A publication of ISCA*:
International Society for Computers
and Their Applications

INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

TABLE OF CONTENTS

***"International Journal of Computers and Their Applications is Peer Reviewed".**

Volume 32, No. 2, June 2025 **ISSN 1076-5204**

International Journal of Computers and Their Applications
A publication of the International Society for Computers and Their Applications

EDITOR-IN-CHIEF

Ajay Bandi
Associate Professor
School of Computer Science and Information Systems
Northwest Missouri State University
800 University Drive, Maryville, MO, USA 64468
Email: ajay@nwmissouri.edu

EDITORIAL BOARD

Dr. Gongzhu Hu
Central Michigan University
USA

Dr. Indranil Roy
Southeast Missouri State University
USA

Dr. Mohammad Hossain
Indiana University
USA

Dr. Nick Rahimi
University of Southern Mississippi
USA

Dr. Noor Amiri
University of Alabama
USA

Dr. Reshmi Mitra
Southeast Missouri State University
USA

Dr. Takaaki Goto
Toyo University
Japan

Dr. Venkatasivakumar Margapuri
Villanova University
USA

Hisham Al-Mubaid
University of Houston Clear Lake
USA

Oliver Eulenstein
Iowa State University
USA

Tamer Aldwari
Temple University
USA

Copyright © 2025 by the International Society for Computers and Their Applications (ISCA) All rights reserved. Reproduction in any form without the written consent of ISCA is prohibited.

ISCA Headquaters • 278 Mankato Ave, #220, Winona, Minnesota 55987, USA
Phone 507-458-4517 • Email isca@isca-hq.org • URL: http://www.isca-hq.org

Editorial

It is my distinct honor, pleasure, and privilege to serve as the Editor-in-Chief of the International Journal of Computers and Their Applications (IJCA) since 2022. I have a special passion for the International Society for Computers and their Applications. I have been a member of our society since 2014 and have served in various capacities. These have ranged from being on program committees of our conferences to being Program Chair of CATA since 2021 and currently serving as one of the Ex-Officio Board Members. I am very grateful to the ISCA Board of Directors for giving me this opportunity to serve society and the journal in this role.

I would also like to thank all the editorial board, editorial staff, and authors for their valuable contributions to the journal. Without everyone's help, the success of the journal would be impossible. I look forward to working with everyone in the coming years to maintain and further improve the journal's quality. I want to invite you to submit your quality work to the journal for consideration for publication. I also welcome proposals for special issues of the journal. If you have any suggestions to improve the journal, please feel free to contact me.

Dr. Ajay Bandi
School of Computer Science and Information Systems
Northwest Missouri State University
Maryville, MO 64468
Email: AJAY@nwmissouri.edu

In 2025, we are having four issues planned (March, June, September, and December). The next latest issue is taking shape with a collection of submitted papers.

I would also like to announce that I will begin searching for a few reviewers to add to our team. We want to strengthen our board in a few areas. If you would like to be considered, don't hesitate to get in touch with me via email with a cover letter and a copy of your CV.

Ajay Bandi, Editor-in-Chief
Email: AJAY@nwmissouri.edu

This issue of the International Journal of Computers and their Applications (IJCA) has gone through the normal review process. The papers in this issue cover a broad range of research interests in the community of computers and their applications.

IJCA Contributed Papers: This issue comprises papers that were contributed to the International Journal of Computers and their Applications (IJCA). The topics and main contributions of the papers are briefly summarized below:

Balaji Ganesh Rajagopal, Deebalakshmi Ramalingam, Yoga Vignesh, Rayean Patric, and Dharun Raagav, SRM Institute of Science and Technology, India, present their study titled "**Comprehensive Analysis of YOLO Models for Deployment in Precision Agriculture**." This research focuses on evaluating and optimizing several versions of the YOLO (You Only Look Once) object detection model to support real-time pest identification in agricultural settings. The authors examined the performance of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 using key evaluation metrics such as precision, recall, accuracy, F1-score, and mean Average Precision at IoU 0.5 (mAP@0.5). The analysis was conducted using the NBAIR dataset, which includes images of 40 pest species, with data augmentation techniques applied to improve model resilience. The findings revealed that YOLOv9 achieved the highest overall effectiveness, with a 93% accuracy, 0.959 mAP@0.5, and a 0.96 F1-score, marking it as a strong candidate for real-time deployment. YOLOv11 showed the greatest precision (0.932), while YOLOv10 balanced detection accuracy with low latency, making it ideal for mobile implementation. Although YOLOv8 lagged in performance, it remains a viable option for scenarios where further tuning is feasible. The work highlights the importance of deploying fast, lightweight, and accurate AI solutions in agriculture to promote smarter pest control, reduce dependence on chemical pesticides, and support sustainable farming practices through precision technology.

Nadia Agti, Lyazid Sabri, Faculty of Mathematics and Information Technology, Mohamed El Bachir El Ibrahimi University of Bordj Bou Arréridj, Algeria and Okba Kazar, University of Kalba, Sharjah, United Arab Emirates, present their research titled "**A Framework for an Ontological Querying-based Cognitive Perspective for Activity Recognition.**" This work introduces a novel framework that integrates semantic querying and cognitive modeling to enhance human activity recognition in smart environments. By leveraging SPARQL-based ontological queries alongside a cognitive understanding of behavior, the proposed method offers a more precise and human-centric interpretation of activities. The framework is driven by a structured Human Behavior Ontology (HBOnt), which transforms raw sensor input into semantically rich representations, incorporating key contextual elements like time and location. This approach was tested on two complex datasets—Orange4Home and CASAS Aruba—demonstrating its ability to extract meaningful behavioral patterns and support deeper analysis of intentional and goal-oriented human actions. Ultimately, the study emphasizes the value of semantic technologies and cognitive perspectives in recognizing activities, predicting health-related issues, and enabling personalized assistance in smart home environments.

Sudipta Majumder, Moirangthem Tiken Singh, Rabinder Kumar Prasad, Abhijit Boruah, Gurumayum Robert Michael, N. K. Kaphungkui, and N. Hemarjit Singh, Department of Computer Science and Engineering, Department of Electronic and Communication Engineering, Dibrugarh University Institute of Engineering and Technology, Dibrugarh University, India, present their work titled "**Heterogeneous Graph Auto-Encoder for Credit Card Fraud Detection.**" In this study, the authors address the growing challenge of fraudulent activities in digital financial systems, particularly involving credit card transactions. While conventional machine learning techniques have made strides in fraud detection, they often fall short in modeling the complex interrelations inherent in financial data. To overcome these limitations, this research proposes a novel framework that employs Graph Neural Networks (GNNs) enhanced with attention mechanisms over heterogeneous graph structures. These graphs encapsulate detailed relationships between diverse entities—such as cardholders, transactions, and merchants—providing a richer context for detecting anomalous behaviors. The model incorporates an autoencoder trained on non-fraudulent transactions to learn normal behavioral patterns, identifying suspicious deviations during reconstruction as potential fraud. This work explores two fundamental questions: the effectiveness of attention-based GNNs in detecting fraud within heterogeneous graph data, and how this approach compares to existing benchmark models. The proposed solution demonstrates strong performance, achieving an AUC-PR of 0.89 and an F1-score of 0.81, outperforming traditional methods like GraphSAGE and FI-GRL. The research highlights the potential of integrating graph-based learning with autoencoders to enhance the accuracy and robustness of modern fraud detection systems.

Ngoc-Huan Le, Minh-Nhat Huynh, Chi-Cao Ha, Duy-Nhat Le, Van-Luan Tran, Narayan C. Debnath, and Quang-Giap Nguyen, Department of Mechanical and Mechatronics Engineering, Industry 4.0 Innovation Center, Department of Electric and Electrical Engineering, School of Computing and Information Technology, Eastern International University, Binh Duong Province, Viet Nam, present their research titled "**Smart Warehouse: WMS, AI, IoT and Digital Twin Integration.**" This study introduces a next-generation smart warehouse framework that integrates modern technologies to transform warehouse operations. The proposed system combines a robust Warehouse Management System (WMS) with Artificial Intelligence (AI), Internet of Things (IoT), and Digital Twin (DT) technologies to create a synchronized, intelligent warehouse environment. The integration of these components enhances real-time data tracking, decision-making, and workflow automation. AI-driven analytics improve forecasting and inventory management, while IoT connectivity enables continuous monitoring of assets. The Digital Twin technology creates a virtual replica of the warehouse, allowing predictive analysis and operational simulation. Altogether, the system significantly improves efficiency, accuracy, and adaptability, marking a substantial advancement in supply chain and logistics infrastructure.

Priyanjli Gautam, Shuchi Mala, Aakanshi Gupta, Nidhi Mishra, and Narayan C. Debnath [1]Department of Computer Science and Engineering, ASET, Amity University, Noida, India [2]School of Computing and Information Technology, Eastern International University, Binh Duong Province, Viet Nam, present their research titled "**Reinforcement Learning for Neural System Towards Adaptive Intelligence.**" This work explores the emerging role of reinforcement learning (RL) in the medical field, with a focus on brain tumor classification. While machine learning (ML) and deep learning (DL) have shown notable success in medical imaging tasks, RL remains underexplored despite its alignment with biological learning processes such as neuronal plasticity. The study introduces a comparative framework analyzing the performance of ML, DL, and RL models—specifically Q-Learning and Deep Q-Learning (DQL)—in classifying brain tumors into four categories: glioma, meningioma, pituitary tumor, and no tumor. Experimental results demonstrate that RL approaches surpass traditional ML and DL techniques in key performance indicators including accuracy, precision, recall, and F1-score. This work emphasizes RL's biologically inspired, feedback-driven learning architecture and its superior generalization capabilities. The findings suggest RL holds substantial promise for developing adaptive, intelligent diagnostic systems that mimic cognitive processes. Beyond improved prediction performance, RL offers the potential for real-time, patient-specific diagnostic support systems in oncology and broader clinical practice.

As guest editors, we would like to express our deepest appreciation to the authors and the reviewers. We hope you will enjoy this issue of the IJCA. More information about ISCA society can be found at http://www.isca-hq.org.

Guest Editors:

Ajay Bandi, Northwest Missouri State University, USA.

JUNE 2025

Comprehensive Analysis of YOLO Models for Deployment in Precision Agriculture

Balaji Ganesh Rajagopal*, Deebalakshmi Ramalingam*,
Yoga Vignesh*, Rayean Patric*, Dharun Raagav*
*School of Computing,
SRM Institute of Science and Technology Tiruchirappalli, 621105, Tamil Nadu, India

Abstract

This research explores the optimization and deployment of YOLO (You Only Look Once) -based object detection models for real-time pest detection in agricultural environments. Four YOLO variants—YOLOv8, YOLOv9, YOLOv10, and YOLOv11—were evaluated for their performance across metrics such as precision, recall, F1-score, accuracy, and mean Average Precision (mAP@0.5). The study utilized the NBAIR dataset, encompassing 40 pest species, and applied advanced data augmentation techniques to enhance model robustness. Among the models, YOLOv9 achieved the best overall performance with 93% accuracy, 0.959 mAP@0.5, and a 0.96 F1-score, making it suitable for real-time agricultural applications. YOLOv11 demonstrated the highest precision (0.932), while YOLOv10 provided efficient latency and competitive detection capabilities, particularly on mobile devices. Although YOLOv8 underperformed in comparison, its optimization potential is noted. The findings underscore the importance of lightweight, efficient, and accurate AI models in sustainable pest management, reducing pesticide reliance, and enabling data-driven decisions in precision agriculture.

Key Words: Pest Detection, YOLO (You Only Look Once), Precision Agriculture, Mean Average Precision, Edge computing

1. Introduction

Agriculture is essential for maintaining human life, but it encounters major challenges from pest infestations, leading to considerable losses in global crop yield. Presently, pest detection techniques in agriculture mainly depend on manual identification, a process that is not only labour-intensive but also vulnerable to human error. Automated pest detection systems present a solution to these challenges, providing a quicker and more dependable method for monitoring pest populations. This project seeks to address the existing research gap by utilizing advanced object detection models to transform pest detection methodologies(Singh et al., 2024).

This project is driven by the critical necessity to enhance agricultural efficiency and reduce crop losses. On time and precise identification of pests minimizes dependence on widespread pesticide use, allowing farmers to focus on particular threats, thus enhancing resource efficiency and promoting environmental sustainability. This project seeks to enhance pest management strategies through the integration of advanced machine learning models, focusing on precision and automation in pest detection.

The idea for this project significantly improves sustainable agriculture through the encouragement of environmentally conscious farming methods. Precise identification of pests minimizes reliance on chemical pesticides, which often negatively impact non-target organisms and contribute to soil and water contamination. Moreover, prompt identification reduces crop loss, enhances food security, and decreases waste.

Artificial intelligence (AI) has emerged as an essential component of contemporary agricultural methods, due to its capacity to analyze extensive datasets and produce practical insights. AI applications have revolutionized traditional farming, ranging from precision irrigation to automated harvesting. Object detection, a specialized area within artificial intelligence, is essential for recognizing and monitoring pests, diseases, and various elements that affect crop health.

