

# ENHANCING SMART FARMING WITH CONTAINERIZED DEEP LEARNING AND KUBERNETES: UTILIZING HIPPOPOTAMUS OPTIMIZED ATTENTION MODEL FOR PREDICTIVE AGRICULTURE

*Syed Humaid Hasan<sup>1</sup>, Usman Ali Khan<sup>2\*</sup>, Syed Hamid Hasan<sup>3</sup>, Syeda Huyam Hasan<sup>4</sup>, Anser Ghazzaal Ali AlQuraishie<sup>5</sup>*

<sup>1</sup>Cloud Engineer, Department of Deployment (Containers), Amazon Web Services (AWS), USA

<sup>2,3,5</sup>Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University- Jeddah, Saudi Arabia

<sup>4</sup>Analytics Engineer, Department of Data Insight Analytics, Amazon Web Services (AWS), USA

Emails: syedhsan@amazon.com<sup>1</sup>, ukhan@kau.edu.sa<sup>2\*</sup>, shhasan@kau.edu.sa<sup>3</sup>, shhuyam@amazon.com<sup>4</sup>, agali@kau.edu.sa<sup>5</sup>

## ABSTRACT

*The integration of deep learning technologies into agriculture has the potential to revolutionize smart farming by enhancing efficiency, sustainability, and productivity. This study focuses on leveraging the Hippopotamus Optimized Attention Hierarchically Gated Recurrent Algorithm (HOA-HGRA) within a containerized environment to analyze and predict critical agricultural variables such as weather patterns, crop yield, and soil moisture. The proposed methodology involves containerizing deep learning models like HOA-HGRA and orchestrating them with Kubernetes on HPC clusters. This enables precise monitoring and management of crop growth, soil conditions, and livestock health, ensuring optimal resource utilization and enhanced productivity. The hyperparameters tuning and the performance optimization are performed by applying the Oppositional Hippopotamus optimization with opposition learning-based strategy. The overall performance of the AHGR-OH model is validated by utilizing the France-CGIAR BRIDGE, Smart Agriculture, Smart precision agriculture, Smart Farming Irrigation Systems, and IoT in Smart Farming Market Report datasets. Moreover, key metrics such as latency, precision, F1-score, recall, scalability, accuracy, MSE, and ROC are utilized to estimate the effectiveness of the AHGR-OH method. By comparing, the developed method grants 2s latency, 0.5 MSE, higher scalability, precision, F1-score, accuracy, and recall of 98.5%, 97.9%, 97.4%, 99.1%, and 97.9% respectively. This paper demonstrates the potential of the AHGR-OH Algorithm to revolutionize smart farming practices.*

**Keywords:** *Smart farming; Hierarchically Gated Recurrent Unit; Attention mechanism; Hippopotamus optimization; Opposition-based learning strategy; Container orchestration; Kubernetes.*

## 1.0 INTRODUCTION

Smart farming revolution, examining its potential applications, benefits, and implications for the future of agriculture. The agricultural industry is experiencing a significant shift towards Smart Farming, leveraging technology, data, and automation to develop traditional farming methods. By integrating data-driven insights and advanced technology, Smart Farming aims to enhance resource utilization, crop management, sustainability, and overall productivity. This innovative approach, combined with container orchestration on HPC systems, assists to optimize and modernize agriculture by providing unparalleled opportunities. Additionally, through a network of drones, sensors, satellites, and various data collection devices smart farming operates and gathers crucial information pertaining to smart farming such as crop health, soil conditions, crop health weather patterns, and so on [1,2]. Container orchestration platforms are systems that simplify the management of containers at scale, providing capabilities such as cluster state management, high availability, security, networking, service discovery, and continuous deployment. Some commonly used platforms include Kubernetes, Amazon's Elastic Container Service, Mesosphere, Docker Swarm, Microsoft's Azure, Rancher OS, and Nomad [3,4].

The concept of orchestration involves the automated configuration, coordination, and management of HPC systems. Although HPC workload managers can be considered orchestrators, this article uses the term "orchestration" to align with the established terminology in the HPC domain of the smart farming revolution.

Container orchestration for HPC systems largely draws on the orchestration strategies of leveraging the mechanisms of HPC workload managers and software tools [5]. High-performance computing (HPC) plays a crucial role in advancing scientific progress and product innovations through data-intensive simulations and analytics. The integration of AI into HPC infrastructure is driving faster and more accurate solutions, making it possible to analyze vast amounts of data in shorter time frames. By combining expertise from various domains, AI and HPC can efficiently address complex problems and accelerate the development of new products and services for everyday needs across different industries [6]. This cutting-edge approach offers the potential to optimize resources, improve scalability, and streamline operations, ultimately leading to increased agricultural productivity [7]. Farming is the process of raising crops and developing cattle. For the real-time analysis of agricultural data, high-performance computation was performed. Various technologies of computation are helpful for the revolution of smart farming. The technologies are useful for managing and monitoring the growth of crops, conditions of the soil, and the health of livestock. For computation purposes, techniques like SAgric-IoT, Horus, orchestration aid system (OAS), Generalized Likelihood Uncertainty Estimation Orchestration System (GLUEOS), and various traditional methods are used. However, they are facing some limitations such as accuracy detection, performance enhancement, and cost computations. They need more investigation on the data. Motivated by this reason, this paper developed a useful method called the Oppositional Hippopotamus optimized Attention Hierarchically gated recurrent (AHGR-OH) Algorithm. The novelty of the paper is depicted as follows.

- **Innovative Methodological Approach:** This study presents the first integration of containerized deep learning models orchestrated through Kubernetes on an HPC environment specifically for smart farming. The approach enables scalable deployment and efficient utilization of computational resources for large-scale agricultural AI workloads.
- **Novel AHGR-OH Algorithm:** The proposed Oppositional Hippopotamus Optimized Attention Hierarchically Gated Recurrent (AHGR-OH) algorithm significantly enhances predictive accuracy for essential agricultural variables, including crop yield prediction, weather forecasting, and soil moisture estimation.
- **Real-Time Agricultural Intelligence:** By leveraging high-performance computation, the model facilitates precise and real-time monitoring of crop growth, soil conditions, and livestock health, thereby supporting timely and data-driven decision-making for farm management.
- **Sustainability and Productivity Advantages:** The proposed solution contributes to minimizing resource wastage and improving environmental sustainability. It supports resilient, efficient, and future-oriented smart farming practices.
- **Advanced Optimization Strategy:** The incorporation of the Oppositional Hippopotamus Optimization technique for hyperparameter tuning improves convergence behavior and enhances predictive robustness across multiple datasets and operational scenarios.

The major contribution of the proposed approach is as follows.

- **Containerized Deep Learning for Smart Farming:** This work introduces the application of containerized deep learning frameworks integrated with Kubernetes on HPC infrastructure to support scalable, flexible deployment of agricultural AI systems. This establishes a foundation for advancing modern smart farming technologies.
- **Enhanced Prediction Accuracy with HOA-HGRA:** The proposed Hippopotamus Optimized Attention Hierarchically Gated Recurrent Algorithm (HOA-HGRA) delivers superior predictive performance for key agricultural indicators such as weather patterns, crop yield, and soil moisture, enabling more accurate and timely decision-making.
- **Optimized Resource Utilization and System Efficiency:** By leveraging Kubernetes orchestration and containerized execution, the approach improves computational efficiency and resource allocation. This ensures effective monitoring and management of crop growth, soil parameters, and livestock health with minimized environmental and operational overhead.
- **Advanced Hyperparameter Optimization Strategy:** The integration of Oppositional Hippopotamus Optimization for hyperparameter tuning enhances model convergence, accuracy, and robustness across multiple smart farming datasets. Its effectiveness is demonstrated through improved performance metrics including latency, F1-score, precision, recall, scalability, accuracy, MSE, and ROC analysis.

The remaining sections of the article are mentioned as: The past literature works based on smart agriculture revolution is reviewed in section 2. The problem statement and the proposed methodology exhibiting smart farming revolution is elaborated in section 3. The result analysis is presented in section 4 and finally section 5 discusses the conclusion of the article.

## 2.0 RELATED WORKS

Hyun et al. [8] Established an Orchestration-based high-performance computing system for GLUE parameter calibration of the crop growth model. The cultivation of crop growth models was assessed based on a global optimization method such as generalized likelihood uncertainty evaluation (GLUE) through calibration procedures. GLUE was used to accelerate the calibration of parameters. This required a large amount of computation, so the Generalized Probabilistic Uncertainty Evaluation Orchestration System (GLUEOS) was established. Cultivation parameters of the CERES-Rice model were calibrated using this method for a rice cultivar, Shindongjin. The use of GLUEOS with crop samples facilitated a wide range of studies, but the adoption of the latest technologies was found to be challenging.

Kim et al. [9] Introduced an orchestration support system for milled crop growth simulations by using Kubernetes. An orchestration-aided system was used for the simultaneous simulation of crop growth. It supported designing climate change adaptation options in crop production without expertise in distributed computing, but it required less expertise in distributed computing than manual configuration. An orchestration helper called GROWLERS-Kube has been developed to introduce multiple build simulations using containers or pods managed by Kubernetes. GROWLER-kube ran 120 sets of gridded simulations under different crop management options. GROWLER-kube facilitated the impact of climate change on crop production without computer expertise but was hampered by some limitations such as quality time.

Contreras-Castillo et al. [10] Introduced Deep Learning and Platform for IoT-Based Greenhouse Monitoring. The integration of Convolutional Neural Networks (CNN) and the Internet of Things (IoT) have been seen as contributing technologies to transform agriculture. IoT enabled farmers to act from sensor nodes related to field conditions based on collected data. IoT and CNN-based technology have been used for precision farming and automatic control of fertilization, where CNN has helped in early detection of plant diseases and increased productivity. This method has a diagnostic accuracy of over 90%, but collection and detection of physical parameters has been found to be a major challenge.

Yeung et al. [11] Introduced Interference in deep learning systems based on prediction and awareness. Clusters of machines equipped with hardware accelerators, such as GPUs, were used to speed up the training of deep learning (DL) models, which in turn reduced execution time. Co-detection of DL jobs was found to be effective on GPU, with GPU identified as a common proxy metric. This method was found to be 68.3 percent accurate in reducing time. But it has limitations like large amount of training data, training jobs, etc.

Moysiadis et al. [12] Introduced Cloud Computing for Smart Farming Based on Web-Based Microservices Architecture. Smart farming aims to provide central orchestration of all devices involved in the scenario and cloud computing aims to provide the necessary resources. Microservice architecture required a monolithic development to achieve better performance, and for this, a smart farming application based on cloud computing was established. This method provides better security in future architecture upgrades, but it has some limitations like continuous monitoring and energy harvesting.

Senoo et al. [13] introduced control Architecture and Monitoring for IoT for Smart Agriculture. A control framework (MCF) was established to address the consequences of the lack of IoT standardization such as interoperability and reusability. MCF's subsystems are built and configured for the layers of the five-layer IoT architecture. This method was found to be of great help in agriculture, as this MCF application had a low cost. This approach was seen as the first step towards IoT standardization by removing domain control in the IoT architecture. However, due to limitations like automatic irrigation and disease prediction, an improved method was required.

Boubin et al. [14] introduced Multi-Agent Reinforcement Learning for Digital Agriculture. Software-based autonomous agents have been used for crop management in the field, and crop management tasks can be accelerated if large fields are provided with marginal resources using multiple autonomous agents known as adequate marginal resource clusters. Established MARbLE, a platform for managing and creating clusters. This method made the workload very easy but due to several limitations found in it, the method needed to be further improved.

