Future memory leak detection and prevention strategies should emphasize integrated techniques that employ insights from different fields to generate complete solutions within HPC frameworks. These methods should be included in future techniques.

## 6. CONCLUSION

MPI-based storage leak diagnosis across HPC infrastructures has yielded valuable ideas and insights. It was found that Machine Learning techniques, such as Random Forests, can discover and categorize irregularities, improving system dependability and efficiency. Nevertheless, the firm still needs help with character selection, model adaption, and processing expenses. According to the findings, the model must have a large amount of training data to operate effectively in various HPC contexts and demanding circumstances. The next phase of studies must also focus on developing forecasting algorithms that are highly adaptable and can treat a wide variety of disorders independent of symptom intensity.

The analysis compared machine learning algorithms to find memory leaks in HPC systems that utilise MPI. Decision Trees as well as Random Forests have F-scores between 0.97 and 1.0, indicating near-perfect accuracy. Obtaining F-scores from 0.97 to 1.0, the Random Forest method regularly beat baseline approaches. However, baseline scores varied between 0.89 to 0.97. Since Random Forest employed several characteristics to enhance classification accuracy, feature selection affected algorithm efficiency. Adaboost accurately detected irregularities with a low false alarm rate and a 1.9% miss rate. Overall, Random Forest predicts memory corruption in extremely efficient computing devices well.

Feature development and model training must be improved to reconcile precision in prediction and computation performance. Large-scale high-performance computing circumstances make this particularly true. In order to manage the massive MPI dimensions and system activity data, record keeping and governance must be improved. Cross-disciplinary collaboration between Machine Learning, HPC, system construction, and data application creation experts is needed to solve the complex memory leak prediction problems in HPC systems. Future endeavours should incorporate principles from several domains to build complete solutions that increase efficient computing effectiveness and dependability.

## REFERENCES

[1] P. Czarnul, J. Proficz, and A. Krzywaniak, "Energy-Aware High-Performance Computing: Survey of State-of-the-Art Tools, Techniques, and Environments," Scientific Programming, vol. 2019, pp. 1–19, 2019, doi: https://doi.org/10.1155/2019/8348791.

[2] O. Tuncer et al., "Online Diagnosis of Performance Variation in HPC Systems Using Machine Learning," IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 4, pp. 883–896, 2019, doi: https://doi.org/10.1109/tpds.2018.2870403.

[3] W. Godoy et al., "Software engineering to sustain a high-performance computing scientific application: QMCPACK," Arxiv, arxiv.org, 2023. Accessed: 2024. [Online]. Available: https://arxiv.org/pdf/2307.11502.pdf

[4] J. M. Wozniak et al., "MPI jobs within MPI jobs: A practical way of enabling task-level fault-tolerance in HPC workflows," Future Generation Computer Systems, vol. 101, pp. 576–589, 2019, doi: https://doi.org/10.1016/j.future.2019.05.020.

[5] V. Srinivasakumar, M. Vanamoorthy, S. Sairaj, and S. Ganesh, "An alternative C++-based HPC system for Hadoop MapReduce," Open Computer Science, vol. 12, no. 1, pp. 238–247, 2022, doi: https://doi.org/10.1515/comp-2022-0246.

[6] N. Zhou, H. Zhou, and D. Hoppe, "Containerisation for High Performance Computing Systems: Survey and Prospects," IEEE Transactions on Software Engineering, vol. 49, pp. 2722–2740, 2022, doi: https://doi.org/10.1109/tse.2022.3229221.

[7] J. Domke et al., "At the Locus of Performance: Quantifying the Effects of Copious 3D-Stacked Cache on HPC Workloads," ACM transactions on architecture and code optimization, vol. 20, no. 4, pp. 1–26, 2023, doi: https://doi.org/10.1145/3629520.

[8] S. Feng, X. Hua, Y. Wang, Q. Lan, and X. Zhu, "Matrix Information Geometry for Signal Detection via Hybrid MPI/OpenMP," Entropy, vol. 21, no. 12, pp. 1184–1184, 2019, doi: https://doi.org/10.3390/e21121184.

[9] A. Latif, P. Ó. Conbhuí, and E. Howley, "Applying neural networks to predict HPC-I/O bandwidth over seismic data on lustre file system for ExSeisDat," Cluster Computing, vol. 25, no. 4, pp. 2661–2682, 2021, doi: https://doi.org/10.1007/s10586-021-03347-8.