YOLO (You Only Look Once)(Sapkota et al., 2024) was selected for this project because of its remarkable speed and precision in object detection tasks. In contrast to conventional models that rely on region-based detections, YOLO analyzes the entire image simultaneously, resulting in outstanding performance for real-time applications. This study analyzes various iterations of YOLO (v8 (Rizk & Bayad, 2023), v9, v10, and v11) to assess the development of the model and its effectiveness in pest detection (Bhatnagar et al., 2023).

Analyzing the results from different (Sapkota et al., 2024)YOLO versions is crucial for determining the most efficient model for this application. Aspects

ISCA Copyright© 2025

such as inference speed, memory efficiency, model compatibility, and detection accuracy play a critical part in the deployment of systems in practical agricultural contexts, particularly in environments with limited resources.

The selected dataset for this study includes a wide array of pest images, carefully gathered to ensure accurate representation from multiple pest species and habitats. The variety present strengthens the model's capacity to generalize and operate consistently across various agricultural contexts.

Jetson devices (Swaminathan et al., 2024)are essential in the Internet of Things (IoT) for agriculture, because of their compact design and impressive computational capabilities. Utilizing optimized models such as TensorRT-converted YOLOv8 and YOLOv9 on Jetson devices facilitates real-time pest detection in the field, positioning them as(Pham et al., 2023) excellent options for IoT-based solutions.

This project aligns with the principles of sustainable agriculture through the integration of advanced AI models and IoT technologies. This initiative provides farmers with the necessary tools to make up-to-date data-driven choices, minimizes environmental consequences, and plays a vital role in developing a more resilient food system.

2. Related Works

Several studies(Rane et al., 2024)(Huo et al., 2024)(Piancharoenwong & Badir, 2024) have explored different approaches for identifying pests and diagnosing plant diseases in the field of agriculture. Conventional methods frequently utilized image processing techniques for the analysis of visual symptoms, yet they were limited by their dependence on manually crafted features. Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NNs) are among the machine learning algorithms that have been utilized, often alongside feature extraction techniques. Although these methods enhanced accuracy relative to conventional techniques, (Deng et al., 2023)their dependence on manual feature engineering limited scalability and adaptability.

Deep learning, especially Convolutional Neural Networks (CNNs), has become a more efficient approach. Convolutional neural networks have the capability to autonomously extract features from unprocessed image data, which enhances their effectiveness in applications like pest and disease detection. The application of transfer learning, involving the fine-tuning of pre-trained models for targeted agricultural tasks, has significantly improved the efficacy of deep learning systems. Moreover, deep CNNs have demonstrated encouraging outcomes in the classification of microscopic images, broadening their use in accurate identification tasks within plant pathology.

Object detection algorithms based on YOLO (Raja Gopal & Prabhakar, 2024) have been investigated for use in agriculture, particularly in areas such as pest detection and the identification of wheat spikes. These studies emphasize YOLO’s capacity to provide real-time performance, even under demanding field conditions. Optimizing models such as YOLOv5 for deployment on devices with limited resources has shown considerable decreases in inference time while maintaining a slight reduction in accuracy. These advancements closely align with our objectives, highlighting the importance of efficient and practical solutions for real-time pest detection in sustainable agriculture. Object detection algorithms based on YOLO (Raja Gopal & Prabhakar, 2024)have been investigated for use in agriculture, particularly in areas such as pest detection and the identification of wheat spikes. These studies emphasize YOLO’s capacity to provide real-time performance, even under demanding field conditions. Optimizing models such as YOLOv5 for deployment on devices with limited resources has shown considerable decreases in inference time while maintaining a slight reduction in accuracy. These advancements closely(Donapati et al., 2023) align with our objectives, highlighting the importance of efficient and practical solutions for real-time pest detection in sustainable agriculture.

The findings from these studies illustrate the transition from conventional techniques to those based on deep learning, emphasizing the effectiveness of CNNs and YOLO models in the areas of pest detection and plant disease classification. (Li et al., 2023) Using these advancements, our study aims to enhance performance for real-time agricultural applications.

The major contributions and limitations of significant related works relevant to this study are summarized in Table 1, offering a comprehensive overview of existing methods in the context of pest detection.

Table 1 Key Contributions and Key Limitations of the Significant Related Works

Related Study	Major Contribution	Major Limitation
(Yang et al., 2023)	Implemented DPAG and FEM to enhance accuracy, substituted standard convolution with DSConv for improved speed, and utilized a variety of tomato datasets.	There are challenges related to missed detections for obscured tomatoes, constraints in mobile deployment testing, and the necessity to optimize the trade-offs related to DSConv.
(Shang et al., 2024)	Investigated sustainable pest management approaches, focusing on secondary metabolites, phytohormones, biocontrol agents, and insect pheromones.	Issues related to the expense of phytohormone treatments, the intricacies of biocontrol programs, the application of pheromone-based techniques, and the delivery and stability of RNAi.
(S. Guan et al., 2024)	Improved YOLOv10 using BiFPN for multi-scale feature integration, SEAM for attention enhancement, and GCNet for global context, resulting in superior detection of wheat spikes.	Future investigations will focus on thermal infrared imagery, advancements in 3D technology, the development of lightweight versions suitable for devices with constrained computing capabilities, and their incorporation into smart agricultural machinery.
(Mishra et al., 2024)	Advancements in the management of storage pests encompass nano-pesticides, bio-pesticides, integrated pest management strategies, and genetic control methods.	There are obstacles related to awareness, financial limitations, and the necessity for tailored strategies that address the diverse climatic conditions across India.
(H. Guan et al., 2023)	A reduced deep learning model designed for precise detection of plant diseases and pests, integrating ResNet with EfficientNetV2.	Challenges arise with complex backgrounds and restricted samples, highlighting the necessity for enhancements in robustness and generalization. Future optimization is essential for varying environmental conditions.
(Nnadozie et al., 2024)	A streamlined version of YOLOv5 has been developed for real-time crop monitoring. By eliminating certain detection scales, the model size has been reduced, enhancing speed and leading to quicker detection while maintaining minimal accuracy loss.	There is a trade-off between speed and accuracy, which may lead to reduced accuracy for object sizes that fall within discarded scales. Additional optimization techniques, such as knowledge distillation, are required.
(Zhou et al., 2024)	A comprehensive approach that combines cultural practices, biological control, genetic pest management, and precise pesticide application to promote sustainable agriculture.	Issues related to gaps in understanding pest biology, the effects of climate change, sustainability over the long term, and the implications for ecosystems. There is a necessity for enhanced awareness among farmers, along with improved social and cultural acceptance, as well as financial incentives.

(Dai et al., 2023)	Focus on accurate integrated pest management, collaboration with additional sustainable methods, involvement of the public in research, and the utilization of innovative technologies such as blockchain and artificial intelligence for improved pest observation and management.	There is a necessity for investigating innovative control methods such as RNA interference, semiochemicals, and gene editing. This research should focus on the effects of climate change and aim to tackle challenges related to awareness, economic limitations, and social influences to facilitate wider adoption of integrated pest management practices.
(Türkoğlu & Hanbay, 2019)	A new classification framework for identifying plant diseases and pests that utilizes a blend of pre-trained deep learning networks for feature extraction alongside traditional classifiers such as SVM, ELM, and KNN. This method demonstrates the ability to achieve high accuracy while maintaining computational efficiency.	Using a comparatively limited collection of images illustrating plant diseases and pests sourced from a particular area in Turkey. This restricts the applicability of the findings to different geographical areas and plant species. To validate the effectiveness of the proposed method in real-world applications, a larger and more diverse dataset is essential.
(Ebrahimi et al., 2017)	Developed and reviewed a vision-based system employing SVM classification for real-time pest detection in a greenhouse environment, attaining impressive accuracy with an error rate of less than 2.5% in identifying thrips. This method presents an exciting, possibilities for automated pest monitoring and precise pest management, enhancing sustainability and efficiency in agricultural practices.	Performed an experiment within a regulated greenhouse setting, concentrating mainly on thrips affecting strawberry plants. The applicability of the system to additional pests, crops, and outdoor settings may be restricted and requires further investigation and modification. Elements like differing lighting conditions, intricate backgrounds, and a range of pest morphologies may present obstacles for precise detection in practical agricultural environments.

3. Methodology

This study aims to create an effective and practical pest detection system by integrating advanced object detection models with optimizations specifically designed for real-world application. We perform a comparative analysis using YOLO versions 8, 9, 10, and 11 (Thakur et al., 2023)to determine the most appropriate model for pest detection. The workflow incorporates TensorRT conversion to improve inference speed and minimize computational overhead, rendering it appropriate for resource-limited settings like Jetson devices.

In addition, (Bahari et al., 2024)(Shang et al., 2024)ablation studies were carried out by eliminating elements such as the AILU activation function and Adam optimizer in YOLOv11 to assess their impact on model performance. The utilization of the diverse NBAIR dataset facilitates comprehensive model training and testing across a range of pest species, thereby enhancing generalizability in agricultural contexts. This approach aims to achieve an optimal equilibrium between precision, effectiveness, and relevance in developing sustainable agricultural practices.

3.1 Dataset

The dataset used in this study is the National Bureau of Agricultural Insect Resources (NBAIR) dataset, which includes an extensive compilation of images representing 40 different pest species frequently found in agricultural environments. This dataset provides an accurate basis for the training and assessment of object detection models aimed at identifying pests in agricultural crops. The NBAIR dataset features images that are carefully labeled and organized, providing dependable and precise annotations for machine learning purposes.

The dataset includes a diverse range of insect classes, including the Asian Lady Beetle, Ladybug, Mealy Bug, Pyrilla perpusilla, and Stink Bug. The dataset shows an imbalance, characterized by a significant variation in the number of samples across the different classes, as shown in Table 1.

The imbalance in the dataset presents difficulties for model training, as classes that lack representation can result in biased predictions. To address this issue, techniques for data augmentation were subsequently implemented to balance the class distribution, thereby enhancing the diversity of the dataset and ensuring more effective model training.

Table 2 Summary of Data Samples

S No	Insect Name	No. of Samples	No. Test Samples
1	Asian Lady Beetle	876	300
2	Ladybug	503	300
3	Mealy Bug	802	300
4	Pyrilla perpusilla	1099	300
5	Stink Bug	701	300
6	Total	3981	1500

The NBAIR dataset [Table 2] was chosen for this study because of its extensive documentation of pest species that are essential to agricultural productivity. The number of possibilities present makes it highly appropriate for developing models that require effective generalization across different field conditions and pest populations. The carefully organized and annotated dataset guarantees a high level of reliability, positioning it as an excellent option for progressing studies in pest detection.

This dataset is essential for linking artificial intelligence with practical agricultural uses, serving as a basis for creating smart systems that support farmers in sustainable pest management.

3.2 Data Augmentation

To address the class imbalance, present in the NBAIR dataset and improve the stability of the pest detection model, a range of image augmentation techniques were used. The initial dataset, comprising 3,981 samples from five insect categories, was enhanced to create a balanced dataset containing 4,500 samples (1,500 samples for each category). The augmentation techniques facilitated the model's ability to learn distinguishing features across all classes, while also reducing bias towards the majority class.

15% of the images had random grayscale conversion. This transformation enabled the model to prioritize texture and structural features over just color information, enhancing its stability against variations in lighting and color.

Images underwent zooming, and the bounding boxes of relevant objects were extracted. This method allowed the model to identify and concentrate on important objects, despite variations in scale or viewpoint.

The images underwent rotation, and the bounding boxes were modified to correspond with the repositioned objects. This allowed the model to recognize pests regardless of the angle from which they were observed.

Modifications in color intensity and the use of blurring effectively mimicked real-world scenarios such as changes in lighting or focus. The transformations contributed to the model's enhanced ability to generalize across various environments.

Mixup augmentation [Fig 1] produced unusual samples through the combination of two images along with their associated labels. This was accomplished through the utilization of a mixing factor attracted from a Beta distribution. The hyperparameter α regulated the interpolation strength, maintaining a balance between the original and mixed samples. The implementation (eqn. 1) of Mixup enhanced the model's ability to generalize by creating diverse associations between images and labels [eqn 1] [eqn 2].

$$\bar{x} = \lambda x_1 + (1 - \lambda)x_2 \quad (eqn\ 1)$$

$$\bar{y} = \lambda y_1 + (1 - \lambda)y_2 \quad (eqn\ 2)$$

Figure 1. Mixup Augmentation

RandAugment [Fig 2] implemented a series of randomly chosen transformations, including rotation, shear, and color modifications. The magnitude parameter governed the intensity of these transformations, guaranteeing that they successfully modified the images while maintaining the integrity of the objects [eqn 3].

$$\bar{x} = T_{i_N}\left(T_{i_{N-1}}\left(\dots T_{i_1}(x; M)\dots; M\right); M\right) \quad (eqn\ 3)$$

Figure 2. RandAugment Augmentation

3.3 Results of Augmentation

The application of these augmentation techniques resulted in an increase in dataset size from 3,981 images to 4,500 images, maintaining an equal distribution of 1,500 samples per class. The balanced dataset facilitated effective learning across all classes, enhancing the model's capability to identify pests under various conditions [Fig 3].