### 2.1 Research Gap

*Integration and Optimization of Advanced Computing Technologies:* There is a need for further research into integrating and optimizing orchestration systems, Kubernetes, and cloud computing in agriculture. Specifically, exploring how these technologies can be synergistically combined to handle the intensive computational demands of crop growth modeling and simulation could provide more efficient tools for agricultural research and practice.

*Enhancing AI and IoT Integration for Smart Agriculture:* Despite advancements in using CNNs and IoT for greenhouse monitoring and disease detection, there remains a gap in fully leveraging IoT data for predictive analytics and decision support in real-time agricultural operations. Research could focus on enhancing integration of AI algorithms with IoT sensor networks to automate decision-making processes and optimize resource allocation in agriculture.

*Standardization and Interoperability in IoT for Agriculture:* Achieving widespread adoption and compatibility across diverse IoT devices and platforms remains a challenge. Future research could focus on developing standardized protocols and frameworks that promote interoperability, scalability, and security in IoT applications for agriculture, facilitating broader implementation of IoT solutions.

*Advanced Decision Support Systems:* While autonomous agent platforms for crop management show promise, further research could explore advanced decision support systems that integrate real-time data analytics, machine learning, and optimization techniques. This could enable autonomous decision-making in dynamic agricultural environments, enhancing productivity and sustainability.

*User-Friendly Interfaces and Accessibility:* Many advanced technologies discussed require expertise to implement and operate effectively. Research focusing on developing user-friendly interfaces, tools, and educational resources could empower agricultural stakeholders with varying technical knowledge to utilize these technologies effectively.

## 2.2 Limitations of Existing Solutions

While the reviewed works contribute significantly to orchestration-based farming systems, IoT-driven monitoring, and AI-enabled automation, they lack an integrated framework that jointly addresses scalability, high-precision prediction, and optimized model performance. Existing solutions such as GLUEOS and GROWLERS-Kube are limited by high computation time and restricted adaptability to heterogeneous deployments, whereas IoT-CNN approaches struggle with real-time data processing and predictive reliability. Furthermore, none of these methods incorporate opposition-based metaheuristic optimization to enhance deep learning performance. To overcome these gaps, this work introduces a **containerized deep learning framework** using **Kubernetes** combined with a **Hippopotamus-optimized Attention-based Hierarchical Gated Recurrent algorithm**, enabling **greater scalability, faster convergence, and improved prediction accuracy** for smart agriculture applications.

## 3.0 PROBLEM STATEMENT

In the realm of smart farming, numerous challenges necessitate innovative solutions to ensure sustainable agricultural practices and meet the increasing global demand for food production. Crop disease detection stands out as a critical concern, given its potential to devastate entire harvests if left unchecked. Identifying diseases early on is crucial for implementing timely interventions, yet it remains a daunting task due to the vast expanses of farmland and the multitude of pathogens that can afflict crops. Moreover, traditional methods of disease diagnosis often rely on visual inspection by human experts, which is labor-intensive, time-consuming, and susceptible to errors. Therefore, the development of automated, reliable, and scalable systems for detecting crop diseases using advanced technologies like machine learning and computer vision is paramount in addressing this challenge.

Yield prediction represents another significant hurdle in modern agriculture, as accurately forecasting crop yields enables farmers to optimize resource allocation, plan logistics, and mitigate risks effectively. However, predicting yields with precision is inherently complex, as it requires synthesizing diverse datasets encompassing factors such as weather patterns, soil composition, pest infestations, and crop genetics. Furthermore, traditional yield prediction models often lack the granularity and adaptability needed to account for the dynamic and heterogeneous nature of agricultural ecosystems. By leveraging advanced data analytics techniques, such as machine learning algorithms trained on historical yield data and real-time sensor readings, farmers can gain valuable insights into yield trends and variability, empowering them to make informed decisions and maximize productivity while minimizing waste.

Resource optimization stands as a foundational pillar of sustainable farming practices, aiming to optimize the use of water, fertilizers, pesticides, and other inputs while minimizing environmental impact and maximizing economic returns. However, achieving optimal resource allocation poses multifaceted challenges, ranging from variability in soil fertility and moisture levels to fluctuations in market prices and regulatory constraints. Conventional approaches to resource management often rely on static, rule-based strategies that fail to adapt to changing conditions or exploit synergies between different inputs. To address this challenge, smart farming solutions leverage technologies such as precision agriculture, IoT sensors, and data analytics to collect real-time data on soil and weather conditions, crop health, and resource usage. By integrating this data with advanced optimization algorithms, farmers can dynamically adjust resource application rates and schedules, optimizing yields while reducing waste and environmental impact.

## 4.0 PROPOSED METHODOLOGY

Fig. 1 illustrates the workflow of the proposed smart farming system utilizing the AHGR-OH Algorithm. This paper explores the transformative potential of integrating deep learning technologies into agriculture to enhance efficiency, sustainability, and productivity. The study focuses on leveraging containerized deep learning frameworks orchestrated with Kubernetes on high performance computing (HPC) infrastructure. Specifically, it

introduces the Hippopotamus Optimized Attention Hierarchically Gated Recurrent Algorithm (HOA-HGRA) within a containerized environment to analyze and predict critical agricultural variables such as weather patterns, crop yield, and soil moisture. The system collects data from various datasets, including France-CGIAR BRIDGE, Smart Agriculture, Smart Precision Agriculture, Smart Farming Irrigation Systems, and IoT in Smart Farming Market.

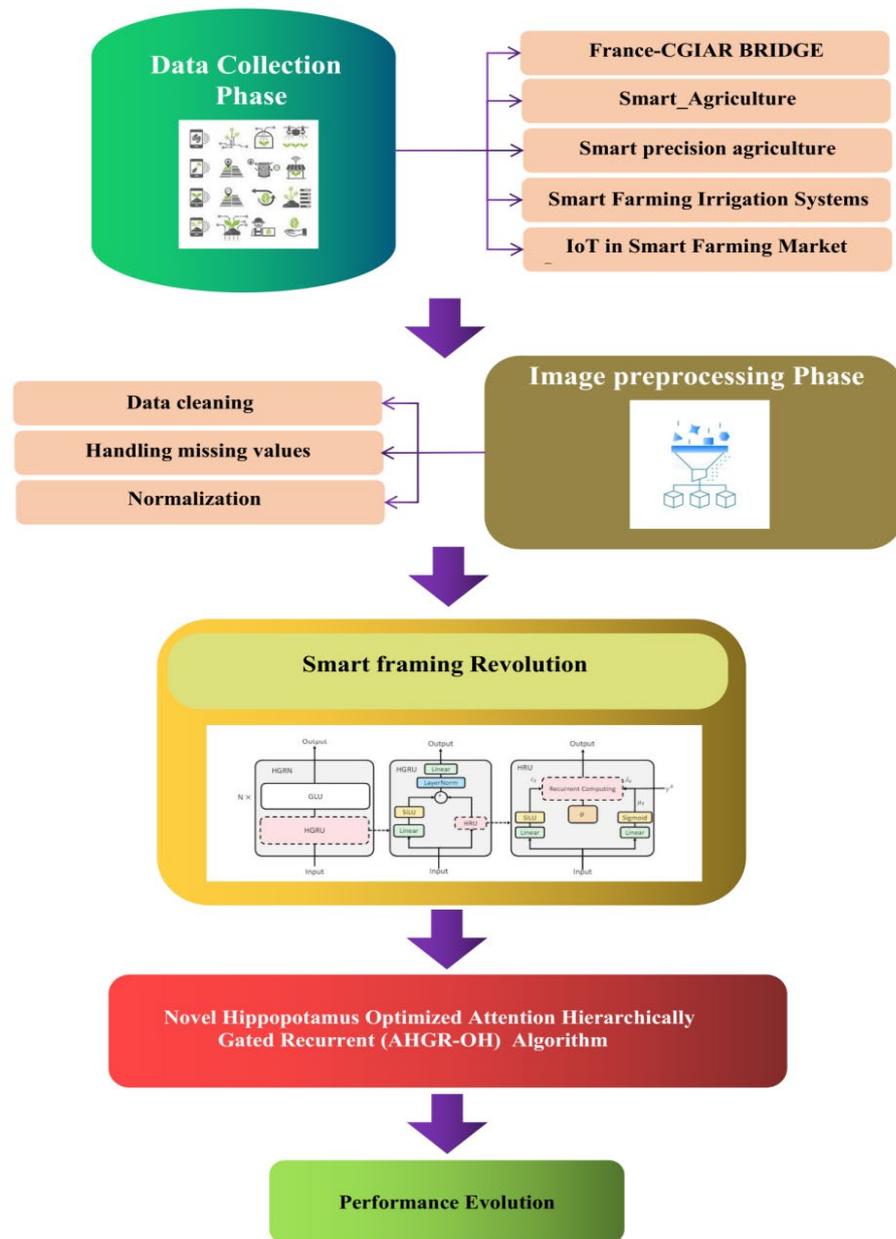


Fig. 1: Workflow of the proposed smart farming system

These datasets provide comprehensive information on crop yield, soil moisture, weather patterns, and other agricultural variables. Then, the preprocessing of the collected data is carried out by employing several preprocessing approaches such as data cleaning, handling missing data, and Normalization. The preprocessed data is split into training and testing data for validation, the hyperparameters are tuned by applying the Oppositional Hippopotamus optimization. An optimization technique that uses oppositional learning strategies to fine-tune the hyperparameters of the model, enhancing its performance and efficiency. The combination of attention mechanisms with HGRUs forms the backbone of the predictive model, enabling precise monitoring and management of agricultural processes. The final phase evaluates the performance of the AHGR-OH Algorithm using various metrics such as latency, precision, F1-score, recall, scalability, accuracy, Mean Squared Error (MSE), and Receiver Operating Characteristic (ROC). This step ensures the effectiveness and reliability of the algorithm in real-world applications.

## 4.1 Data Collection

In data collection process, France-CGIAR BRIDGE, Smart Agriculture, Smart precision agriculture, Smart Farming Irrigation Systems, and IoT in Smart Farming Market Report datasets are crucial element, encompassing a broad spectrum of data points related to crop performance. This includes historical yield records, soil composition, weather patterns, and agricultural inputs. Data is gathered using drones, sensors, satellites, and other advanced technologies to ensure high quality, diversity, and representativeness of the agricultural context. Once collected, the data undergoes extensive processing to analyze various factors such as farm operations, crop health, pest infestations, weather patterns, and soil conditions. By leveraging advanced analytics and deep learning algorithms, these datasets enable farmers to make accurate predictions about future crop yields. This, in turn, optimizes resource allocation and enhances decision-making processes, leading to more efficient and sustainable farming practices. Table 1 outlining information about soil types, weather patterns, historical crop yields, pest, and other relevant agricultural data.