[10] T. Nakaegawa, "High-Performance Computing in Meteorology under a Context of an Era of Graphical Processing Units," Computers, vol. 11, no. 7, p. 114, 2022, doi: https://doi.org/10.3390/computers11070114.

[11] G. Zhang, A. Zhu, J. Liu, S. Guo, and Y. Zhu, "PyCLiPSM: Harnessing heterogeneous computing resources on CPUs and GPUs for accelerated digital soil mapping," Transactions in GIS, vol. 25, no. 3, pp. 1396–1418, 2021, doi: https://doi.org/10.1111/tgis.12730.

[12] E. Ates et al., "HPAS: An HPC Performance Anomaly Suite for Reproducing Performance Variations," Conference: the 48th International Conference, 2019, doi: https://doi.org/10.1145/3337821.3337907.

[13] S. Park, J. Lee, and H. Kim, "Hardware Resource Analysis in Distributed Training with Edge Devices," Electronics, vol. 9, no. 1, p. 28, Jan. 2019, doi: https://doi.org/10.3390/electronics9010028.

[14] E. Jeannot and R. Sartori, "Improving MPI Application Communication Time with an Introspection Monitoring Library," HAL (Le Centre pour la Communication Scientifique Directe), pp. 691–700, 2020, doi: https://doi.org/10.1109/ipdpsw50202.2020.00124.

[15] A. Afzal, G. Hager, G. Wellein, and S. Markidis, "Exploring Techniques for the Analysis of Spontaneous Asynchronicity in MPI-Parallel Applications," Lecture notes in computer science, pp. 155–170, 2023, doi: https://doi.org/10.1007/978-3-031-30442-2_12.

[16] D. Huang, Z. Qin, Q. Liu, N. Podhorszki, and S. Klasky, "A Comprehensive Study of In-Memory Computing on Large HPC Systems," 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), pp. 987–997, 2020, doi: https://doi.org/10.1109/icdcs47774.2020.00045.

[17] P. O. A. Navaux, A. F. Lorenzon, and M. da S. Serpa, "Challenges in High-Performance Computing," Journal of the Brazilian Computer Society, vol. 29, no. 1, pp. 51–62, 2023, doi: https://doi.org/10.5753/jbcs.2023.2219.

[18] E. Casalicchio and S. Iannucci, "The state-of-the-art in container technologies: Application, orchestration and security," Concurrency and Computation: Practice and Experience, vol. 32, no. 17, p. e5668, 2020, doi: https://doi.org/10.1002/cpe.5668.

[19] N. More, M. Galphade, V. B. Nikam, and B. Banerjee, "High-Performance Computing: A Deep Learning Perspective," EAI/Springer Innovations in Communication and Computing, pp. 247–268, Jan. 2021, doi: https://doi.org/10.1007/978-3-030-60265-9_15.

## BIOGRAPHIES OF AUTHORS

Dr. **IEHAB ABDULJABBAR KAMIL** Born in Iraq, Baghdad, 11-11-1976, he obtained a bachelor's degree from Al-Rafidain University College in computer science. He holds a master's degree from Belarusian State University of Informatics and Radioelectronics in Computer Science - Information Security. He obtained a doctorate from Tomsk State University (Russian Federation). He is fluent in English and Russian. He worked as an employee in the Ministry of Science and Technology, Iraq - Baghdad. He worked as an assistant lecturer at Saratov State University. He currently works as a lecturer at Anbar University, Department of Information Systems, and has more than 10 research papers in national and international conferences. His area of interest is Fault Tolerance, Real-Time System and Computer security. He can be contacted at email:iehab.a.k@uoanbar.edu.iq

**Dr. Mohanad A. Al**-Askari Born in Baghdad, Iraq, 1974, received the (B.S.) degree in computer science from Al-Monsour University, Iraq, in 1996, and the (M.S.) degree in information contral system and technology from Volodymur Dahl, Ukrainian, in 2013, and (PhD), Candidate, information control system and technology from Merdovskiy gosudarstveny University, Russian, in 2019, research interests include image processing, signal processing, and artificial intelligence. He can be contacted at email: mohanad.abdul@uoanbar.edu.iq