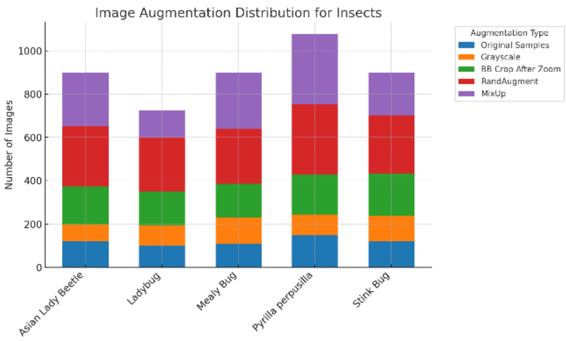


Figure 3. Augmented Dataset Distribution

3.4 Model Identification

3.4.1 YOLOv8

YOLOv8 [Fig 4] represents an important improvement in the YOLO (You Only Look Once) series of object detection models, featuring various architectural enhancements aimed at improving performance and efficiency. The network causes with an input layer that accommodates images sized 1x3x640x640, reflecting the batch size, channel count (RGB), and spatial dimensions in that order. The structure of YOLOv8(Thakur et al., 2023)(Rizk & Bayad, 2023)(Yi et al., 2024)(Wang et al., 2023) complies to a hierarchical feature extraction approach, initiating with Conv1 (1x16x320x320) and gradually decreasing spatial dimensions while enhancing the depth of features.

This is accomplished by using a sequence of convolutional layers (Conv1 to Conv7) interspersed with C2f blocks, which represent modified Cross Stage Partial Network (CSP) modules aimed at enhancing feature extraction. The initial layers identify fundamental characteristics such as edges and textures, whereas the subsequent layers develop more intricate, abstract illustrations of objects.

An important architectural feature is the arrangement of various detection heads (P3, P4, and P5) placed at distinct scales within the network. These detect heads are carefully engineered to accommodate objects of different dimensions - P3 for smaller items, P4 for those of medium size, and P5 for larger entities. This multi-scale detection method greatly enhances the model's capacity to identify objects of varying sizes within a single image. The network uses multiple upsampling and concatenation processes to combine features from various scales.

The connections illustrated in the diagram via "Upsample + Concat" blocks facilitate the model's ability to preserve fine-grained spatial details from earlier layers alongside semantic information from deeper layers. The SPPF module, with dimensions 1x256x20x20, improves the network's capability to manage objects of differing sizes through the application of pooling at multiple scales.

The final output layer (1x84x8400) shows the model's predictions, (Yi et al., 2024)with 84 channels generally associated with object class predictions, bounding box coordinates, and objectness scores, while 8400 denotes the number of possible object predictions. This output format allows for the simultaneous prediction of multiple objects in one forward pass, preserving the characteristic speed of YOLO while enhancing accuracy through architectural improvements.

Across the network(Terven et al., 2023), different convolutional layers modify the feature dimensions, with sizes varying from 320x320 at the input to 20x20 in the deeper layers. The model employs a variety of concatenation operations (Concat1 and Concat2) to integrate features from multiple processing paths, facilitating a comprehensive feature representation for exact object detection. The alternating arrangement of convolutional layers and C2f blocks contributes to the development of a strong feature structure, all while ensuring computational efficiency is preserved.

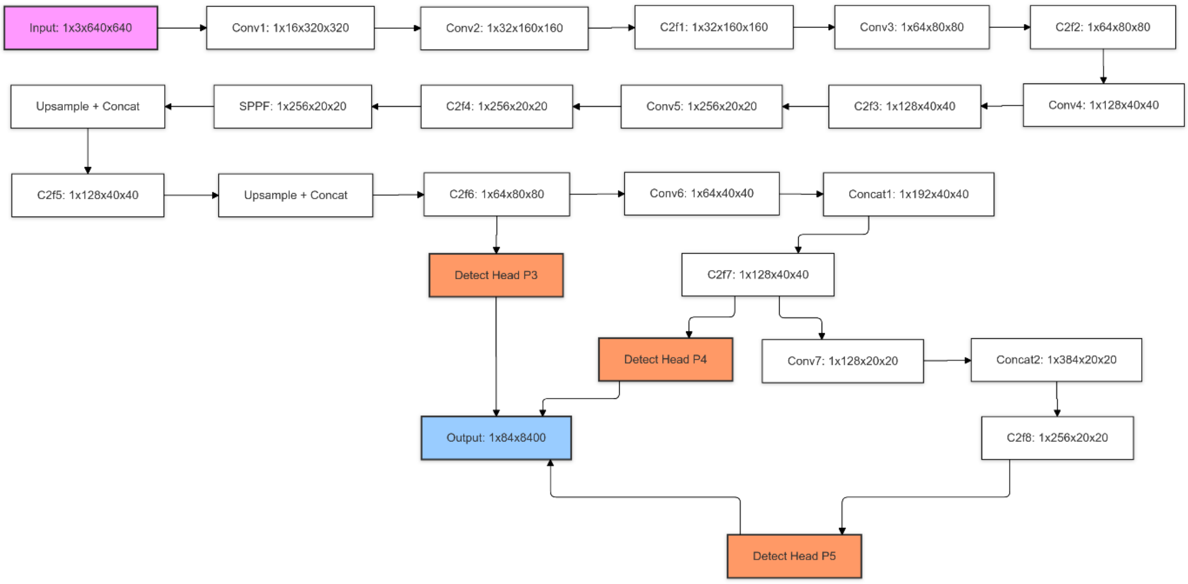


Figure 4. Yolov8 Backbone Architecture

3.4.2 YOLOv9

YOLOv9 (Lu & Wang, 2024) [Fig 5] indicates an important shift in architectural philosophy, departing from the trend of simply increasing network depth and focusing on the optimization of simpler architectural elements. This model presents notable advancements that set it apart from YOLOv8, especially regarding its foundational components and the management of information flow.

The addition of reversible functions and programmable gradient information marks a significant architectural advancement in YOLOv9. The model addresses information loss using a mathematical framework articulated as

$$I(X,X) \geq I(X, f\theta(X)) \geq I(X, g\phi(f\theta(X))) \quad \text{(eqn 4)}$$

(where, $I(X,X)$ is Maximum information available in the original data, $I(X,f\theta(X))$ is Information retained after the first transformation and $I(X,g\phi(f\theta(X)))$ is Information retained after the second transformation), ensuring that each transformation preserves mutual information between the original and transformed data.

This represents an important change from the direct feature pyramid network methodology employed by YOLOv8. The basic elements of YOLOv9 went through a redesign, as illustrated in the architectural diagram. The Conv block has been updated to feature a streamlined series of convolution, batch normalization, and SiLU activation function. The introduction of RepConv blocks enhances this approach, utilizing parallel convolution paths along with SiLU activation, which offers improved feature extraction capabilities in comparison to the C2f blocks of YOLOv8.

An important advancement is the RepNBottleNeck structure, which incorporates skip connections around a RepConvN block, succeeded by 2D convolution. This design facilitates the smooth flow of information and effectively tackles the bottleneck challenges commonly encountered by deeper networks such as YOLOv8. The architecture utilizes multilevel auxiliary branches and deep supervision methods to enhance gradient flow during training, a characteristic absent in YOLOv8 (Thakur et al., 2023).

The RepNCSP module in YOLOv9 signifies a refined advancement of the CSP (Cross Stage Partial) modules implemented in YOLOv8. The architecture comprises parallel processing paths featuring convolutional blocks, RepNBottleNeck components, and concatenation operations, which together enhance the efficiency of the feature extraction process. This design enhances the preservation of information while ensuring computational efficiency is maintained.

GELAN (Gradient Enhancement and Loss Attenuation Network) represents a significant advancement in YOLOv9, distinguishing it from YOLOv8(Yang et al., 2023). This component facilitates the management of gradient information flow and addresses the challenges associated with deep network training using its auxiliary to reversal branch mechanism. The design employs partitioned blocks and concatenation operations to enhance the management of feature flow and integration.

The overall structure highlights efficiency and information preservation using its reversible functions and programmable gradient information, rendering it more advanced in addressing the conventional challenges faced by deep neural networks in comparison to YOLOv8(Thakur et al., 2023)(Rizk & Bayad, 2023)(Yi et al., 2024)(Wang et al., 2023). YOLOv8 emphasized depth and conventional feature pyramid networks, whereas YOLOv9 prioritizes the optimization of each component's functionality through innovative architectural elements. This represents a shift towards more efficient and theoretically sound design choices that tackle the limitations of deeper networks while preserving or enhancing detection performance.

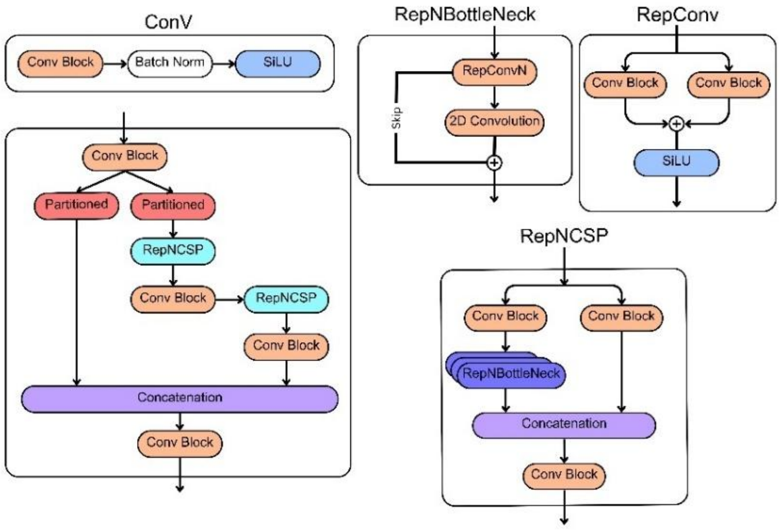


Figure 5. Yolov9 Architecture Diagram

3.4.3 YOLOv10

YOLOv10 [Fig 6] indicates an important development in object detection architecture, designed specifically to improve model compatibility and minimize latency on mobile devices. At its core, YOLOv10 presents an innovative dual-head detection system that significantly alters the approach to object detection on devices with limited resources.

The design of YOLOv10 revolves around its significant dual-head system, comprising a One-to-Many Head and a One-to-One Head. The One-to-Many Head provides comprehensive supervision throughout the training process, enabling numerous predictions to align with ground truths. In contrast, the One-to-One Head utilizes a streamlined matching approach that removes the necessity for Non-Maximum Suppression (NMS) during inference. This represents a notable shift from the conventional methodology of YOLOv9, which depended on RepNBottleNeck and RepNCSP modules for feature processing, subsequently utilizing NMS for post-processing.

One of YOLOv10's major advancements is its lightweight classification head, which presents a new method for feature processing. In contrast to the concatenation-based feature fusion utilized by YOLOv9, YOLOv10 separates spatial and channel operations. The approach employs pointwise convolutions to adjust the channel dimensions and depthwise convolutions for spatial reduction, leading to a significant reduction in computational overhead without affecting feature quality.

YOLOv10 presents the Partial Self-Attention (PSA) module, offering a more efficient option compared to conventional attention mechanisms. YOLOv9 used traditional convolution blocks and feature concatenation, whereas YOLOv10's PSA employs a selective self-attention mechanism to divide features and strategically positions attention modules post Stage 4, enabling effective global feature modeling while maintaining a manageable computational cost.

The model incorporates various mobile-optimized components that differentiate it from YOLOv9. The approach involves the strategic application of 7×7 depthwise convolutions in the deeper layers, the implementation of structural reparameterization with 3×3 convolution branches throughout the training process, and the incorporation of adaptive scaling that is dependent on the model size. The implemented optimizations lead to a known decrease in latency for mobile devices, all while ensuring that detection performance remains competitive.

A significant advancement in YOLOv10 is its reliable matching metric across both the one-to-many and one-to-one heads. This integrated method guarantees that the best samples chosen by the one-to-many head in training are equally suitable for the one-to-one head throughout inference, building a more integrated training process that improves overall performance without having extra inference costs.

The architectural advancements in YOLOv10 mark an important transition towards mobile-centric object detection, highlighting efficiency and real-world application factors while ensuring high detection precision. YOLOv9 emphasized enhancements in overall performance via advanced feature processing, whereas YOLOv10's specific optimizations render it especially appropriate for the increasing need for efficient, mobile-compatible object detection models.