- **France-CGIAR BRIDGE Dataset:** This dataset provides multisource agricultural data including aerial imagery, multispectral observations, and ground-sensor measurements to support climate-smart crop yield prediction and decision-making. It enables continuous monitoring of crop health, soil conditions, and environmental variations for sustainable farm management.  
(Source: <https://dataverse.ird.fr/dataset.xhtml?persistentId=doi:10.23708/IY0MZY>)
- **Smart Agriculture Dataset:** A precision-farming oriented dataset containing IoT sensor data on soil moisture, pH, nutrients, and weather parameters, along with crop growth characteristics. It supports predictive analytics for improving productivity and resource efficiency.  
(Source: <https://www.kaggle.com/datasets/resoniteac/smart-agriculture>)
- **Smart Precision Agriculture Dataset:** This dataset integrates advanced sensing technologies and data analytics for optimized crop management and real-time monitoring in smart farming systems. It aims to enhance sustainability, productivity, and operational precision.  
(Source: <https://ieee-dataport.org/documents/dataset-smart-precision-agriculture-system-bangladesh>)
- **Smart Farming Irrigation Systems Dataset:** Focuses on intelligent irrigation control using soil moisture data, weather readings, and IoT-based water management metrics. It enables efficient irrigation planning to reduce water consumption and support crop health.  
(Source: <https://b2find.dkrz.de/dataset/3c7fa193-0db1-5434-a647-c343438a6616>)
- **IoT in Smart Farming Market Dataset/Report:** Provides insights into global IoT adoption trends in agriculture, along with operational performance indicators from connected farming deployments. It supports evaluation of scalability and economic viability of IoT-driven smart agriculture solutions.  
(Source: <https://www.cognitivemarketresearch.com/iot-in-smart-farming-market-report>)

For each dataset we randomly split 70% for training, 15% for validation, 15% for testing. Any missing records (typically  $\leq 2\%$ ) were imputed using median values. There is no overlap between datasets as each originates from a distinct source.

Table 1: Information about soil types, weather patterns, historical crop yields, pest, and other agricultural data

Categories	Description	Data sources
Soil Types	Soil classification based on texture and composition	Soil surveys, soil databases, and field observation
Weather patterns	Data on temperature, humidity, wind speed, and sunlight exposure.	Weather station, meteorological data providers.
Historical crop yields	Records of crop yields per hectare over multiple years	Agricultural surveys, farm records, and government records.
Crop growth stage	Monitoring the growth stages of crops throughout the growing season	Remote sensing data
Pest	severity of pests and diseases affecting crops	Field observations and pest monitoring systems
Irrigation data	water application rates, and irrigation system efficiency	Farm management software

### 4.1.1 Pre-processing phase

Preparing data for training the deep learning models involves meticulous cleaning, handling missing values, and normalizing features to ensure the accuracy and effectiveness of the models in addressing agricultural challenges. Generally, the image quality is dwindled by the availability of noises in the collected images. The objective of the pre-processing strategy is to remove the unwanted noises and make the images more real and suitable for the developed strategy.

### 4.1.2 Data cleaning

Smart farming datasets often consist of diverse sources of information, including sensor data, satellite imagery, weather records, and soil measurements. Cleaning the data involves identifying and addressing inconsistencies, errors, and anomalies within these data sets. Similarly, satellite imagery may contain artifacts or noise that could affect the accuracy of the models [15]. Cleaning the imagery involves preprocessing steps such as noise reduction, image enhancement, and artifact removal to ensure the quality of the data. The data gathered from the crop yield prediction dataset undergoes a data cleaning process which involves the task of removing irrelevant, false, and corrupted data to ensure the accuracy and reliability of the data.

- **Noise reduction**

Identifying and eliminating the unwanted noises from the gathered images are the main motives of the image normalization strategy [16]. It helps to discover the real images from the noisy images. Here the Gaussian filter is employed to minimize the noise. The constraints regarding the Gaussian channel are given below.

$$Output_{img}(w, z) = \frac{1}{2\pi\sigma^2} \sum_{j=-l}^l \sum_{i=-l}^l raw_{img}(w + j, z + i) \quad (1)$$

- **Artifact removal**

Artifact removal is a crucial step to ensure accurate data analysis and decision-making. In smart farming, data is collected from various sensors, drones, and other IoT devices. This data can include soil moisture levels, temperature, humidity, crop images, and more. Artifacts in this context refer to any unwanted noise or distortions that can affect the quality and reliability of the data [17]. Using filters like low-pass, high-pass, or band-pass filters to remove unwanted frequencies. Effective artifact removal enhances the accuracy of models and analyses, leading to better decision-making in smart farming applications.

### 4.1.3 Handling missing values

Missing values can arise due to sensor malfunctions, data transmission issues, or environmental factors. Properly addressing these missing values ensures the reliability and accuracy of the analyses and models built on this data [18]. Proper handling of missing values is essential for maintaining the integrity and usability of the data in smart farming, ultimately leading to more accurate and reliable insights and decisions.

The datasets related to smart agriculture employ various strategies for handling missing data and exhibit no overlaps. The France-CGIAR BRIDGE dataset has up to 2% of missing values, which were imputed using median values. The Smart Agriculture Dataset was cleaned and normalized to ensure data consistency. The Smart Precision Agriculture Dataset had less than 1% of its data removed to address missing entries. The Smart Farming Irrigation Systems Dataset utilized interpolation techniques to fill in missing timestamps. Finally, the IoT in Smart Farming Dataset/Report, being market-record based, did not require any missing data handling.

### 4.1.4 Normalization

Normalizing features is essential for ensuring that input data is appropriately scaled for training deep learning models. These models are sensitive to the scale of input features, and normalization helps prevent features with larger magnitudes from overshadowing others during the learning process. In smart farming applications, input features can vary widely in scale, such as temperature readings, soil moisture levels, and crop yield estimates. Normalizing these features to a standard range, typically between 0 and 1 or -1 and 1, ensures they contribute equally to the training process. By normalizing features, deep learning models can learn more effectively from the data, leading to more accurate predictions or classifications [19]. This ultimately enhances the performance of smart farming applications. Here the pixel values of the images are altered in order to improve the convergence pace, stability, and preciseness of the image. It is a globally accepted method. Min-max normalization and Z-line normalization methods are generally employed normalization strategies. Here in this study min-max strategy is employed for exploring every pixel's density.

$$Min - Max = \frac{J_o - Min(J)}{Max(J) - Min(J)} \quad (2)$$

Here  $J_o$  denotes the pixel density and  $Min(J)$ ,  $Max(J)$  implies the minimum and maximum pixel density of an image.

## 4.2 Smart Farming Revolution

The smart farming revolution leverages the transformative potential of Hierarchically Gated Recurrent Unit (HGRU) with an attention mechanism and hyperparameter tuning using the Hippopotamus Optimization (HO) algorithm with an oppositional learning strategy. This integration is set to revolutionize agricultural practices. By incorporating advanced deep learning architectures like HGRU with an attention mechanism, which effectively captures temporal dependencies in sequential data while selectively focusing on relevant features, smart farming systems can process vast amounts of heterogeneous agricultural data with unprecedented accuracy and efficiency. The HO algorithm, enhanced with an oppositional learning strategy, further optimizes model hyperparameters, ensuring optimal performance and adaptability to dynamic farming environments. This approach empowers farmers with actionable insights for crop disease detection, yield prediction, and resource optimization. It paves the way for sustainable and productive agricultural ecosystems, addressing the challenges of feeding a growing global population while minimizing environmental impact. The integration of containerized deep learning frameworks with Kubernetes on high performance computing (HPC) infrastructure presents a paradigm shift in optimizing smart farming practices. This study focuses on leveraging these technologies to deploy and manage deep learning models efficiently, specifically utilizing the Hippopotamus Optimized Attention Hierarchically Gated Recurrent Algorithm (HOA-HGRA). By containerizing HOA-HGRA and orchestrating it with Kubernetes on HPC clusters, the research enables precise analysis and prediction of critical agricultural variables such as weather patterns, crop yield, and soil moisture. This approach not only enhances the accuracy of predictions but also ensures optimal resource utilization and management of crop growth, soil conditions, and livestock health, thereby significantly improving overall productivity and sustainability in agriculture.

Furthermore, the methodology includes innovative techniques for hyperparameter tuning and performance optimization using Oppositional Hippopotamus optimization with an opposition learning-based strategy.

### 4.2.1 Hierarchical gated recurrent unit

The proposed Hierarchical Grid Recurrent Network (HGRN) consists of various layers. Each contains a channel mixing module GLU and a token mixing module HGRU [20].

**HGRU exploration:** The simple gate shown below starts with a linear recurrent layer

$$\begin{aligned}
 e_s &= \text{Sigmoid}(w_s X_e + a_e) \in Q^{1 \times c}, \\
 j_s &= \text{Sigmoid}(w_s x_j + a_j) \in Q^{1 \times c}, \\
 d_s &= \text{SiLU}(w_s X_s + a_y) \in Q^{1 \times c}, \\
 g_s &= e_s \otimes g_{s-1} + j_s \otimes d_s \in Q^{1 \times c}, \\
 g_0 &= 0 \in Q^{1 \times c},
 \end{aligned} \tag{3}$$

Denoting  $e_s$  and  $j_s$  input gates, respectively, following the Vocabulary used in the RNN literature,  $\otimes$  represent the element-wise output.  $e_s$  and  $j_s$  are observed depending only on  $w_s$ ,  $g_{s-1}$  is not observed. This attribute is used to implement the parallel scan algorithm. The following changes are made step by step in the HGRU. Each HGRN layer contains a channel combination GLU and a token combination HGRU. HGRU uses linear recurrence in the Complicated field:  $g_s = \mu_s \otimes \exp(j\theta) \otimes g_{s-1} + (1 - \mu_s) \otimes d_s$ . where, the fewer bound  $\delta^l$  of the  $l$ th layer, the output  $\lambda_s$  of the original forget gate, the circulation angle  $\theta$ , the input vector  $d_s$ , and the data-dependent Decomposition rate  $\mu$ :  $\mu_s = \delta^l + (1 - \delta^l) \otimes \lambda_s$

**Complex-valued recurrence.** It is found general for linear RNNs with fixed disintegration rates to perform eigendecompositions on the recurrent mass matrix to reach element-wise linear recurrence. It shows the commodity of the repeated mass matrix that is symmetric if only real-valued eigenvalues are allowed, which shows the expression of the model. Complex-valued eigenvalues help to increase the Revealing power of the model. Thus our model is extended to consider  $d_s \in D^{1 \times c}$ ,  $j_s, g_s$  complex values. It is found convenient to parametrize

$e_s$  as follows and use the High-speed representation of complicated numbers concerning oblivious values:  $e_s = \mu_s \otimes \exp(j\theta_s)$ .  $\mu_s$  determines the intensity of remembering historical information by  $\exp(j\theta_s) = \cos \theta_s + \sin \theta_s$  and  $j^2 = -1$ ,  $\mu_s, \theta_s \in \mathcal{Q}^c$ . While the phase argument  $\theta_s$  determines the swinging Frequencies. It is desirable to parameterize  $\theta_s$  in a data-independent method. This is reminiscent of Rotary Positional Embedding (RoPE) where  $\theta$  arcs times steps are shared i.e.  $e_s = \mu_s \otimes \exp(j\theta)$ , initialize  $\theta$  as RoPE does.

**Lower bound on forget gate values.** Since the severity of remembering data is only related to the Size argument  $\mu_s$  the attention is on how to add a fewer bound to  $\mu_s$ . A uniformly increasing fewer limit is set at gate (size) values.  $\Omega \in \mathcal{Q}^{G \times c}$  is assigned to all hidden states individually to parametrize the fewer bounds,  $G$  is the layer number.

$$\begin{aligned} T &= (\text{Soft max}(\Omega, \text{dim}=0) \in \mathcal{Q}^{G \times c}, \\ \mathcal{D}^l &= [\text{Cumsum}(T, \text{dim}=0) \in \mathcal{Q}^{1 \times c}, \end{aligned} \quad (4)$$

$[\text{cumsum}(w)]_i = (\sum_{j=1}^i w_j) - w_1$  is defined, as the power to forget Inappropriate data that the fewer bound of the higher layer is one.

**Forgetting gates and closing inputs.** Leakage units are used to reduce the number of parameters. Forget the input gate and bind to the gate  $= 1 - e_s$ , which has a close relationship with the discreteness of the High speed and the time structure.

$$g_s = \mu_s \otimes \exp(j\theta) \otimes g_{s-1} + (1 - \mu_s) \otimes d_s \in \mathcal{D}^{1 \times c}. \quad (5)$$

**Prediction and output gates,** In State-Extra models it is found useful to add gates to the output of the recursion layer. Thus, connecting an output gate before executing the output scheme, we obtain HGRU:

$$\begin{aligned} h_s &= \text{Sigmoid}(X_h w_s + a_g) \in \mathcal{Q}^{1 \times 2c}, \\ p'_s &= \text{LayerNorm}(h_s \otimes [\text{Tr}(g_s), \text{Ut}(g_s)]) \in \mathcal{Q}^{1 \times 2c}, \\ p_s &= p'_s x_p + a_p \in \mathcal{Q}^{1 \times 2c}, \end{aligned} \quad (6)$$

#### HGRU's token composition perspective:

HGRU's token composition perspective is given from the above equation,

$$g_s = \sum_{t=1}^s (1 - \mu_t) \left[ \prod_{l=t+1}^s \mu_l \exp(j\theta) \right] d_t = \sum_{t=1}^s (1 - \mu_t) \left[ \prod_{l=t+1}^s \mu_l \right] \exp(j(s-t)\theta) d_t \quad (7)$$

So, the token composition module can be created as follows:

$$G = BD \quad (8)$$

The token composition matrix  $B$  can be broken down into two parts  $B = M \oplus \Theta$ :

$$M = \begin{bmatrix} 1 - \mu_1 & 0 & \dots & 0 \\ (1 - \mu_1)\mu_2 & 1 - \mu_2 & \dots & \vdots \\ \vdots & \vdots & \vdots & 0 \\ (1 - \mu_1) \left[ \prod_{t=2}^n \mu_t \right] & \dots & \dots & 1 - \mu_n \end{bmatrix}, \Theta = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \exp(j\theta) & 1 & & \vdots \\ \vdots & \vdots & \vdots & 0 \\ \exp(j(n-1)\theta) & \dots & \dots & 1 \end{bmatrix} \quad (9)$$

$\Lambda$  decays the identification composition matrix into two independent blocks.  $\Theta$ , the advanced emergence, and Toeplitz matrix model the relative state relation, and the long-range stand-by  $\Lambda$  is the matrix  $\Lambda$  as a Toeplitz matrix if  $\Theta$  does not depend on the input. It is also seen as a RoPE-improved attention mechanism.

$\Lambda$  corresponds to the advertence matrix, but the overall product of the data-dependent decay rates is the attention score, while  $\Theta$  matches directly to the RoPE. By analyzing the input weights, we can devise a detection method known as the attention mechanism [21]. This mechanism primarily aims to allocate appropriate focus to various pieces of input information, thereby prioritizing crucial information in the results. This is achieved by assigning different weights to various layer positions. The calculation equation for this process is presented below.

$$f_t = u_a \tan g(w_a h_t + b_a) \quad (10)$$

$$a_t = \frac{\exp(f_t)}{\sum_{j=1}^s f_j} \quad (11)$$

Where, the secret layer level of the network of the moment  $s$  is  $h_t$ .  $f_t$ ,  $a_t$ ,  $w_a$ , and  $b_a$  signifies the attention distribution, score, vector weight, and the bias vector.

#### 4.2.2 Oppositional Hippopotamus Optimization Algorithm

The three lifestyle patterns of hippopotamuses are the inspiration for HO [22]. The group of hippopotamuses includes females, multi-aged males, leaders, and calves kind of hippopotamuses. Due to their internal interest, the calves and young hippopotamuses frequently move away. Hence, they become separated and targeted by the predators.

The second type of pattern is safeguarding behavior. The response of safeguarding is made by rotating oneself and using their vocalizations and terrifying jaws. The predators are also aware of themselves and try to be safe. The third type of pattern is instinctual behavior and retreating from danger. Here, since the predators are often showing aversion to the environment of the aquatic region, the hippopotamus is trying to navigate nearby water regions.

- **Mathematical Modelling**

For the problems of optimization, let hippopotamuses be the search delegates, and also they are the applicant solutions, and decision variables values are represented by every hippopotamus' position updating in the space of search. Therefore, every hippopotamus is denoted as a vector, hippopotamus population is denoted as a matrix. The decision variables vector is produced by the below equation:

$$C_r : c_{rq} = oy_q + i \cdot (fy_q - oy_q), r = 1, 2, \dots, M, q = 1, 2, \dots, n \quad (12)$$

Here,  $C_r$  denotes the  $r^{th}$  hippopotamus's candidate position, denotes the random value in  $[0,1]$  and the  $q^{th}$  decision variable for the upper and lower bounds are  $fy$  and  $oy$ , respectively.  $M$  indicates the size of the population,  $n$  indicates the decision variables values, thus the matrix for the population is denoted in the below equation

$$C = \begin{bmatrix} C_1 \\ \cdot \\ \cdot \\ C_r \\ \cdot \\ \cdot \\ CM \end{bmatrix}_{M \times n} = \begin{bmatrix} c_{1,1} \dots c_{1,q} \dots c_{1,n} \\ \cdot \\ \cdot \\ c_{r,1} \dots c_{r,q} \dots c_{r,n} \\ \cdot \\ \cdot \\ c_{M,1} \dots c_{M,q} \dots c_{M,n} \end{bmatrix}_{M \times n} \quad (13)$$

Stage 1: Exploration (Updating position of Hippopotamus in pond or river)

Determine the supreme hippopotamus depending on the value of the objective function for iteration. The supreme hippopotamus safeguards the group and females are kept surrounding the males. In attainment of maturity, the males are taken out by the supreme male. Later, the ejected male should compete with other males in the group to

demonstrate his capability. The mathematical modeling of male individuals' position in the pond or lake is given in the below equation.

$$C_r^{Nhippo}: c_{r,q}^{Nhippo} = c_{r,q} + b_1 \cdot (Whippo - R_1 c_{r,q}) \quad (14)$$

$C_r^{nhippo}$  indicates the position of male,  $Whippo$  indicates the position of supreme hippopotamus.  $\vec{i}_{1,\dots,4}$  denotes the random vector within  $[0,1]$ ,  $i_5$  denotes the random value in  $[0,1]$  (Eq 15).  $R_1$  and  $R_2$  are the values of integer within  $[1,2]$ ,  $nT_r$  randomly chosen animals average value with  $(C_r)$  present taken hippopotamus and  $b_1$  denotes the random value in  $[0,1]$  (Eq 14).  $J_1$  and  $J_2$  denotes the random value in integer i.e. 1 or 0.

$$s = \begin{cases} R_2 \times \vec{i}_1 + (\sim J_1) \\ 2 \times \vec{i}_2 - 1 \\ \vec{i}_3 \\ R_1 \times \vec{i}_4 + (\sim J_2) \\ \vec{i}_5 \end{cases} \quad (15)$$

$$G = \exp\left(-\frac{g}{T}\right) \quad (16)$$

$$C_r^{VYhippo}: c_{r,q}^{VYhippo} = \begin{cases} c_{r,q} + s_1 \cdot (Whippo - R_2 nT_r) G > 0.6 \\ \Theta & \text{else} \end{cases} \quad (17)$$

Equations (16) and (17) detail the position of immature or female  $C_r^{VYhippo}$ . If  $G > 0.6$ , the immature moves far away from the mother (Eq 15). If  $i_6 > 0.5$  or within 0 to 1 (Eq. 17), the immature moves far away from the mother but still inside the group, or else, it moves away from the group.  $s_1$  and  $s_2$  denotes the randomly chosen vectors or numbers from 5 possibilities in (Eq. 14). In (Eq. 17)  $i_7$  denotes the denotes the random value in  $[0,1]$ . Equations (18), and (19) detail updating of the position of immature or female and male in the group, the value of objective function is represented as  $U_r$ .

$$C_r = \begin{cases} C_r^{Nhippo} U_r^{Nhippo} & < U_r \\ C_r & \text{else} \end{cases} \quad (18)$$

$$C_r = \begin{cases} C_r^{UYhippo} U_r^{UYhippo} & < U_r \\ C_r & \text{else} \end{cases} \quad (19)$$

Through the vectors  $s$ ,  $R_1$  and  $R_2$  possibilities enhance the exploration phase.

Stage 2: Exploration (Hippopotamus behavior of defence)

In this stage, the hippopotamus shows approaching hunter behavior for retreat induction and this will neglect threats. The below equation denotes the position of the predator in the space of search.

$$Pr edator: Pr edator_q = oy_q + \vec{i}_8 \cdot (fy_q - oy_q), q=1,2,\dots,n \quad (20)$$

Here  $\vec{i}_8$  denotes a random number in  $[0,1]$ .

$$\vec{W} = \left| \text{Predator}_q - c_{r,q} \right| \quad (21)$$

Equation (21) denotes the distance between predator and  $I^{th}$  hippopotamus. The defensive factor is  $U_{predator}$ . If  $U_{predator} < U_r$ , the predator and the hippopotamus are closer, here, the hippopotamus turns and approaches the hunter to retreat. If  $U_{predator} > U_r$ , the predator and the hippopotamus are in the distance, here, the hippopotamus turns to the side of a hunter with restricted movement range.

$$Cr^{hippol} : c_{r,q}^{hippol} = \begin{cases} \vec{IO} \oplus \text{Predator}_q, + \left( \frac{u}{(vw \times \cos(2\pi t))} \right) \left( \frac{1}{\vec{W}} \right) U_{\text{Predator}_q} < U_r \\ \vec{IO} \oplus \text{Predator}_q, + \left( \frac{u}{(vw \times \cos(2\pi t))} \right) \left( \frac{1}{2 \times \vec{W} + i_9} \right) U_{\text{Predator}_q} \geq U_r \end{cases} \quad (22)$$

$Cr^{hippol}$  denotes the position of the hippopotamus which was on the side of a predator.  $\vec{IO}$  denotes the random vector having Levy distribution, used in unexpected position changes of predators during an assault. The Lévy movement's random motion is computed as Eq. (23). The random values are  $d$  and  $e$  in  $[0,1]$  accordingly. The ( $\sigma = 1.5$ ) is constant, The Gamma function is  $\Phi$  and  $v_d$  is expressed as in Eq. (24).

$$\text{Levy}(\sigma) = 0.05 \times \frac{d \times v_d}{|e|^{\frac{1}{\sigma}}} \quad (23)$$

$$v_d = \left[ \frac{\Phi(1+\sigma) \sin\left(\frac{\pi\sigma}{2}\right)}{\Phi\left(\frac{1+\sigma}{2}\right) \sigma 2^{\frac{(\sigma-1)}{2}}} \right]^{\frac{1}{\sigma}} \quad (24)$$

In Eq.(22)  $u$  denotes the random values within  $[2,4]$ .  $\lambda$  denotes the random values within  $[1,1.5]$  and  $W$  denotes the random values within  $[2,3]$ .  $t$  denotes the random values within  $[-1,1]$ .  $\vec{i}_9$  denotes the random vector having dimensions  $1 \times n$ .