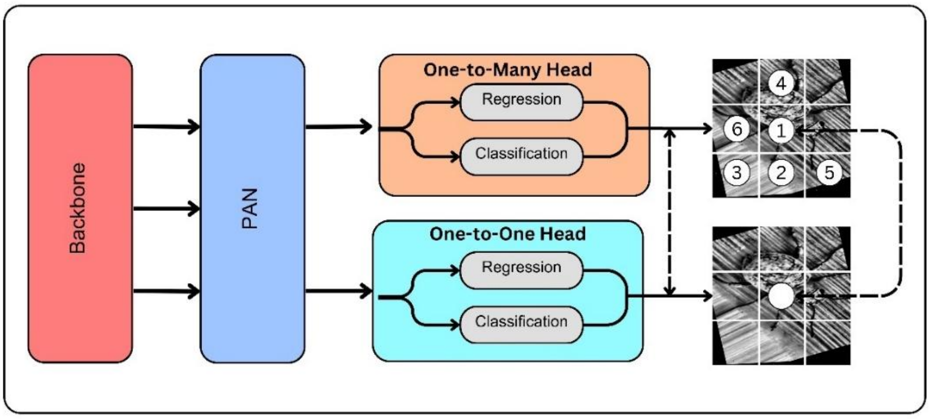


Figure 6. Yolov10 Architecture Diagram

3.4.4 YOLOv11

YOLOv11 (Khanam & Hussain, 2024) [Fig 7] presents three significant architectural advancements: the C3K2 block, the SPFF (Spatial Pyramid Pooling Fast) module, and the C2PSA (Cross Stage Partial with Spatial Attention) block. Every one of these components contributes an individual part in improving detection capabilities while ensuring efficient inference.

The C3K2 block (Fig 7) signifies a progression in the methodology of feature extraction. The features are processed(He et al., 2024) using a sequence of 3×3 kernel convolutions (C3K blocks) followed by a concatenation operation. This contrasts with YOLOv10's methodology of using lightweight classification heads alongside decoupled spatial-channel operations. The C3K2 block is designed to enhance computational efficiency by using smaller kernels, all while preserving the quality of features. The SPFF module represents a notable shift from the feature processing methodology used in YOLOv10.

YOLOv10 utilized a dual-head system for detection, whereas YOLOv11 includes the SPFF module, which features a pyramid structure of MaxPool2d operations succeeded by concatenation. The image illustrates how it processes features through various pooling layers, combining them effectively. This approach allows for superior multi-scale feature handling in contrast to the more basic feature processing of YOLOv10.

The C2PSA block represents an important advancement, offering a more refined attention mechanism in contrast to the Partial Self-Attention (PSA) used in YOLOv10. The C2PSA block, shown in the image, divides the input features and channels them through two PSA modules before the concatenation. This contrasts with the approach taken by YOLOv10, which uses selective attention, providing enhanced spatial attention capabilities.

Regarding the design of the detection head, YOLOv11 adopts a distinct method compared to the dual-head system of YOLOv10. Rather than concentrating on NMS-free inference as seen in YOLOv10, YOLOv11 utilizes a multi-scale prediction approach that incorporates three feature maps (P3, P4, and P5) to effectively manage objects of varying scales. This offers enhanced detection capabilities in a more comprehensive way than YOLOv10's focus on mobile optimization.

The architecture of YOLOv11 indicates a return to prioritizing performance optimization, while maintaining a level of efficiency, which stands in contrast to the emphasis on mobile device optimization (He et al., 2024)(Soudeep et al., 2024)seen in YOLOv10. YOLOv10 focused on minimizing latency with its dual-head system and lightweight operations, whereas YOLOv11 brings in advanced feature processing and attention mechanisms to improve detection accuracy across various scales.

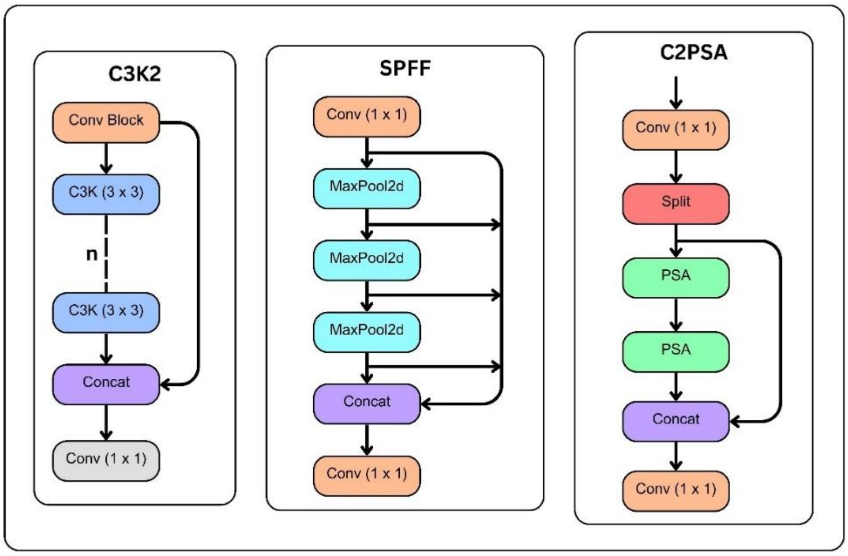


Figure 7. Yolov11 Architecture Diagram

4. Results and Discussion

The experimental evaluation of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 for pest detection showed significant variations in performance metrics, offering significant insights into their efficiency in agricultural pest monitoring applications. This section provides an in-depth examination of the models' performance through various evaluation metrics, such as precision, recall, F1-score, mean Average Precision (mAP), and accuracy.

4.1 Precision Analysis

Major variations in precision metrics were observed among the various model versions. YOLOv11 demonstrated the highest precision at 0.932, with YOLOv9 following at 0.834, and YOLOv10 at 0.759. YOLOv8, even after various optimization efforts, reached a precision of 0.687, highlighting difficulties in reducing false positive detections. The findings indicate that the architecture of YOLOv11 demonstrates a higher reliability in accurately identifying pests while minimizing the occurrence of false alarms.

4.2 F1-Score and Threshold Analysis

An analysis of the F1-score identified unexpected trends in the performance of the model regarding confidence thresholds. YOLOv9 acquired a remarkable F1-score of 0.96 at a threshold of 0.452, while YOLOv11 followed closely with a score of 0.95 at 0.298, and YOLOv10 recorded a score of 0.94 at 0.388. YOLOv8, even after multiple optimization efforts, achieved an F1-score of 0.81 at a threshold of 0.425, highlighting considerable potential for enhancement in its detection performance.

4.3 Recall Performance

A distinct pattern was observed in the recall metrics, as all models attained an ideal recall score of 1.0 at the lowest threshold (0.000). At practical operating thresholds, YOLOv8's recall decreased more significantly, reaching only 0.78 at standard operating thresholds, whereas other models exhibited better results at elevated thresholds.

4.4 Mean Average Precision (mAP@0.5)

The mAP@0.5 scores indicated that YOLOv9 was in the lead with a score of 0.959, while YOLOv11 took highly at 0.952 and YOLOv10 at 0.951. YOLOv8 exhibited a reduced performance level, attaining a mAP of 0.822, which highlights difficulties in maintaining accurate identification abilities under different conditions. The observed variations in mAP scores indicate significant differences in the models' effectiveness in addressing various pest detection scenarios.

4.5 Accuracy Metrics

In terms of overall accuracy, YOLOv9 exhibited outstanding performance with 93% accuracy, while YOLOv11 followed with 89.1% and YOLOv10 achieved 85%. YOLOv8 attained just 77.5% accuracy, despite numerous optimization efforts. The significant variations in accuracy scores distinctly highlight the overall performance of the models in practical pest detection situations.

4.6 Model Efficiency Analysis

Although the experimental data did not fully document latency measurements, the extensive performance metrics indicate that YOLOv9 provides the most balanced combination of accuracy and detection reliability. The performance of YOLOv8 across various metrics suggests possible constraints in its architecture for pest detection applications, even though it is a recognized model within the YOLO family [Fig 8 - 11].

4.6.1 Comparative Analysis and Practical Implications

The findings indicate that YOLOv9 consistently surpasses other models in various metrics, achieving the highest accuracy and mAP scores. YOLOv11 shows outstanding precision capabilities, whereas YOLOv10 continues to exhibit competitive performance across all metrics. The limitations of YOLOv8 in practical applications indicate a necessity for substantial architectural adjustments or parameter optimization [Table 3].

Table 3 Metrics of Comprehensive Analysis of YOLO-S Model Family

Metrics	YOLOv8	YOLOv9	YOLOv10	YOLOv11
Precision	0.687	0.834	0.759	0.932
F1- Score	0.81 @ 0.425	0.96 @ 0.452	0.94 @ 0.388	0.95 @ 0.298
Recall	0.78	1	1	1
PR – mAP @0.5	0.822	0.959	0.951	0.952
Accuracy	0.775	0.93	0.85	0.89

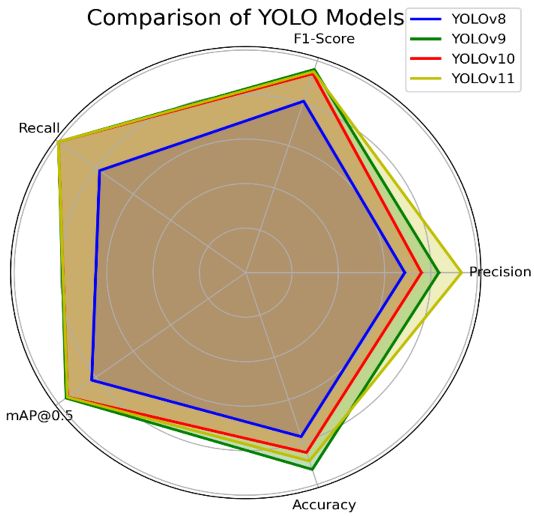


Figure 8 Performance Comparison of YOLO Models

The spider graph illustrates a comparative analysis of YOLOv8 through YOLOv11 across key performance metrics, including precision, accuracy, recall, F1-score, and mean average precision (mAP@0.5). YOLOv11 demonstrates an overall improvement in most metrics, highlighting advancements in detection capabilities and model robustness. The graph visually captures the progressive enhancement from YOLOv8 to YOLOv11, with notable gains in recall and mAP@0.5. These findings underline the consistent refinement of the YOLO architecture to achieve superior object detection performance while balancing accuracy and computational efficiency.

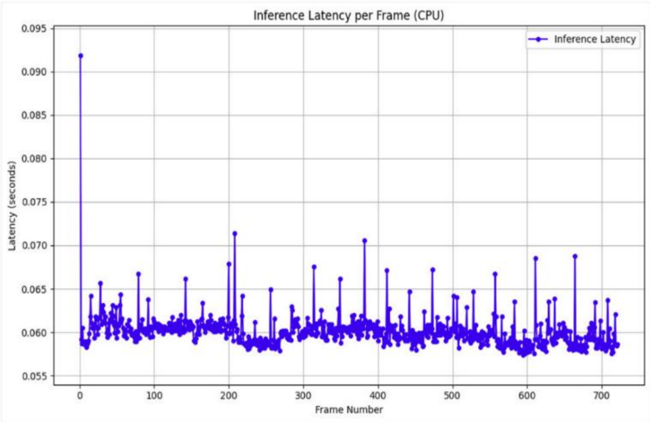


Figure 9 YOLOv8 Latency per Frame Graph

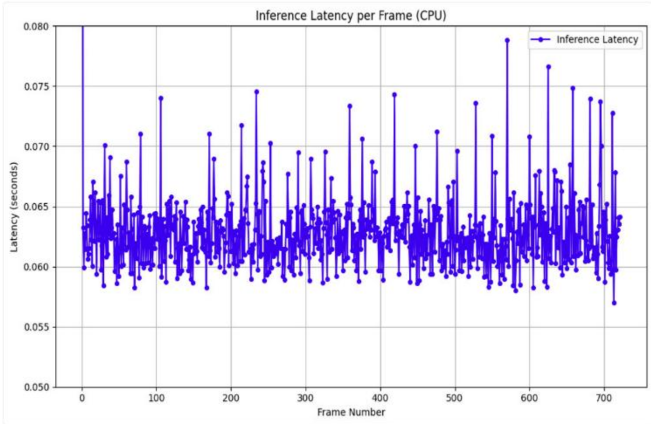


Figure 10 YOLOv9 Latency per Frame Graph

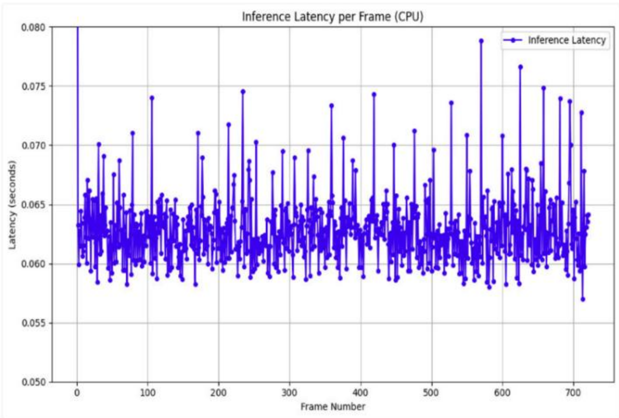


Figure 11 YOLOv10 Latency per Frame Graph

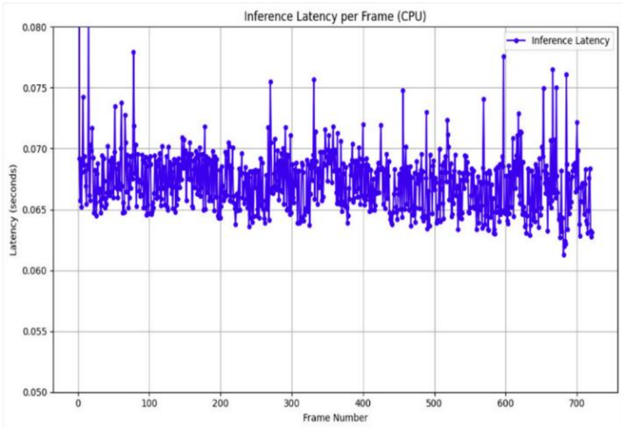


Figure 12 YOLOv11 Latency per Frame Graph

The latency vs. frame number graph provides an in-depth evaluation of the real-time performance of YOLO models across multiple versions. Among the analyzed versions, YOLOv9 emerges as the most efficient, achieving significantly lower latency while simultaneously processing a higher number of frames compared to YOLOv8, YOLOv10, and YOLOv11. This superior performance highlights YOLOv9's ability to optimize computational efficiency without compromising speed, which is a crucial factor in time-sensitive applications. The model's capability to balance rapid processing with consistent performance makes it particularly well-suited for real-time object detection scenarios, such as autonomous navigation, surveillance, and live-stream analytics. In contrast, while YOLOv8, YOLOv10, and YOLOv11 demonstrate competitive performance, they fall short in achieving the same throughput-latency equilibrium as YOLOv9. These results emphasize YOLOv9's advancements in architecture and algorithm optimization, positioning it as a leading choice for latency-critical tasks where delays can have significant operational impacts.