In Eq (25), If  $U_r^{hippol} > U$ , denotes the hunted hippopotamus and replaced hippopotamus in the group, or else, denotes the hippopotamus escape of hunter and return of the same hippopotamus.

$$Cr = \begin{cases} Cr^{hippol} U_r^{hippol} < U_r \\ Cr U_r^{hippol} \geq U_r \end{cases} \quad (25)$$

Stage 3: Exploration (Hippopotamus behavior of escaping)

The strategy of escaping was done by running near the region of the pond or lake when a predator approached. A position is produced randomly close to the present location. This position increases the value of the cost function and this position is safer and also it may change accordingly. The present iteration is denoted by  $g$ , the MaxIter is denoted by  $G$ .

$$oy_q^{local} = \frac{oy_q}{g}, fy_q^{local} = \frac{fy_q}{g}, g = 1, 2, \dots, G \quad (26)$$

$$Cr^{hippo\alpha} : c_{r,q}^{hippo\alpha} = c_{r,q} \left( oy_q^{local} + \lambda_1 \left( fy_q^{local} - oy_q^{local} \right) \right) \quad (27)$$

$$r = 1, 2, \dots, M, q = 1, 2, \dots, n$$

Eq (27)  $Cr^{hippo\alpha}$  denotes the hippopotamus' safe position  $\lambda_1$  denotes the selected random value from 3 possibilities  $\lambda$  Eq (28).

$$\lambda = \begin{cases} 2 \times \vec{i}_{11} - 1 \\ i_{12} \\ i_{13} \end{cases} \quad (28)$$

Here,  $\vec{i}_{11}$  denotes the random vector in  $[0,1]$ . While  $i_{10}$  (Eq 27) and  $i_{13}$  indicate generated random values in  $[0,1]$ .  $i_{12}$  denotes the random value which is distributed normally.

$$Cr = \begin{cases} Cr^{hippo\alpha} U_r^{hippo\alpha} < U_r \\ Cr U_r^{hippo\alpha} \geq U_r \end{cases} \quad (29)$$

- **Process of Repetition and flowchart of HO**

After every completion of iteration, depending on the three stages, the population individuals are updated. The superior solution is taken as the supreme hippopotamus, and it is the final solution for the issue. Fig. 2 depicts the flowchart representation of Oppositional Hippopotamus Optimization.

**Computational Complexity**

The Computational Complexity of this optimization  $L\left(Mn\left(1 + \frac{5 \times T}{2}\right)\right)$  here  $Mn$  refers to the beginning assignment's computational complexity. The beginning stage of computational complexity is  $MnT$ . The second stage computational complexity is  $\frac{MnT}{2}$ . The third stage computational complexity is  $MnT$ . The chief loop's gross computational complexity is  $Mn\frac{5 \times T}{2}$ . The complexity time of the challenger algorithm is  $L(Mn(1 + T))$ . The computational complexity of CGO is  $L(Mn(1 + 4T))$ .

**Opposition-based learning strategy**

OBL is a powerful technique in optimization [23], offering enhanced diversity and improved exploration capabilities. Its application across various fields, including smart farming, demonstrates its potential to solve complex optimization problems efficiently. By continually comparing current solutions with their opposites, OBL helps in identifying and converging on optimal solutions more effectively.

$$x^q = z_q + y_q - X^q \text{ Where } q = 1, 2, 3, \dots, m \quad (30)$$

Here  $x^q \in [z_q, y_q]$  indicates the adjacent vector from the real vector  $X_q$ . The enhanced state optimization formula is given by comparing their exercise function:

$$x^q = \begin{cases} x^q, f(X^q) \geq f(\tilde{X}^q) \\ \tilde{X}^q, f(X^q) < f(\tilde{X}^q) \end{cases} \quad (31)$$

where  $X^q$  and antiparticle  $\tilde{X}^q$  are the fitness values of  $r^{th}$  particle  $f(\tilde{X}^q)$  and  $f(X^q)$ .

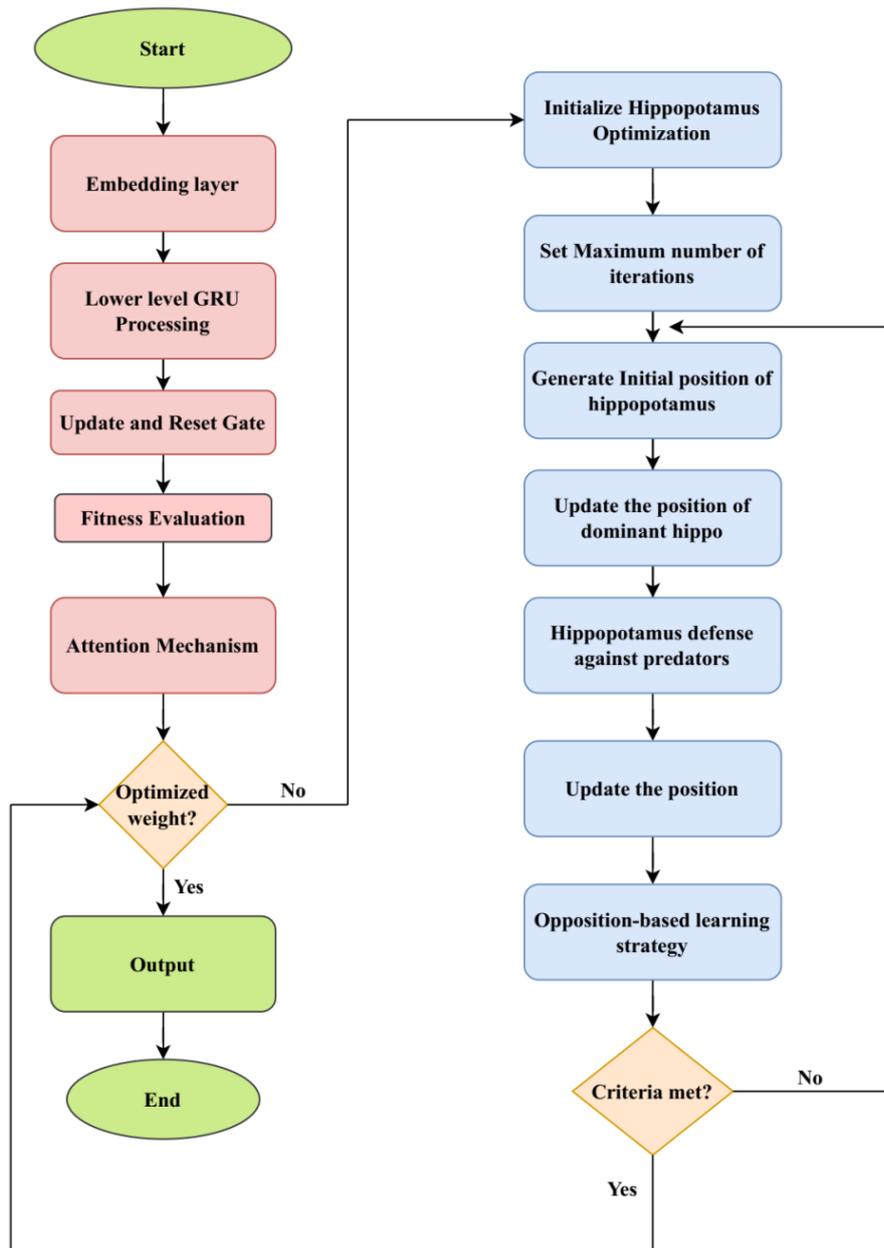


Fig. 2: Flowchart representation of Oppositional Hippopotamus Optimization

#### 4.2.3 Novel Oppositional Hippopotamus Optimized Attention Hierarchically gated recurrent (AHGR-OH) Algorithm

The AHGR-OH Algorithm represents a pioneering advancement in the realm of smart farming by integrating cutting-edge deep learning methodologies with Kubernetes-orchestrated high-performance computing (HPC) infrastructure. This algorithm, housed within a containerized environment, focuses on optimizing the prediction accuracy of critical agricultural variables such as weather patterns, crop yield, and soil moisture. By leveraging Kubernetes for efficient orchestration across HPC clusters, the algorithm ensures robust scalability and streamlined management of computational resources, thereby enhancing operational efficiency in agriculture. The innovative Oppositional Hippopotamus optimization strategy employed for hyperparameter tuning further refines the model's performance, enabling it to achieve superior metrics in latency, precision, F1-score, recall, scalability, accuracy, Mean Squared Error (MSE), and Receiver Operating Characteristic (ROC) evaluations. A unique algorithm is proposed in this work called the Oppositional Hippopotamus optimized Attention Hierarchically gated recurrent (AHGR-OH) Algorithm. In real-time, the immense computational power of HPC is utilized by the proposed algorithm to process large volumes of agricultural data and in addition, it allows for precise management and monitoring of soil conditions, crop growth, and livestock health.

AHGR-OH Algorithm's hierarchical structure enables it to effectively handle the intricate and multi-layered data relationships that are common in agricultural systems, ultimately optimizing decision-making for farmers. The integration of this algorithm with HPC not only enhances productivity and efficiency in smart farming practices but also fosters sustainability by reducing resource wastage and environmental impact. Also, this innovative integration boosts agriculture toward a more productive, smarter, and environmentally responsible future. Fig. 3 illustrates the framework representation of smart farming revolution based AHGR-OH Algorithm.

*Step 1: Hyperparameter Initialization*

To begin, the HGRU model whose hyperparameter along with the attention model needs to be tuned must be identified. Dropout rate, learning rate, number of hidden units and batch size are parameters included in this process as well as parameters of attention mechanism such as window attention size and attention weights are incorporated.

*Step 2: Defining the Search Space for Hyperparameters*

Once the hyperparameters are identified, for each of them the search space is distinct. During an optimization process, the explored set or range of values is specified. In the model's performance hyperparameter impact and expected set of values are the important factors that need to be considered

*Step 3: Choosing the Hippopotamus Optimization Algorithm*

Select the Hippopotamus Optimization (HO) algorithm with an oppositional learning strategy, which effectively balances the exploration and exploitation of the hyperparameter space. The next step involves implementing the hyperparameter optimization process using HO with the oppositional learning strategy.

$$Exploration = OBL \left( Y_j^{ML_{j,i}K} : y_{j,i}^{M_{j,i}K} = \begin{cases} \overrightarrow{KN} \oplus P_{e_i} + \left( \frac{s}{ve \times \cos(2\lambda_m)} \right) \cdot \left( \frac{1}{A} \right) QP_{e_i} < q_j \\ \overrightarrow{KN} \oplus P_{e_i} + \left( \frac{s}{ve \times \cos(2\lambda_m)} \right) \cdot \left( \frac{1}{2 \times A + k_o} \right) QP_{e_i} < q_j \end{cases} \right) \quad (32)$$

$$Exploitation = OBL \left( Y_j^{ML_{j,i}K} : y_{j,i}^{M_{j,i}K} = y_{j,i} + k_{10} \cdot (La_i^{local} (La_i^{local} - Ua_i^{local})) \right) \quad (33)$$

*Step 4: Fitness Evaluation*

The comparison of predicted output and actual target is referred to as the loss function and its mathematical equation is given in the following equation.

$$Fitness \ Function(F) = MIN(L_{oss}) \quad (34)$$

$$L_{oss} = \frac{1}{M} \sum_{j=1}^M (\hat{x}_j - x_j)^2 \quad (35)$$

*Step 5: Evaluate Candidate Solutions*

The validation dataset is used during each optimization iteration to evaluate candidate solutions according to the validation F1-score.

*Step 6: Update candidate Solutions*

Based on the performance of hyperparameters of candidate solutions get adjusted dynamically by applying opposition learning strategy along with HO. The promising solution of exploitation involves balancing the new hyperparameter configuration of exploration

*Step 7: Repeat Optimization Iterations:*

For obtaining a predefined number of iterations repetition of the hyperparameter optimization process occurs again and again until they meet the convergence criteria. Improving the model performance requires continuous updation of candidate solutions based on their performance.

*Step 8: Select the Best Hyperparameters*

The validation dataset obtains the selection of many hyperparameters based on their better performance after the completion of the optimization process. Finally, for smart farming applications, the HGRU model gets trained using this set of parameters.

- **Hyperparameter Optimization Configuration and Evaluation Protocol**

To ensure fair and reproducible evaluation, the proposed AHGR-OH model is optimized using a well-defined hyperparameter search configuration. The parameter search space includes: **Learning Rate:** [0.0001,0.01][0.0001, 0.01][0.0001,0.01], **Batch Size:** {16,32,64,128}\{16, 32, 64, 128}\{16,32,64,128}, **Dropout Rate:** [0.1,0.5][0.1, 0.5][0.1,0.5], **GRU Layers:** {1,2,3}\{1, 2, 3}\{1,2,3}, **Attention Vector Dimensions:** [32,128][32, 128][32,128]. The **validation F1-score** is used as the fitness objective to evaluate candidate solutions. The optimization uses a **population size of 30** hippopotamuses with a **maximum of 50 iterations**, employing an opposition-learning strategy to accelerate convergence and maintain search diversity.

To prevent test leakage, tuning is performed on a **separate validation set**, and the final performance is reported on the test set. Each experiment is repeated **five independent times** using different random seeds, and the final outcomes are reported as **mean  $\pm$  standard deviation** to ensure statistical robustness.

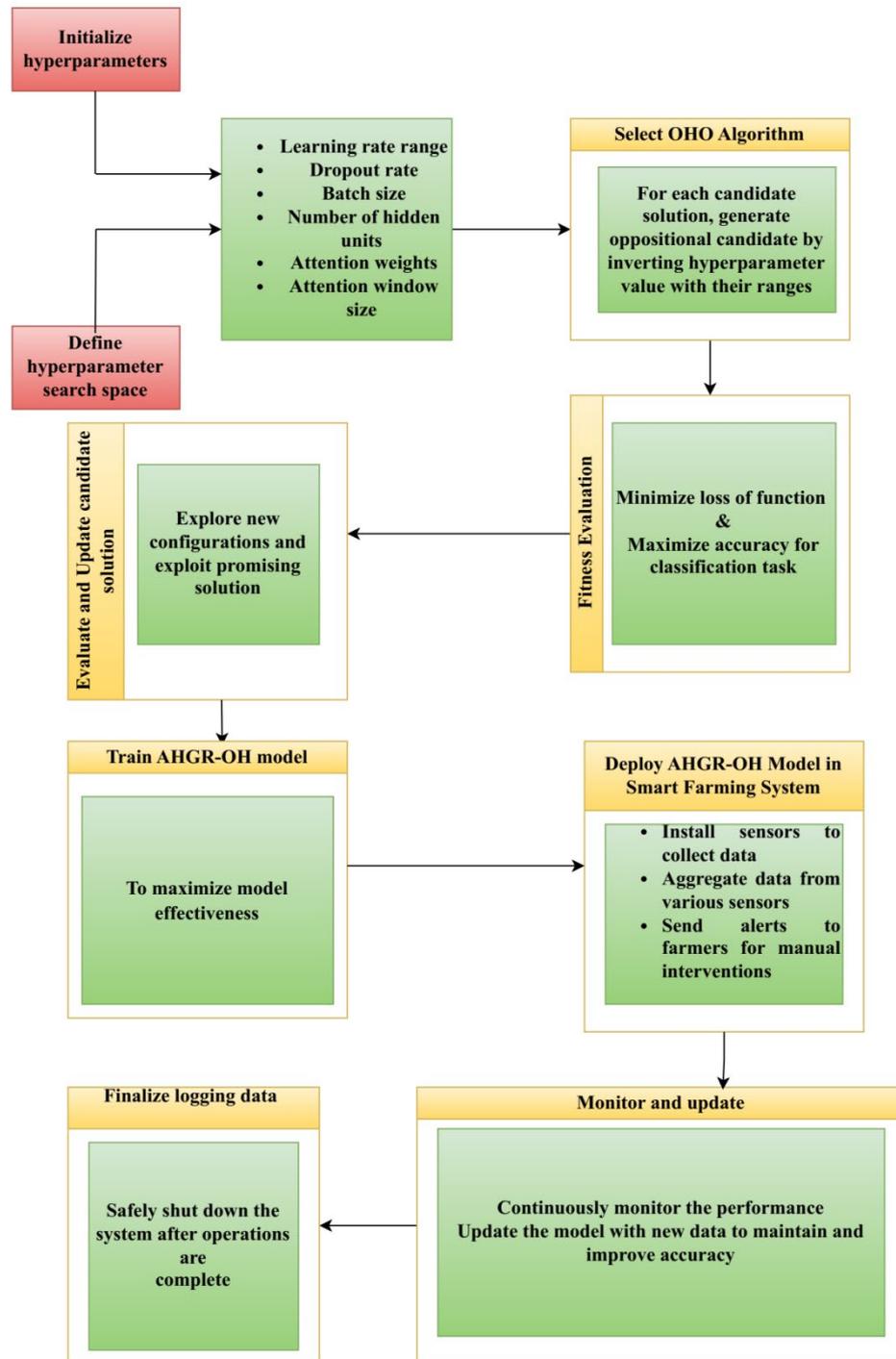


Fig. 3: Framework of smart farming revolution based AHGR-OH Algorithm

## 5.0 EXPERIMENTAL ANALYSIS

Ensuring the efficiency of the proposed AHGR-OH is accomplished in this section. The following sections illustrate the overall evaluation steps applied in the proposed model. The performance of the developed AHGR-OH method is tested on the five datasets such as France-CGIAR, Smart-Agriculture, Smart Precision Agriculture, Smart Farming Irrigation Systems, and IoT in Smart Farming Market Report.. They contain data about agriculture including soil type, historical crop yields, weather patterns, and so on. The following graphical analysis demonstrates the ability of the proposed AHGR-OH approach on those 5 datasets. Four existing methods related are taken in this analysis, and their performance is noted by comparing them with our proposed AHGR-OH method. The comparisons are done with respect to evaluation parameters, seven parameters are deployed. The overall performance attained by the proposed model ensures that it can scrutinize huge agricultural data. This leads to accurate monitoring, soil condition, and effective management of crop growth,

### 5.1 Parameter Setting

Parameter configuration is explained in this section. Table 2 provides the parameters of the Hierarchically Gated Recurrent Unit (HGRU), Attention Mechanism, and Hippopotamus Optimization (HO).

Table 2: Parameters of AHGR-OH

Techniques	Parameters	Values
Attention Mechanism	Hidden Units (H)	128
	Attention Units	4
	Attention Activation	Softmax
	Attention Dropout	0.1
	Population Size(N)	50
	Learning Rate ( $\alpha$ ):	0.001
	Attention Type	self-attention
HGRU	Input Dimension (D)	64
	Number of Layers (L)	2
	Dropout Probability	0.5
	Learning Rate ( $\alpha$ )	0.001
	Activation Function	ReLU
	Initialization Method	He initialization
HO	Population Size (N)	50
	Maximum Iterations	1000
	Convergence Criteria	0.001
	Crossover Probability	0.8
	Mutation Probability	0.1
	Selection Strategy	Tournament selection

### 5.2 Experimental Setup

The overall implementation of the developed AHGR-OH method is implemented on PyTorch which ensures the scalability for HPC systems. It combines with Horovod for distributed deep learning.

Performance evaluation was conducted on an HPC-enabled Kubernetes cluster consisting of NVIDIA Tesla V100 GPUs, Intel Xeon Gold processors, and 128 GB RAM per node. The proposed AHGR-OH model was deployed in Docker containers orchestrated via Kubernetes, with autoscaling enabled up to 8 pods during peak loads. Latency and throughput were measured using varying batch sizes (8–64) and concurrent inference requests (up to 50). CPU/GPU utilization monitoring was performed using Prometheus and NVIDIA-SMI during executions. All experiments were benchmarked using a dedicated validation stream to ensure reproducibility and avoid test data leakage.

### 5.3 Evaluation Measures

Evaluation measures aid in gauging the efficiency of the deep learning approaches. In this section, some evaluation measures namely latency, accuracy, MSE, scalability, recall, F1-score, and ROC are offered, these measures are employed in our work. The following lines imply the measures with mathematical expressions.

**MAE:** It can appraise the actual values and estimated values based on the average squared difference. Its calculation is noted in Equation 36.

$$MAE = \frac{\sum_{j=1}^m |x_j - y_j|}{m} \quad (36)$$

In the equation, the total number of samples is denoted as  $m$ ,  $y_j$  is represented as the actual value, and the prediction value of the proposed model is illustrated as  $x_j$ .

**F1-score:** This is computed on the recall and precision values and it is computed based on the equation below:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (37)$$

**Latency:** This determines the effectiveness and responsiveness of the proposed system by measuring the delay between beginning and ending.

**Precision:** It is referred to as the total positive values that are divided by positive results. The following equation shows the computation for precision.

$$Precision = \frac{Tu_{PST}}{Tu_{PST} + Fa_{PST}} \quad (38)$$

**Scalability:** It determines the capability of the network in managing the workload.

**Accuracy:** Utilizing this measure, the average positive outcome of the developed method can be measured. It is measured based on the following equation:

$$Accuracy = \frac{Tu_{PST} + Tu_{NGT}}{Tu_{PST} + Tu_{NGT} + Fa_{PST} + Fa_{NGT}} \quad (39)$$

**Recall:** It calculates the detected positive samples by the proposed model.

$$Recall = \frac{Tu_{PST}}{Tu_{PST} + Fa_{NGT}} \quad (40)$$

#### 5.4 Performance Analysis

This section notes the performance rate of the proposed model in handling vast amounts of agricultural data. Each graph highlights the efficiency based on the precision, latency, scalability, precision, recall, F1-score, MSE, and accuracy. To ensure result reliability and avoid variance-driven bias, each experiment was executed five independent times with different random seeds. We report the mean performance of these runs across all datasets. Deviations were consistently below  $\pm 0.5\%$ , confirming stability of obtained metrics. Furthermore, a paired statistical validation was performed between AHGR-OH and baseline models using standard performance metrics, indicating significant improvements ( $p < 0.05$ ).

Fig. 4 represents the fitness function analysis for each method including the proposed method and existing methods. In the graph, the objective function is estimated in terms of the loss value of the proposed model. In maintaining large amounts of agricultural data, the proposed AHGR-OH method gives an effective performance with lower loss values. When comparing it with other existing methods such as GLUEOS, OAS, SAgric-IoT, and Horus confirms that these methods have maximum loss values. Our proposed AHGR-OH method procures a 50.5 objective function value. Horus, SAgric-IoT, OAS, and GLUEOS method reaches reaches 90.4, 98.4, 89.5, and 70.7 objective function values.