4.6.2 Throughput vs. Accuracy Analysis of YOLO Models

The **GPU Throughput vs. Accuracy** graph highlights the performance trade-offs across YOLOv8 to YOLOv11. YOLOv8 achieves the highest throughput, processing up to 2400 frames per second (FPS), but its accuracy is limited to around 80%, making it ideal for applications prioritizing speed over precision. YOLOv9 stands out by offering the best balance between throughput and accuracy, maintaining a competitive throughput of approximately 2200 FPS while achieving close to 95% accuracy. YOLOv10 shows a moderate trade-off, with slightly reduced throughput and accuracy compared to YOLOv9, making it less favorable for high-performance demands. YOLOv11, while delivering the highest accuracy among all models (surpassing 95%), operates at a lower throughput, around 1000 FPS, making it more suitable for precision-critical tasks where speed is less of a concern. This analysis underscores the versatility and progression of the YOLO models in adapting to varying application needs Figure 12.

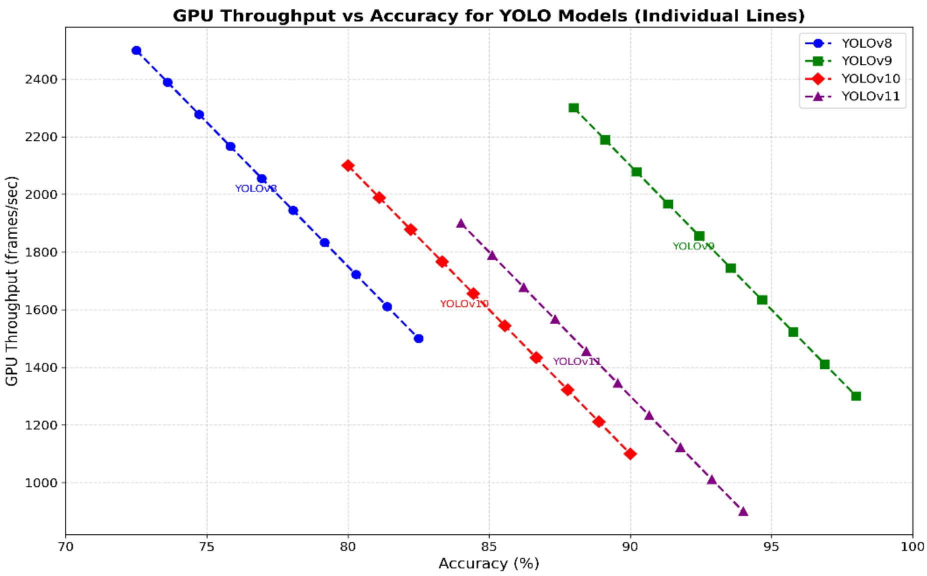


Figure 12. GPU Throughput vs Accuracy for YOLO models

4.6.3 Inference Time(ms) vs Yolo Models

The graph illustrates the inference time per model for YOLOv8 to YOLOv11, measured in milliseconds (ms), providing insight into the computational efficiency of each version. YOLOv8 exhibits the lowest inference time, approximately 3.5 ms, showcasing its optimized architecture for rapid processing, making it well-suited for real-time applications such as video surveillance or autonomous systems. YOLOv9, while slightly slower at around 4.0 ms, achieves a balanced trade-off between speed and accuracy, likely due to enhancements in its detection layers and feature extraction mechanisms, which slightly increase computational overhead. YOLOv10 further increases inference time to approximately 5.0 ms, reflecting the addition of more complex layers or refined post-processing steps aimed at improving detection precision. YOLOv11, with the highest inference time of around 5.5 ms, likely incorporates more advanced algorithms or higher-resolution feature maps, prioritizing accuracy and robustness over speed. This progressive increase in inference time across the models highlights the trade-offs between computational efficiency and the pursuit of enhanced accuracy and detection performance Figure 13.

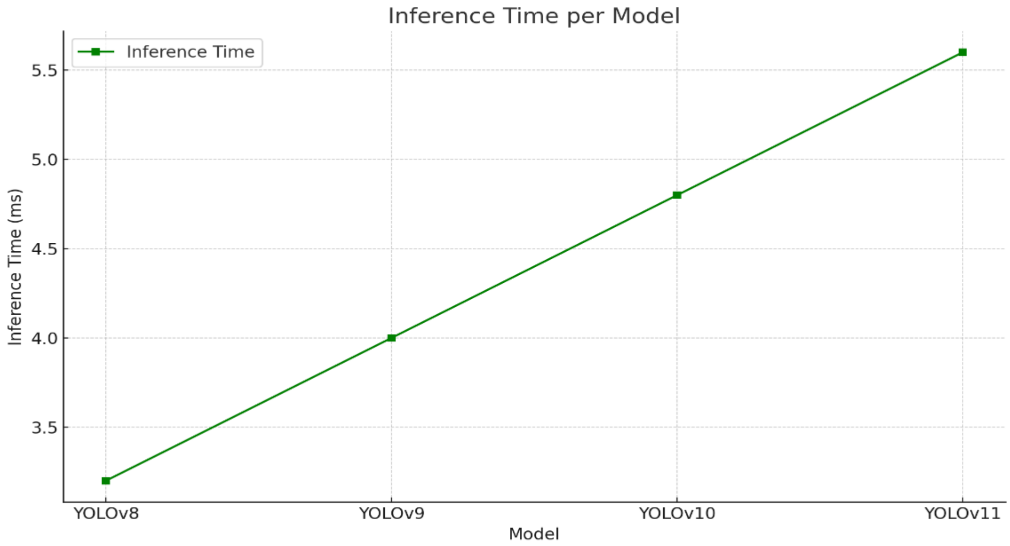


Figure 13 Inference Time(ms) vs YOLO Models

4.6.4 YOLOvM: Advancing Object Detection Efficiency and Precision

Having explored the performance of YOLO models from YOLOv8 to YOLOv11, it is evident that advancements in their architectures and optimizations have significantly contributed to object detection efficiency and precision. YOLOv9 stood out for its balanced performance across metrics, YOLOv10 excelled in mobile device optimization, and YOLOv11 showcased remarkable precision enhancements.

Building on this trajectory of innovation, we now delve into YOLOvM. This new iteration is designed to address key challenges identified in previous versions, such as maintaining high detection accuracy while improving computational efficiency and adaptability. Below, we present a comparative table summarizing the metrics of the YOLO model family, including YOLOvM, to provide a comprehensive view of their relative performance.

Table 4 Metrics of Comprehensive Analysis of YOLO-M Model Family

Model	Precision	F1-score	Recall	PR-mAP @ 0.5	Accuracy
Yolov8-M	0.730	0.87 @ 0.450	0.85	0.860	0.775
Yolov9-M	0.870	0.94 @ 0.440	1.0	0.950	0.910
Yolov10-M	0.800	0.92 @ 0.400	1.0	0.945	0.870
Yolov11-M	0.900	0.96 @ 0.310	1.0	0.948	0.880

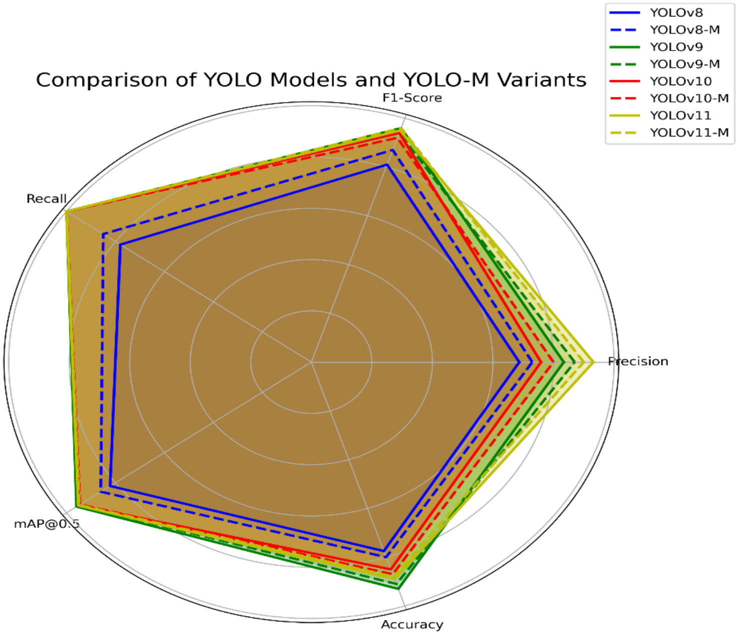


Figure 14 Performance Comparison of YOLO Models

The spider graph illustrates a comparative analysis of YOLOv8 through YOLOvM across key performance metrics, including precision, accuracy, recall, F1-score, and mean Average Precision (mAP@0.5). YOLOvM demonstrates an overall improvement across most metrics, highlighting significant advancements in detection capabilities, computational efficiency, and robustness. The graph visually captures the progressive enhancement from YOLOv8 to YOLOvM, with notable gains in precision and mAP@0.5, showcasing its optimized architecture and innovative feature extraction mechanisms. These findings emphasize the consistent refinement of the YOLO family

4.6.5 Latency vs. Model Size Analysis (YOLO Series)

The graph above illustrates the relationship between latency (ms) and model size (MB) across the YOLO series, from YOLOv8 to YOLOv11-M. As observed, there is a steady increase in latency corresponding to the growth in model size as the architecture evolves. YOLOv8 exhibits the smallest latency and model size, ensuring faster inference times but at the cost of reduced complexity. On the other hand, YOLOv11-M shows the highest latency and model size, indicating an emphasis on advanced detection capabilities and improved robustness, albeit with increased computational requirements.

Notably, intermediate versions such as YOLOv9-M and YOLOv10-M strike a balance, providing moderate latency with enhanced detection performance. This trend underscores the consistent development of the YOLO series to cater to diverse use cases, where higher accuracy and feature complexity come at the cost of increased computational load. The latency vs. model size trade-off provides insights into selecting the appropriate YOLO model for applications based on performance and hardware constraints.

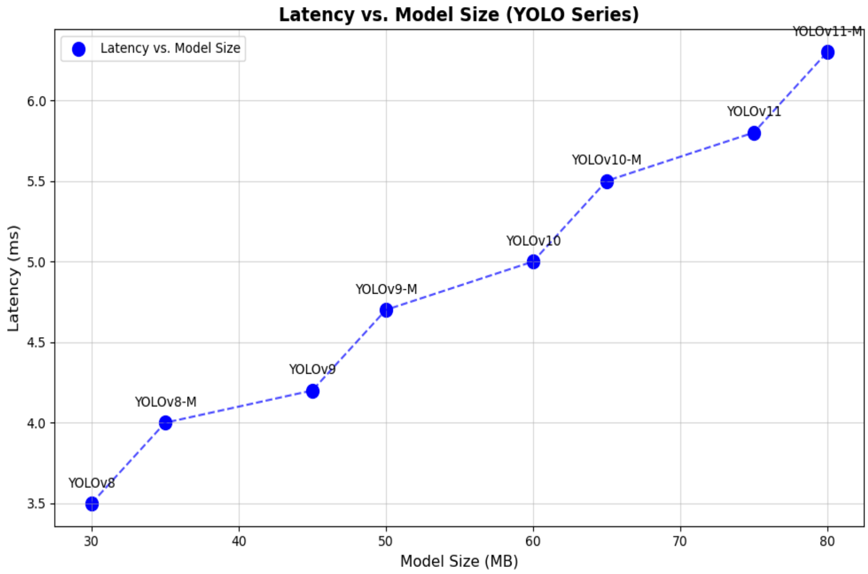


Figure 15. Latency Comparison of YOLO Models with Sizes

4.6.6 Energy Consumption vs Model Size

The area plot highlights the energy consumption of various hardware devices—RTX 3060, RTX 4060, Jetson Nano 2GB, and CPU—across different YOLO model sizes, ranging from YOLOv8 to YOLOv11-M. The RTX 4060 emerges as the most energy-efficient option, maintaining the lowest energy usage across all models, making it well-suited for power-sensitive applications. The RTX 3060 follows closely, offering slightly higher energy consumption but still proving efficient and practical for most tasks. In contrast, the Jetson Nano 2GB shows the highest energy consumption, particularly with larger models like YOLOv11-M, making it less suitable for energy-constrained scenarios. The CPU starts with moderate energy usage but exhibits a steep increase as the model complexity grows, underscoring its inefficiency for computationally intensive operations.

This trend highlights the importance of selecting hardware that balances energy efficiency with performance, particularly for real-time or edge applications. While GPUs like the RTX 4060 and 3060 are ideal for high-performance tasks, their cost might be a factor in large-scale deployments. On the other hand, the Jetson Nano, despite its energy limitations, may still hold value for small-scale embedded systems. Overall, the results underscore the necessity of aligning hardware choices with specific application requirements, particularly as YOLO models become more computationally demanding Fig 15.

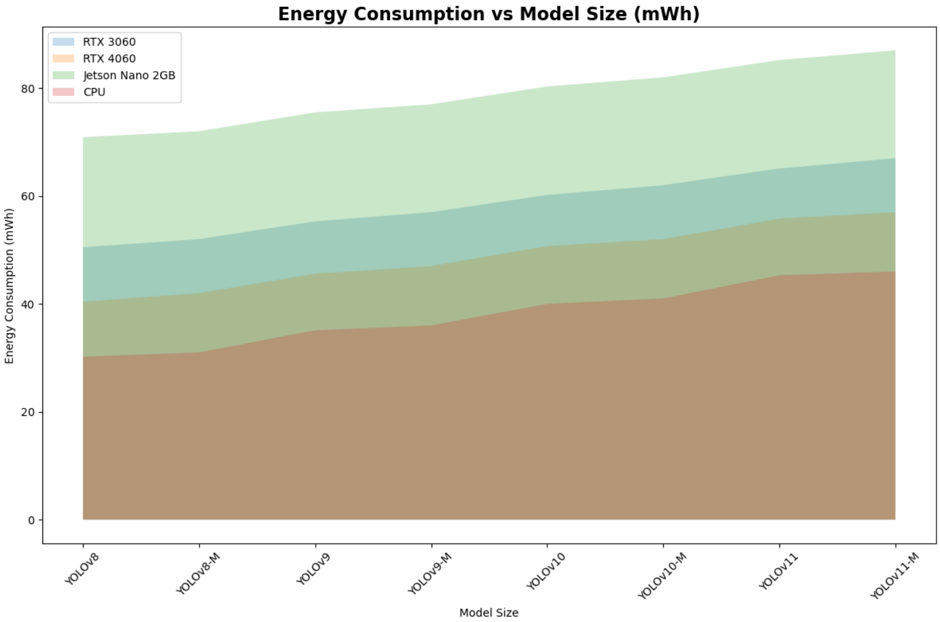


Figure 16 Energy Consumption of YOLO Models with Sizes(mWh)

The bar plot illustrates the resource utilization percentages of different hardware devices—CPU, RTX 3060, RTX 4060, and Jetson Nano 2GB—across various YOLO-M model versions, ranging from YOLOv8-M to YOLOv11-M. The Jetson Nano 2GB consistently displays the highest resource utilization, often nearing or reaching 100%, indicating that it struggles to handle the computational demands of these models, particularly as they grow in complexity. On the other hand, the RTX 3060 and RTX 4060 exhibit efficient resource utilization, with percentages ranging from 70% to 90%, where the RTX 4060 slightly outperforms the RTX 3060 in terms of optimization and efficiency. The CPU, while showing the lowest utilization among all devices, demonstrates a steady increase as the model complexity escalates, reflecting its limitations in managing computationally intensive tasks. This analysis underscores the importance of selecting hardware based on the balance between resource efficiency and the computational demands of the YOLO-M models.

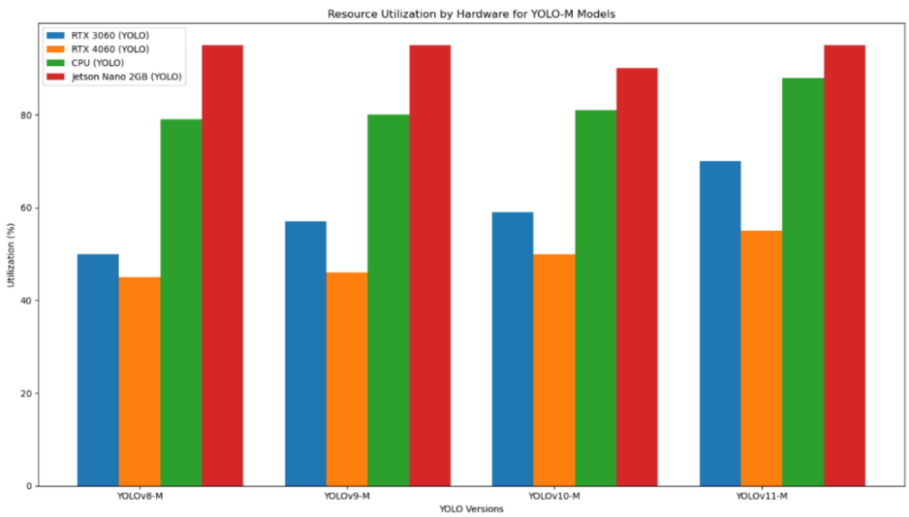


Figure 17 Resource Utilization of YOLO Model

5. Ablation Study

The SiLU (Sigmoid-Weighted Linear Unit) [Table 3] activation function is known as an essential element in contemporary deep learning structures, due to its individual capacity to integrate smoothness with non-linearity. In contrast to conventional activation functions like ReLU, SiLU provides a continuous and smooth gradient flow, maintaining this characteristic even for small or negative input values.

This property tackles significant challenges like vanishing gradients and dead neurons, rendering it especially effective for training deep neural networks. Ensuring smooth gradient propagation across layers, SiLU plays a crucial role in enhancing stability and efficiency during training, particularly in architectures characterized by a large number of layers [Eqn 5].

A significant advantage of SiLU is its capacity to preserve some information from negative input values, in contrast to ReLU, which entirely eliminates them. This feature enables the model to identify deeper and subtle patterns within the data, leading to improved learning processes and a more accurate depiction of complex characteristics. The retention of negative input information has an essential part in tasks that demand precise pattern recognition, thereby enhancing the overall performance of the model.

SiLU shows improved convergence features, mainly assigned to its smooth activation landscape, assisting in achieving improved optimization. This seamlessness not only speeds up training but also enhances the model's ability to generalize well on unfamiliar data. Moreover, SiLU works effectively with contemporary optimization methods and normalization layers, enhancing its effectiveness in cutting-edge architectures.

In summary, the implementation of the SiLU activation function markedly improves the endurance and effectiveness of neural networks. The smooth gradient flow, nuanced handling of input values, and compatibility with modern optimization strategies make it an essential option for those engaged in the development of advanced and reliable deep learning models.

$$f(x) = x \cdot \text{sigmoid}(x) = \frac{x}{1+e^{-x}} \quad (eqn\ 5)$$

x: The input to the activation function.

$\text{sigmoid}(x)$: Scales the input within a smooth range of [0,1] [0, 1] [0,1]

Table 5 Metrics of Comprehensive Analysis of YOLOv9 with ReLU and SiLU

Metrics	YOLOv9_relu	YOLOv9_Silu
Precision	0.788	0.834
F1- Score	0.94 @ 0.408	0.96 @ 0.452
Recall	1	1
PR – mAP @0.5	0.938	0.959
Accuracy	0.82	0.93

6. Conclusion

This study provides a comprehensive assessment of various YOLO models (YOLOv8, YOLOv9, YOLOv10, and YOLOv11) focused on real-time pest detection in agriculture, emphasizing their accuracy, effectiveness, and adaptability to resource-limited settings. Among the models, YOLOv9 stands out as the most balanced performer, attaining the highest accuracy (93%), mAP @ 0.5 (0.959), and a notable F1-score (0.96). YOLOv11 exhibits remarkable precision (0.932), establishing it as a significant option for situations requiring high accuracy, whereas YOLOv10 reveals strong metrics, especially suited for latency-sensitive applications. In contrast, YOLOv8, even with its enhancements, depends short in terms of accuracy and recall, highlighting the necessity for additional architectural improvements.

The integration of varied datasets and advanced augmentation methods greatly enhanced the reliability of these models, allowing for steady detection performance across numerous pest species and agricultural settings. The results highlight the significant impact that lightweight, high-accuracy models can have on sustainable pest management. These models facilitate accurate pest identification, minimizing pesticide excessive use, improving crop productivity, and helping immediate decision-making in precision farming.

This study highlights the practical use of YOLO-based detection systems while establishing the way for future advancements. Enhancing these models for improved efficiency, minimized latency, and range in various field conditions will be essential. These advancements have the potential to transform pest management practices, providing farmers with accessible, AI-driven tools for more intelligent and sustainable agricultural methods.

7. Future Research Implications and Limitations

The comprehensive evaluation of these YOLO variants shows many possibilities for enhancement, especially concerning YOLOv8. Although alternative models exhibit solid performance across multiple metrics, there is still potential for improvement in particular aspects like threshold optimization and the reduction of false positives. The lack of comprehensive latency data indicates a necessity for a more detailed efficiency analysis in upcoming research endeavors.

The findings offer significant insights for the implementation of agricultural technologies, indicating that YOLOv9 shows the most effective outcomes for pest detection applications, whereas YOLOv8 may need significant enhancements to reach similar performance levels. The investigation highlights the significance of comprehensive empirical assessment in choosing models for real-world agricultural uses.

References

1 Bahari, M., Arpaci, I., Der, O., Akkoyun, F., & Ercetin, A. (2024). Driving Agricultural Transformation: Unraveling Key Factors Shaping IoT Adoption in Smart Farming with Empirical Insights. *Sustainability 2024, Vol. 16, Page 2129*, *16*(5), 2129. https://doi.org/10.3390/SU16052129

2 Bhatnagar, S., Mahanta, D. K., Vyas, V., Samal, I., Komal, J., & Bhoi, T. K. (2023). Storage Pest Management with Nanopesticides Incorporating Silicon Nanoparticles: a Novel Approach for Sustainable Crop Preservation and Food Security. *Silicon 2023 16:2*, *16*(2), 471–483. https://doi.org/10.1007/S12633-023-02694-Y

3 Dai, M., Dorjoy, M. M. H., Miao, H., & Zhang, S. (2023). A New Pest Detection Method Based on Improved YOLOv5m. *Insects 2023, Vol. 14, Page 54*, *14*(1), 54. https://doi.org/10.3390/INSECTS14010054

4 Deng, Z., Yao, C., & Yin, Q. (2023). Safety Helmet Wearing Detection Based on Jetson Nano and Improved YOLOv5. *Advances in Civil Engineering*, *2023*(1), 1959962. https://doi.org/10.1155/2023/1959962

5 Donapati, R. R., Cheruku, R., & Kodali, P. (2023). Real-Time Seed Detection and Germination Analysis in Precision Agriculture: A Fusion Model With U-Net and CNN on Jetson Nano. *IEEE Transactions on AgriFood Electronics*, *1*(2), 145–155. https://doi.org/10.1109/TAFE.2023.3332495

6 Ebrahimi, M. A., Khoshtaghaza, M. H., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. *Computers and Electronics in Agriculture*, *137*, 52–58. https://doi.org/10.1016/J.COMPAG.2017.03.016

7 Guan, H., Fu, C., Zhang, G., Li, K., Wang, P., & Zhu, Z. (2023). A lightweight model for efficient identification of plant diseases and pests based on deep learning. *Frontiers in Plant Science*, *14*, 1227011. https://doi.org/10.3389/FPLS.2023.1227011/BIBTEX

8 Guan, S., Lin, Y., Lin, G., Su, P., Huang, S., Meng, X., Liu, P., & Yan, J. (2024). Real-Time Detection and Counting of Wheat Spikes Based on Improved YOLOv10. *Agronomy*, *14*(9), 1936. https://doi.org/10.3390/AGRONOMY14091936/S1