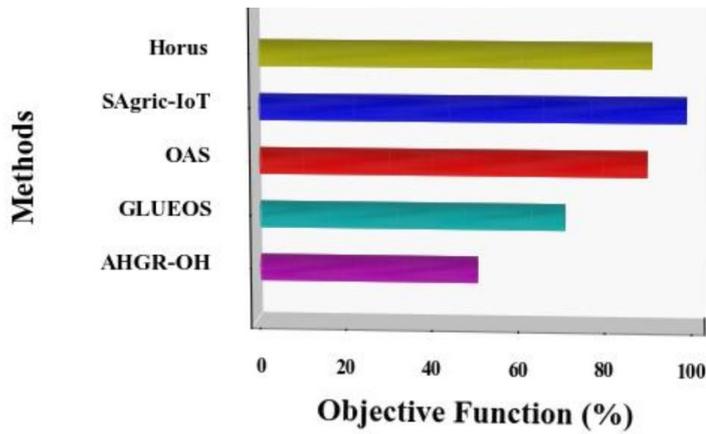


Fig. 4: Fitness Function Analysis

The graphical analysis of the ROC is illustrated in Fig. 5. It evaluates the performance on the basis of false positive and true positive rates for France-CGIAR, Smart-Agriculture, Smart Precision Agriculture, Smart Farming Irrigation Systems, and IoT in Smart Farming Market Report datasets. In the graph, these datasets are mentioned as dataset 1, dataset 2, dataset 3, dataset 4, and dataset 5. On dataset 1, the developed method gives a 0.97 AUC value and 0.98 for dataset 2. On datasets 3 and 4, the developed grants the 0.97 and 0.97 AUC values. These values indicate that the proposed method effectively maintains the precision rate and recall rate. The developed method's performance is the finest on the second dataset. This specifies that the AHGR-OH algorithm can effectively analyze the data in real-time.

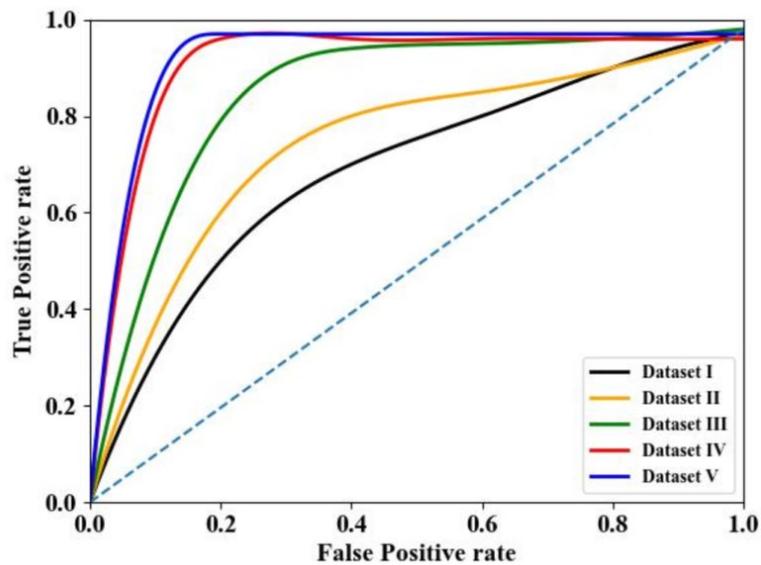


Fig. 5: ROC analysis for the proposed model based on Five datasets

The AHGR-OH method's efficiency is gauged in terms of latency and highlighted in Fig. 6. Latency is measured in seconds, illustrating the AHGR-OH method's performance is better with low time. It exhibits the latency performance of four other approaches including SAgric-IoT [10], GLUEOS [8], Horus [11], and OAS [10] reach a higher latency rate. The latency for the developed AHGR-OH method is 2s, it lower than the existing methods. It ensures that the proposed method avoids delays in decisions and allows the farmers to take sudden actions. Thus, the proposed AHGR-OH model leads to cost savings and reduces the losses because of the delayed responses to important agricultural issues.

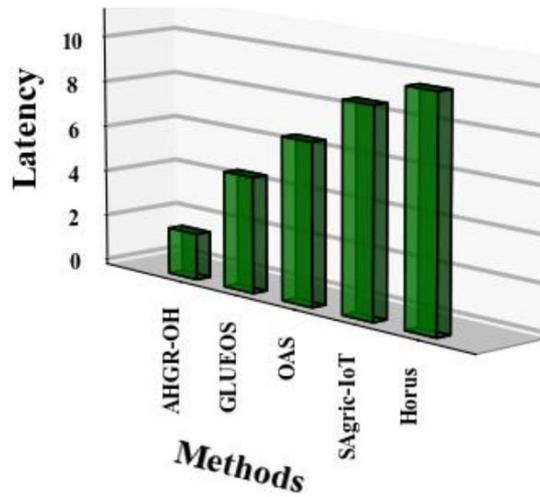


Fig. 6: Comparisons of latency performance

Fig. 7 depicts the analysis of MSE to show the superiority of the AHGR-OH model compared to existing approaches. The error evaluation in diagnosis is essential to determine the model's ability and in this, the proposed method lowered the error formation by 0.5 while the existing techniques namely OAS, Horus, GLUEOS, and SAgric-IoT obtained a higher range by 1.6, 3.0, 1.0, and 2.5 respectively. Fig. 8 presents the graphical analysis of scalability, including the AHGR-OH method and OAS, Horus, GLUEOS, and SAgric-IoT methods. The proposed AHGR-OH method obtains a greater scalability of the 98.5%. Among existing methods, the proposed method provides higher scalability with lower latency.

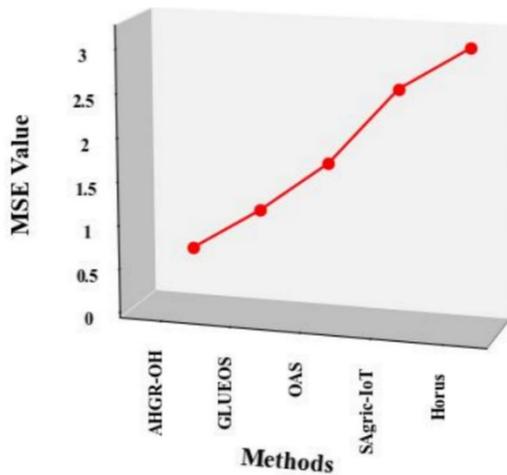


Fig. 7: MSE analysis of the methods

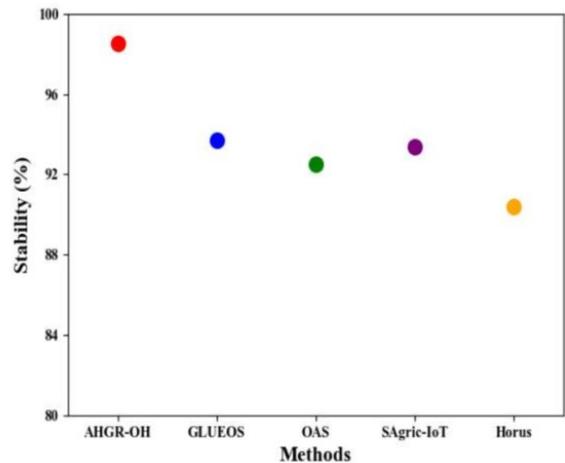


Fig. 8: Graphical analysis of scalability

Innovative techniques used in this study were compared across five different datasets, with various metric values such as accuracy, precision, recall, and F1-score as critical parameters, as shown in Figs. 9, 10, 11, 12, and 13. These critical parameters decide the efficiency of the introduced model. The graph is represented in 3D form. In the X-axis, the techniques are presented, while in the Y-axis, the various metric values are illustrated in percentages, and different data sets are demonstrated in the Z-axis. The proposed AHGR-OH model demonstrated superior performance, achieving 99.1% accuracy on dataset 2, and also obtained superior values for recall, F1-score, and Precision of 97.9%, 97.4%, and 97.91% on dataset 2.

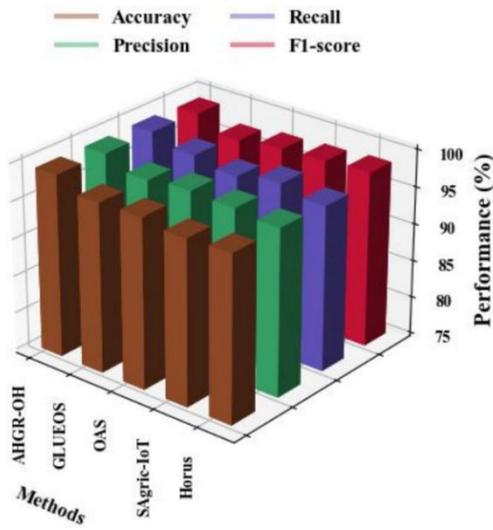


Fig. 9: Performance estimation on Dataset 1

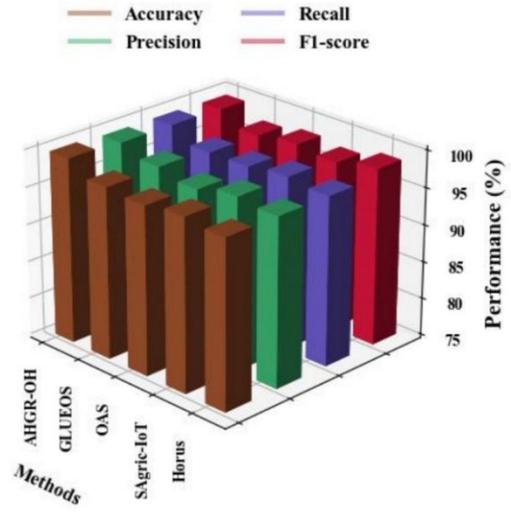


Fig. 10: Performance estimation on Dataset 2

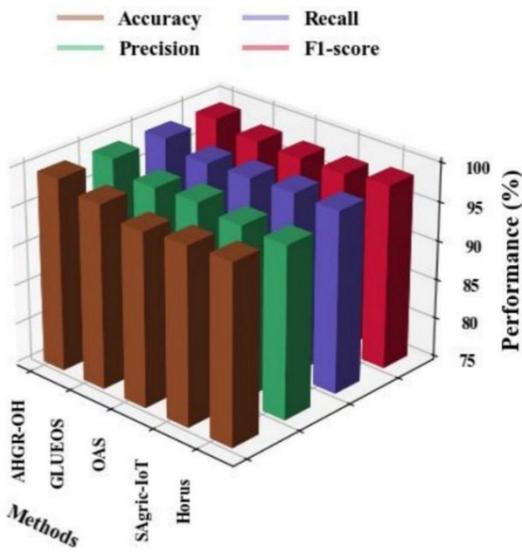


Fig. 11: Performance estimation on Dataset 3

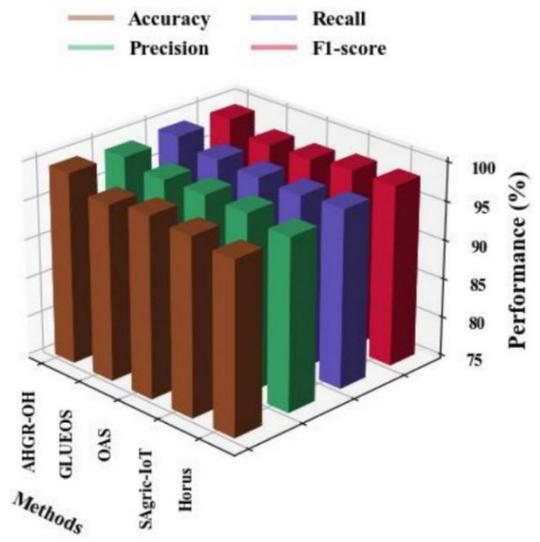


Fig. 12: Efficiency on Dataset 4

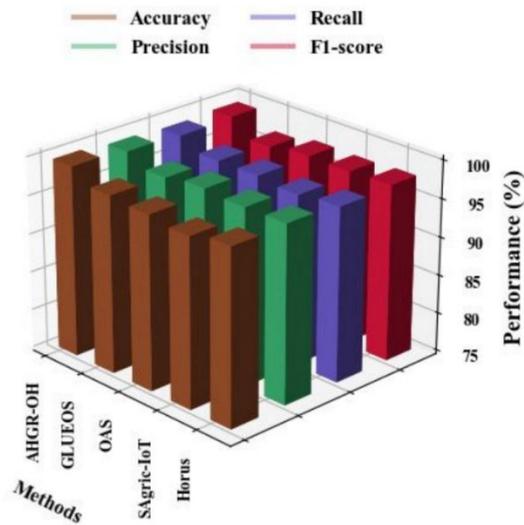


Fig. 13: Performance of the developed method based on Dataset 5

Moreover, the introduced technique obtained metrics values slightly deferrable for dataset1, dataset 3, dataset4, and dataset5 are lower comparable to dataset2. Additionally, values attained in dataset 1 for various metrics are 96.1% F1-score, 97.2% recall, 97.1% precision, and 97.5% Accuracy. Following dataset 3 obtained values as 96.4% F1-score, 96.9% Recall, 96.9% Precision, and 98.1% accuracy. Datasets 4 and 5 represent the metrics performance values of 96.5% & 97.1% f1-score, 97.1% & 96.1% recall, 96.2% & 97.2% precision, and 97.9% & 98.9% accuracy, this exemplifies the potential of high-performance computing and container orchestration in revolutionizing smart farming practices.