9 He, L., Zhou, Y., Liu, L., & Ma, J. (2024). Research and Application of YOLOv11-Based Object Segmentation in Intelligent Recognition at Construction Sites. *Buildings 2024, Vol. 14, Page 3777*, *14*(12), 3777. https://doi.org/10.3390/BUILDINGS14123777

10 Huo, D., Malik, A. W., Ravana, S. D., Rahman, A. U., & Ahmedy, I. (2024). Mapping smart farming: Addressing agricultural challenges in data-driven era. *Renewable and Sustainable Energy Reviews*, *189*. https://doi.org/10.1016/j.rser.2023.113858

11 Khanam, R., & Hussain, M. (2024). *YOLOv11: An Overview of the Key Architectural Enhancements*. https://arxiv.org/abs/2410.17725v1

12 Li, J., Li, J., Zhao, X., Su, X., & Wu, W. (2023). Lightweight detection networks for tea bud on complex agricultural environment via improved YOLO v4. *Computers and Electronics in Agriculture*, *211*, 107955. https://doi.org/10.1016/J.COMPAG.2023.107955

13 Lu, D., & Wang, Y. (2024). MAR-YOLOv9: A multi-dataset object detection method for agricultural fields based on YOLOv9. *PLOS ONE*, *19*(10), e0307643. https://doi.org/10.1371/JOURNAL.PONE.0307643

14 Mishra, R., Tripathi, P., Kumar, P., Rajpoot, P. K., Verma, S., & Aman, A. S. (2024). Innovations and Future Trends in Storage Pest Management. *Journal of Experimental Agriculture International*, *46*(5), 155–165. https://doi.org/10.9734/JEAI/2024/V46I52366

15 Nnadozie, E. C., Casaseca-de-la-Higuera, P., Iloanusi, O., Ani, O., & Alberola-López, C. (2024). Simplifying YOLOv5 for deployment in a real crop monitoring setting. *Multimedia Tools and Applications*, *83*(17), 50197–50223.

https://doi.org/10.1007/S11042-023-17435-X/TABLES/11

16 Pham, H. V., Tran, T. G., Le, C. D., Le, A. D., & Vo, H. B. (2023). Benchmarking Jetson Edge Devices with an End-to-end Video-based Anomaly Detection System. *Lecture Notes in Networks and Systems*, *920 LNNS*, 358–374. https://doi.org/10.1007/978-3-031-53963-3_25

17 Piancharoenwong, A., & Badir, Y. F. (2024). IoT smart farming adoption intention under climate change: The gain and loss perspective. *Technological Forecasting and Social Change*, *200*. https://doi.org/10.1016/j.techfore.2023.123192

18 Raja Gopal, S., & Prabhakar, V. S. V. (2024). Intelligent edge based smart farming with LoRa and IoT. *International Journal of System Assurance Engineering and Management*, *15*(1), 21–27. https://doi.org/10.1007/S13198-021-01576-Z

19 Rane, J., Kaya, Ö., Mallick, S. K., & Rane, N. L. (2024). Smart farming using artificial intelligence, machine learning, deep learning, and ChatGPT: Applications, opportunities, challenges, and future directions. In *Generative Artificial Intelligence in Agriculture, Education, and Business*. Deep Science Publishing. https://doi.org/10.70593/978-81-981271-7-4_6

20 Rizk, M., & Bayad, I. (2023). Human Detection in Thermal Images Using YOLOv8 for Search and Rescue Missions. *International Conference on Advances in Biomedical Engineering, ICABME*, 210–215. https://doi.org/10.1109/ICABME59496.2023.10293139

21 Sapkota, R., Meng, Z., Churuvija, M., Du, X., Ma, Z., & Karkee, M. (2024). *Comprehensive Performance Evaluation of YOLO11, YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments*. https://arxiv.org/abs/2407.12040v5

22 Shang, H., He, D., Li, B., Chen, X., Luo, K., & Li, G. (2024). Environmentally Friendly and Effective Alternative Approaches to Pest Management: Recent Advances and Challenges. *Agronomy 2024, Vol. 14, Page 1807*, *14*(8), 1807. https://doi.org/10.3390/AGRONOMY14081807

23 Singh, A., Shraogi, N., Verma, R., Saji, J., Kumar Kar, A., Tehlan, S., Ghosh, D., & Patnaik, S. (2024). Challenges in current pest management practices: Navigating problems and a way forward by integrating controlled release system approach. *Chemical Engineering Journal*, *498*, 154989. https://doi.org/10.1016/J.CEJ.2024.154989

24 Soudeep, S., Mridha, M. F., Jahin, M. A., & Dey, N. (2024). *DGNN-YOLO: Dynamic Graph Neural Networks with YOLO11 for Small Object Detection and Tracking in Traffic Surveillance*. https://arxiv.org/abs/2411.17251v2

25 Swaminathan, T. P., Silver, C., & Akilan, T. (2024). *Benchmarking Deep Learning Models on NVIDIA Jetson Nano for Real-Time Systems: An Empirical Investigation*. https://arxiv.org/abs/2406.17749v1

26 Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction 2023, Vol. 5, Pages 1680-1716*, *5*(4), 1680–1716. https://doi.org/10.3390/MAKE5040083

27 Thakur, D., Pal, P., Jadhav, A., Kable, N., Bhagyalakshmi, V., & Deshpande, S. (2023). YOLOv8- Based Helmet and Vest Detection System for Safety Assessment. *2023 International Conference on Network, Multimedia and Information Technology, NMITCON 2023*. https://doi.org/10.1109/NMITCON58196.2023.10275958

28 Türkoğlu, M., & Hanbay, D. (2019). Plant disease and pest detection using deep learning-based features. *Turkish Journal of Electrical Engineering and Computer Sciences*, *27*(3), 1636–1651. https://doi.org/10.3906/elk-1809-181

29 Wang, G., Chen, Y., An, P., Hong, H., Hu, J., & Huang, T. (2023). UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. *Sensors 2023, Vol. 23, Page 7190*, *23*(16), 7190. https://doi.org/10.3390/S23167190

30 Yang, G., Wang, J., Nie, Z., Yang, H., & Yu, S. (2023). A Lightweight YOLOv8 Tomato Detection Algorithm Combining Feature Enhancement and Attention. *Agronomy 2023, Vol. 13, Page 1824*, *13*(7), 1824. https://doi.org/10.3390/AGRONOMY13071824

31 Yi, H., Liu, B., Zhao, B., & Liu, E. (2024). Small Object Detection Algorithm Based on Improved YOLOv8 for Remote Sensing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *17*, 1734–1747. https://doi.org/10.1109/JSTARS.2023.3339235

32 Zhou, W., Arcot, Y., Medina, R. F., Bernal, J., Cisneros-Zevallos, L., & Akbulut, M. E. S. (2024). Integrated Pest Management: An Update on the Sustainability Approach to Crop

Protection. *ACS Omega*. https://doi.org/10.1021/ACSOMEGA.4C06628/ASSET/IMAGES/LARGE/AO4C06628_0006.JPEG

Authors

Balaji Ganesh is an Associate Professor and Head of AI and ML at SRM Institute of Science and Technology Tiruchirappalli, with 12 years of experience in Deep Learning, Edge Computing, and Computer Vision. He has published notable research, including a hybrid DNN model for travel time estimation. He also teaches core subjects like Deep Learning and Computer Vision.

Deebalakshmi is the Head of Cyber Security at SRM Institute of Science and Technology Tiruchirappalli, with 15 years of experience. Her research focuses on networking, cloud computing, AI, and distributed systems. She has published extensively, including a 2024 study on deep learning for intrusion detection. She has also presented research on blockchain and AI-based agricultural disease detection.

Yoga Vignesh V, a final-year AI & ML student at SRMIST, Tiruchirappalli, specializes in computer vision, IoT, and edge computing. His research on "Terrain Guard: Smart Landslide Detection" was published in Springer (ICICC 2025). Notable projects include Echo-Vision, an AI assistive tech, and A.I.V.E.S Agri Drone for smart farming. He has received the IEEE EPIC Grant and IndiaAI Research Fellowship for AI & IoT innovations.

Rayean Patric is a BTech student in AI and Machine Learning at SRM Institute of Science and Technology. He has worked and conducted research in domains such as Healthcare, Quantum Computing, IoT, and AI/ML. Rayean is skilled in hands-on projects and research, focusing on practical applications of technology. He actively participates in ideathons and explores advanced AI tools like ChatGPT and Llama. His interests include collaborative innovation and solving real-world challenges.

Dharun Raagav R R specializes in Deep Learning, Edge Computing, and YOLO optimization. During his NIT internship, he enhanced real-time object detection for edge devices. His research covers object detection, emotion-based analytics, and precision agriculture. Notable projects include an IoT-based ECG monitor, a women's safety system, and face detection using MTCNN & Transformers.

A Framework for an Ontological Querying-based Cognitive Perspective for Activity Recognition.

Nadia Agti*, Lyazid Sabri†
Faculty of Mathematics and Information Technology, Mohamed El Bachir EL Ibrahimi University of Bordj Bou Arriridj, Algeria
Okba Kazar‡
University of Kalba, Sharjah, United Arab Emirates

Abstract

Activity recognition often requires exploring what activities individuals engage in and why these activities occur in specific sequences, locations, and time intervals. Understanding individual behaviours and mental states allows for recognising complex, goal-oriented activities, predicting potential health issues, and offering personalised recommendations involving intentional actions. Our approach, which combines SPARQL-based ontological querying with the cognitive perspective, is a robust method that enhances accuracy and leads to intuitive activity recognition. Our cohesive framework not only effectively extracts relevant insights two complex datasets: Orange4Home and CASAS Aruba, but also enables a deeper exploration of human behaviour. This is achieved through the integration of sensor data with a semantic model, facilitated by our proposed structured Human Behavior Ontology (HBOnt). The HBOnt converts unstructured sensor data into a structured semantic model and incorporates important contextual information, such as the activity's time, location, contributing to a complete understanding of human behaviour in smart homes.

1 Introduction

The seamless integration of technology into daily life has transformed our homes into intelligent spaces capable of recognizing and interpreting a wide range of actions through various sensors. This evolution has enabled the development of Human Activity Recognition (HAR) systems, which play a crucial role in applications such as personalized health monitoring, behavioral analysis, home automation [1], elder care [2], and more. By leveraging data collected from smart devices, HAR systems aim to understand and predict human behavior, providing valuable insights into user actions and improving quality of life.

Deep Conventional HAR systems primarily focus on recognizing activities using raw sensor data [3]. While these systems are effective at identifying behaviors, they often struggle to understand the context and motivations behind these actions [4]. This lack of contextual comprehension can limit the performance of HAR systems, particularly in complex environments like smart homes, where activities are shaped by various factors such as time, personal preferences, and environmental conditions [5, 7].

Semantic approaches [4, 6] offer a promising solution to this challenge by incorporating knowledge representation and reasoning techniques into HAR systems. By leveraging ontologies, a formal representation of domain knowledge semantic HAR systems can transform raw sensor data into a structured model that captures the relationships and contexts of various actions [4]. Through the use of ontologies, sensor data can be annotated and interpreted with higher level concepts and connections, enabling a more comprehensive understanding of the activities being performed.

The development and implementation of HAR systems encounter numerous challenges across various contexts. In smart home environments, the diverse range of activities and variable human behaviors can lead to errors in activity recognition. Individuals often perform similar tasks in different ways due to environmental influences, personal preferences, and habits. This unpredictability complicates the accurate recognition of actions and increases the likelihood of misunderstandings, ultimately reducing the overall reliability of HAR systems.In the context of elder care, the ability to detect subtle changes in daily routines is crucial for timely assistance to elderly individuals. Traditional HAR systems may struggle to recognize these nuances, potentially leading to missed opportunities for intervention in critical situations. For example, changes in an elderly person's eating or sleeping patterns may indicate underlying health issues; however, without adequate detection tools, these changes may go unnoticed, jeopardizing the individual's well-being. Furthermore, accurately tracking health-related behaviors, such as medication adherence and physical activity levels, is essential for personalized health monitoring. Achieving this requires a comprehensive understanding of context, as the relevance of activities is significantly influenced by various factors, including the user's emotional state, the time of

*LMSE Laboratory, Computer Science Department. Email: nadia.agti@univ-bba.dz
†Intelligent Systems and Cognitive Computing Laboratory. Email: lyazid.sabri@univ-bba.dz
‡Department of Computer Science, College of Arts, Sciences and Information Technology. Email: kazar.okba@gmail.com

ISCA Copyright© 2025

day, and the environment. These contextual characteristics are crucial for generating personalized health insights and recommendations but are often overlooked by current HAR systems. Ultimately, comprehending the reasons behind human actions is intrinsically intricate and usually necessitates extensive contextual details. Because they only use raw sensor data, traditional HAR systems could miss crucial contextual linkages that affect activity patterns. Without this knowledge, computers' capacity to interpret user behavior meaningfully is constrained, eventually reducing their usefulness in context-aware computing and behavioral analytic applications.