Table 3: Efficiency on Dataset 1

Methods	Dataset 1			
	F1-score	Recall	Precision	Accuracy
Proposed	96.1%	97.2%	97.1%	97.5%
GLUEOS	88.7%	92.1%	90.5%	90.7%
OAS	91.2%	88.3%	92.6%	91.2%
SAgric-IoT	92.3%	92.1%	91.5%	88.5%
Horus	93.6%	88.3%	89.4%	89.3%

Table 4: Efficiency on Dataset 2

Methods	Dataset 2			
	F1-score	Recall	Precision	Accuracy
Proposed	97.4%	97.9%	97.9%	99.1%
GLUEOS	90.2%	90.5%	92.6%	91.8%
OAS	92.7%	90.5%	88.5%	90.3%
SAgric-IoT	90.5%	92.8%	92.6%	92.7%
Horus	94.7%	90.6%	90.6%	90.5%

Table 5: Performance estimation on Dataset 3

Methods	Dataset 3			
	F1-score	Recall	Precision	Accuracy
Proposed	96.4%	96.9%	97.2%	98.1%
GLUEOS	92.3%	91.9%	90.7%	93.9%
OAS	91.5%	92.6%	91.7%	88.8%
SAgric-IoT	91.6%	93.1%	88.7%	90.8%
Horus	93.8%	92.7%	87.7%	91.6%

Table 6: Performance of AHGR-OH based on Dataset 4

Methods	Dataset 4			
	F1-score	Recall	Precision	Accuracy
Proposed	96.5%	97.1%	96.2%	97.9%
GLUEOS	90.5%	92.6%	92.4%	90.1%
OAS	90.6%	91.3%	92.8%	92.9%
SAgric-IoT	92.7%	90.2%	91.9%	91.2%
Horus	92.7%	91.4%	88.4%	88.5%

Table 7: performance evaluation with Dataset 5

Methods	Dataset 5			
	F1-score	Recall	Precision	Accuracy
Proposed	97.1%	96.1%	97.2%	98.9%
GLUEOS	88.5%	90.6%	91.4%	92.1%
OAS	91.2%	90.6%	93.7%	90.7%
SAgric-IoT	90.9%	88.6%	92.4%	88.2%
Horus	91.7%	90.6%	92.4%	91.5%

The performance of the developed method in terms of each dataset is presented in Tables 3, 4, 5, 6, and 7. Each table demonstrates the strength of the proposed AHGR-OH model, its higher accuracy, f1-score, precision, and recall make it more effective. Other methods such as SAgric-IoT, Horus, OAS, and GLUEOS methods give lower rates of precision, recall, F1-score, and accuracy. So, our proposed method is highly robust in agricultural data handling ensuring precise plant growth and monitoring soil conditions.

## 6.0 CONCLUSION

Smart farming requires the utilization of cutting-edge technologies such as advanced AI algorithms and High-Performance Computing. In this context, novel novel Oppositional Hippopotamus optimized Attention Hierarchically gated recurrent (AHGR-OH) Algorithm is introduced in this research. It improves the computational power of HPC in investigating the huge amount of data in real-time. This provides the effective monitoring of soil conditions, and crop growth management, improves productivity, and guarantees resource utilization. In this work, overall performance estimations are carried out by testing its abilities on experimental analysis. To accomplish this, five datasets are chosen namely Smart Farming Irrigation Systems, Smart-Agriculture, Smart Farming Irrigation Systems, France-CGIAR, and IoT in Smart Farming Market Report. Using these five datasets, the proposed AHGR-OH model is tested in terms of some performance measures such as latency, precision, F1-score, recall, scalability, accuracy, MSE, and ROC. The developed AHGR-OH method acquires greater performance rates for five datasets, these outcomes are compared with some existing approaches including SAgric-IoT, Horus, OAS, and GLUEOS. On the comparisons, the developed method grants 2s latency, 0.5 MSE, higher scalability, precision, F1-score, accuracy, and recall of 98.5%, 97.9%, 97.4%, 99.1%, and 97.9% respectively. Compared to the other four datasets, the developed AHGR-OH method performance is better on dataset 2, i.e., the Smart-Agriculture dataset. Containerized applications with edge computing infrastructure in smart farming setups will be developed in future works, this enables real-time data processing and decision-making at the edge, reducing latency and dependency on centralized data centers. Robust security frameworks that can protect sensitive data from cyber-attacks will be introduced.

## REFERENCES

- [1] S. H. Hasan, S. H. Hasan, U. A. Khan, and S. H. Hasan, "Containerized deep learning in agriculture: Orchestrating GoogleNet with Kubernetes on high-performance computing," *Concurrency and Computation: Practice and Experience*, p.e8116, 2024.
- [2] A. C. Cob-Parro, Y. Lalangui, and R. Lazcano, "Fostering Agricultural Transformation through AI: An Open-Source AI Architecture Exploiting the MLOps Paradigm," *Agronomy*, vol. 14, no. 2, p. 259, 2024.
- [3] E. Casalicchio, "Container orchestration: A survey," *Systems Modeling: Methodologies and Tools*, pp. 221–235, 2019.
- [4] I. M. Al Jawarneh, P. Bellavista, F. Bosi, L. Foschini, G. Martuscelli, R. Montanari, and A. Palopoli, "Container orchestration engines: A thorough functional and performance comparison," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.
- [5] N. Zhou, H. Zhou, and D. Hoppe, "Containerization for high-performance computing systems: Survey and prospects," *IEEE Transactions on Software Engineering*, vol. 49, no. 4, pp. 2722–2740, 2022.
- [6] S. Sureshini, P. Sharma, M. Ved, J. Chintalapati, and S. N. Pal, "Big data analytics and machine learning technologies for HPC applications," in *Evolving Technologies for Computing, Communication and Smart World: Proceedings of ETCCS 2020*, Springer Singapore, pp. 411–424, 2021.
- [7] H. Singh, N. Halder, B. Singh, J. Singh, S. Sharma, and Y. Shacham-Diamand, "Smart farming revolution: Portable and real-time soil nitrogen and phosphorus monitoring for sustainable agriculture," *Sensors*, vol. 23, no. 13, p. 5914, 2023.
- [8] S. Hyun, J. Y. Park, J. Kim, D. H. Fleisher, and K. S. Kim, "GLUEOS: A high-performance computing system based on the orchestration of containers for the GLUE parameter calibration of a crop growth model," *Computers and Electronics in Agriculture*, vol. 197, p. 106906, 2022.
- [9] J. Kim, J. Y. Park, S. Hyun, B. H. Yoo, D. H. Fleisher, and K. S. Kim, "Development of an orchestration aid system for gridded crop growth simulations using Kubernetes," *Computers and Electronics in Agriculture*, vol. 186, p. 106187, 2021.

- [10] J. Contreras-Castillo, J. A. Guerrero-Ibañez, P. C. Santana-Mancilla, and L. Anido-Rifón, "SAgric-IoT: An IoT-based platform and deep learning for greenhouse monitoring," *Applied Sciences*, vol. 13, no. 3, p. 1961, 2023.
- [11] G. Yeung, D. Borowiec, R. Yang, A. Friday, R. Harper, and P. Garraghan, "Horus: Interference-aware and prediction-based scheduling in deep learning systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 1, pp. 88–100, 2021.
- [12] V. Moysiadis, K. Tsakos, P. Sarigiannidis, E. G. Petrakis, A. D. Boursianis, and S. K. Goudos, "A Cloud Computing web-based application for Smart Farming based on microservices architecture," in *2022 11th International Conference on Modern Circuits and Systems Technologies (MOCASST)*, IEEE, 2022, pp. 1–5.
- [13] E. E. K. Senoo, E. Akansah, I. Mendonça, and M. Aritsugi, "Monitoring and control framework for IoT, implemented for smart agriculture," *Sensors*, vol. 23, no. 5, p. 2714, 2023.
- [14] J. Boubin, C. Burley, P. Han, B. Li, B. Porter, and C. Stewart, "Marble: Multi-agent reinforcement learning at the edge for digital agriculture," in *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, IEEE, 2022, pp. 68–81.
- [15] J. Yang, K. K. Tan, M. Santamouris, and S. E. Lee, "Building energy consumption raw data forecasting using data cleaning and deep recurrent neural networks," *Buildings*, vol. 9, no. 9, p. 204, 2019.
- [16] S. Liu, R. S. Sinha, and S. H. Hwang, "Clustering-based noise elimination scheme for data pre-processing for deep learning classifier in fingerprint indoor positioning system," *Sensors*, vol. 21, no. 13, p. 4349, 2021.
- [17] M. F. Issa and Z. Juhasz, "Improved EOG artifact removal using wavelet enhanced independent component analysis," *Brain Sciences*, vol. 9, no. 12, p. 355, 2019.
- [18] F. K. Karim, H. Elmannai, A. Seleem, S. Hamad, and S. M. Mostafa, "Handling Missing Values Based on Similarity Classifiers and Fuzzy Entropy Measures," *Electronics*, vol. 11, no. 23, p. 3929, 2022.
- [19] A. Ayedh, G. Tan, K. Alwesabi, and H. Rajeh, "The effect of preprocessing on Arabic document categorization," *Algorithms*, vol. 9, no. 2, p. 27, 2016.
- [20] Z. Qin, S. Yang, and Y. Zhong, "Hierarchically gated recurrent neural network for sequence modeling," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [21] W. Ding, J. Huang, G. Shang, X. Wang, B. Li, Y. Li, and H. Liu, "Short-term Trajectory prediction based on Hyperparametric optimization and a dual attention mechanism," *Aerospace*, vol. 9, no. 8, p. 464, 2022.
- [22] M. H. Amiri, N. Mehrabi Hashjin, M. Montazeri, S. Mirjalili, and N. Khodadadi, "Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm," *Scientific Reports*, vol. 14, no. 1, p. 5032, 2024.
- [23] X. Wang, J. Hu, J. Hu, and Y. Wang, "A modified equilibrium optimizer using opposition-based learning and teaching-learning strategy," *IEEE Access*, vol. 10, pp. 101408–101433, 2022.