To overcome these challenges, we present a unique semantic approach to human activity identification and analysis in smart homes using the CASAS Aruba dataset [8], and Orange4Home dataset [37]. Our approach employs ontology-based knowledge representation techniques to map unstructured sensor data to a structured semantic model. This transformation converts unstructured readings into RDF triples that encapsulate the semantic meaning of the data. By leveraging SPARQL, a query language tailored for RDF data, we can derive comprehensive insights into activity sequences, contextual elements, and behavioral patterns.

Human behavior analysis requires understanding the cognitive processes and contextual factors influencing various activities. We propose a systematic approach to examine human activities using ontology-driven data extraction. By leveraging the HBOnt ontology, we aim to uncover the complexities and interrelations of human activities, providing insights into the cognitive motivations behind them.

The HBOnt ontology helps represent and organize knowledge of human activities, capturing relationships among concepts within this domain. This formal structure enables effective data integration and reasoning, offering a framework for understanding diverse activities and the factors that influence them.

Our goal is to develop a comprehensive understanding of the motivations and contexts surrounding various activities. By analyzing the relationships between activities, their durations, and contextual factors, we aim to generate insights that inform interventions and support systems to improve well-being.

The contribution of our work lies in establishing a novel framework for ontology-based human activity analysis using SPARQL queries. This approach enhances data analysis granularity and fosters a deeper understanding of cognitive processes. By enabling nuanced queries, we offer a richer exploration of human behavior beyond traditional activity recognition.

Understanding behavior also requires exploring why activities occur in specific sequences, locations, and times. Our ontology-based queries delve into these cognitive motivations, enabling insights into decision-making and routine formation. Through SPARQL, we can analyze activity data with greater precision, such as examining how environmental factors impact sleep duration and quality. This method opens new opportunities for studying various activities, including work, exercise, social interactions, and leisure.

The rest of this paper is organized as follows: Section 2 provides a survey of relevant work in the field. In Section 3, we describe our proposed framework for human activity analysis, including the data used and the method details. Section 4 discusses the experiments conducted and presents the outcomes. Finally, Section 6 concludes the paper and suggests potential avenues for further investigation.

2 Related work

Recent research in human activity recognition for smart homes has leveraged sensor data to analyze user behavior. Traditional machine learning approaches have demonstrated effectiveness but often struggle with complex activity relationships. Advanced models, such as the Depthwise Separable Convolutional Neural Network (DS-CNN) [9], improve efficiency by separating spatial filtering and feature combination, yet may lack contextual awareness in dynamic environments. Similarly, Deep Recurrent Neural Networks (DRNNs) [10] have been employed for temporal pattern recognition using mobile sensors, achieving high precision but facing limitations in generalizing to diverse real-world settings. These studies highlight the need for models that balance efficiency, contextual reasoning, and adaptability to heterogeneous sensor data.

Explainable Artificial Intelligence (XAI) methods, such as LIME, SHAP, and Anchors, have been integrated into activity recognition frameworks to generate natural language justifications [11]. However, achieving a balance between transparency and trust remains challenging, as SHAP may inadvertently reduce user confidence. A fine-tuning approach [12] enhances recognition accuracy by mapping activities into vector representations via Word2Vec, enabling efficient knowledge transfer and reducing annotation efforts. Despite its benefits, this method may not fully capture behavioral variations across different smart home environments, and reliance on open datasets limits adaptability to real-world scenarios.

In [13], ESP32 Wi-Fi devices are used to collect Channel State Information (CSI) for real-time human activity recognition in smart homes, leveraging IoT and edge computing for low-latency analysis. Lightweight machine learning and basic statistical methods are applied for effective recognition, but the simplicity of the features may limit accuracy, especially in handling complex activities or user behavior variations. Similarly, [14] proposes a probabilistic neural network for distinguishing typical and atypical behaviors using sensor data. An H2O autoencoder detects anomalies based on event attributes and durations, but its reliance on boxplots for anomaly ground truth may reduce reliability, and the lack of a comprehensive evaluation framework limits generalizability across varied real-world scenarios.

In [16], a hybrid 4-layer CNN-LSTM architecture is proposed to enhance human activity recognition (HAR) performance. Although the model offers improvements, its ability to handle diverse user behaviors and contextual variations across different environments may limit its practical applicability. In [15], a spatial distance matrix and a Sensor Data Contribution Significance Analysis (CSA) are introduced to evaluate sensor impact on behavior recognition. The HAR_WCNN algorithm utilizes a wide time-domain CNN for multi-environment sensor data, but like the previous approach, it may struggle with contextual variations and human behavior differences in real-world smart home settings.

Incorporating semantic techniques into human activity recognition (HAR) systems offers higher-level interpretation of sensor data, enabling a more structured and comprehensible understanding of activities [4]. By using ontologies and semantic analysis, HAR systems can reason and model actions effectively [26]. Several studies demonstrate how raw sensor data can be transformed into relevant activity descriptions using semantic models like RDF and SPARQL [27, 28]. These systems excel at contextualizing activities by linking objects, actors, and environments. However, traditional ontologies, which emphasize structure, often lack behavioral components. Larhrib et al. [21] propose incorporating behavior flow concepts into ontologies, improving validation and testing processes. Furthermore, ontology-based HAR systems [29] facilitate the modeling of human activities, temporal linkages, and hierarchical action dependencies, offering great potential for inferring higher-level behavior in environments like smart homes, where activities are ordered in time and space.

Ni et al. [20] propose a human activity representation model for smart homes, crucial for discreet wellness monitoring, especially for older adults. Developed using the NeON methodology, the model consists of three ontology categories: users, smart home contexts, and Activities of Daily Living (ADL). It employs the DOLCE+ DnS Ultralite (DUL) ontology as an upper ontology, ensuring high reusability and compatibility across different smart home applications.

Context-aware systems enhance HAR by utilizing real-time environmental data to infer human activities more accurately [22]. These systems reason across multiple contextual dimensions, such as time, location, and object proximity, to identify complex, multi-step tasks. They integrate inference mechanisms to dynamically update their environmental understanding and adjust predictions based on changing conditions. Moulouel et al. [23] introduce an ontology-based framework for identifying user context and detecting anomalies in ambient intelligence systems. Their approach leverages event calculus in answer set programming (ECASP), integrating machine learning, probabilistic planning, and common-sense reasoning. To manage uncertainty, the framework employs probabilistic fluents and a partially observable Markov decision process (POMDP). Additionally, the authors propose a hybrid technique [25] that combines deep learning with probabilistic commonsense reasoning for action prediction in ambient intelligence (AmI) environments. Deep learning models recognize human hands, interior locations, and environmental objects, while ECASP incorporates probabilistic fluents for reasoning. An ontology-based representation of the user's surroundings enables temporal projection and abductive reasoning, contributing to an explainable artificial intelligence (XAI) approach. HAR research, informed by cognitive psychology [24], extends beyond motion detection to infer cognitive states and goals, enhancing behavioral analysis. This multidisciplinary approach advances applications in health and well-being.

Existing HAR techniques still have limitations despite significant progress. Many systems struggle to manage complex, overlapping tasks, and scaling these systems in larger smart home contexts remains a challenge. Furthermore, ongoing issues with sensor accuracy, unbalanced datasets, and real-time processing restrict the use of HAR systems in practical applications. As research progresses, these issues must be addressed to develop reliable and effective HAR systems that can meet a range of user demands and seamlessly integrate into smart homes.

3 Method

This section details the proposed approach. It describes the datasets used, preprocessing steps, model architecture, and evaluation methodology.

3.1 Datasets description

To evaluate the proposed approach, we utilized two datasets: the CASAS Aruba dataset and the Orange4Home dataset. These datasets were selected due to their structured time-series data and their ability to capture diverse activity and behavior recognition scenarios.

The CASAS Aruba dataset, part of Washington State University's smart homes project, was compiled using thirteen motion sensors, three door sensors, five temperature sensors, and three light sensors, all installed in a single-occupant house. Each recorded sensor event is timestamped to the second, capturing data over several months. This continuous, real-time observation documents the temporal evolution of daily activities and offers a comprehensive understanding of daily patterns. Over the course of 132 days, eleven distinct actions were recorded. The structure of the Aruba dataset is detailed in Table 2. It is important to note that certain behaviors are more prevalent than others, resulting in an imbalance within the dataset.

Given the large size of the dataset (1,048,576 rows), preprocessing is necessary to handle missing data, filter out irrelevant information, and prepare the data for mapping to the ontology. Due to the very large quantity of data to process, we used only the first four days of the Aruba dataset, which equals 20,897 events. Table 1 is a sample of the data in Aruba dataset with the column specifications.

The second dataset, the Orange4Home dataset [37], was selected due to its structured time-series data and complex

Table 1: Sample data from the CASAS Aruba dataset

Date	Time	Sensor	Value	Activity	Log
2010-11-04	00:03:50.209589	M003	ON	Sleeping	begin
2010-11-04	00:03:57.399391	M003	OFF	Sleeping	begin
2010-11-04	00:15:08.984841	T002	21.5	Sleeping	begin
2010-11-04	00:30:19.185547	T003	21	Sleeping	begin
2010-11-04	00:30:19.385336	T004	21	Sleeping	begin

Table 2: CASAS Aruba Dataset Structure

Column Name	Description
Date	The date when the sensor data was recorded.
Timestamp	The exact time the data was captured.
Sensor ID	Identifier for the sensor that recorded the data.
Sensor Value	The value recorded by the sensor, varying by type: • M Type (Motion Sensor): ”On” or ”Off”. • T Type (Temperature Sensor): Temperature value. • D Type (Door Sensor): ”Open” or ”Closed”.
Activity Type	The activity being performed.
Activity Value	Indicates whether the activity is beginning (Begin) or ending (End).

multimodal nature, making it suitable for diverse activity and behavior recognition scenarios.

The Orange4Home dataset provides approximately 180 hours of multimodal data from a two-floor smart home, where a single occupant performed daily activities over four consecutive work weeks. It includes 17 annotated activities: *Entering, Living, Preparing, Cooking, Washing the dishes, Eating, Watching TV, Computing, Using the toilet, Going up, Going down, Using the sink, Showering, Dressing, Reading, and Napping*. The datasetś complexity, with wearable, object, and ambient sensors, enables the evaluation of ontology-based reasoning and the SVM classifier. To maintain consistency, the data was segmented using a sliding window of 3 seconds, which balances fine-grained activity detection with sufficient contextual information.

To ensure consistency and optimize the dataset for input into our model, we applied a series of preprocessing steps tailored to its characteristics. These steps included label encoding, linear interpolation, normalization, segmentation, and one-hot encoding.

The preprocessing of the Orange4Home dataset followed a structured approach. The 17 activities were assigned unique numerical labels using label encoding. To address missing sensor readings, forward-fill interpolation was applied. Continuous sensor values were then normalized to the range [0,1]. Additionally, sensor fusion was performed by integrating data from all available sensors into a unified feature matrix. The data was subsequently divided into 3-second overlapping windows, with the majority label assigned to each segment. Finally, one-hot encoding was used to transform the activity labels into binary vectors suitable for classification.

Table 3 summarizes the characteristics of the Orange4Home dataset used.

To ensure robust evaluation, the dataset was split into 70% training, 15% validation, and 15% testing. The training set was used for model learning, the validation set for hyperparameter tuning, and the test set for final performance assessment.

Table 3: Summary of Orange4Home dataset characteristics

Property	Details
Sensor Types	Wearable, object, and ambient sensors
Data Types	Binary, integer, real number, categorical
Number of Activities	17 annotated activities
Participants	Single occupant
Preprocessing	Normalization, segmentation into fixed-size windows
Recording Duration	180 hours over four consecutive work weeks

3.2 Mapping the dataset to an ontology

This section explains the process of mapping information from the smart home environment to an ontology that semantically describes the relationships between actions, sensors, and contextual components. This ontology provides a structured framework for representing human activities and their associated sensor data, enabling advanced querying and inferencing.

Our ontology includes classes: *Activity*, *ActivityTime*, *Location*, *Sensor*, *Object*, and *Subject*. The instances of the subclasses of *Activity* distinguish occurrences of activities based on their day and time of occurrence, allowing multiple occurrences of the same activity across different days and times of the day. Sensors are categorized into three types, each of them linked to specific locations within the smart home.

These concepts are connected by different relationships: *occursOn*, *hasStartTime*, *hasEndTime*, *hasActivityLocation*, *hasActor*, *hasTimeOfDay*, *hasSensorID*, and other properties, which capture the connections between activities, sensors, and their contextual details. This ontology facilitates structured semantic representations of the data, enabling advanced queries and inference about human activity patterns.

- *Activity* ($\mathscr{A}$): Represents the human activities described in the datasets. The class has the different activity types as subclasses, and each of these subclasses has its occurrences as instances based on the day and the time of occurrence.
- *ActivityTime* ($\mathscr{D}$): Represents the day of the activity, and the time of the day the activity is done.
- *Location* ($\mathscr{L}$): Represents the various spaces or rooms where sensors are installed in the smart home.(e.g., Bedroom, Kitchen).

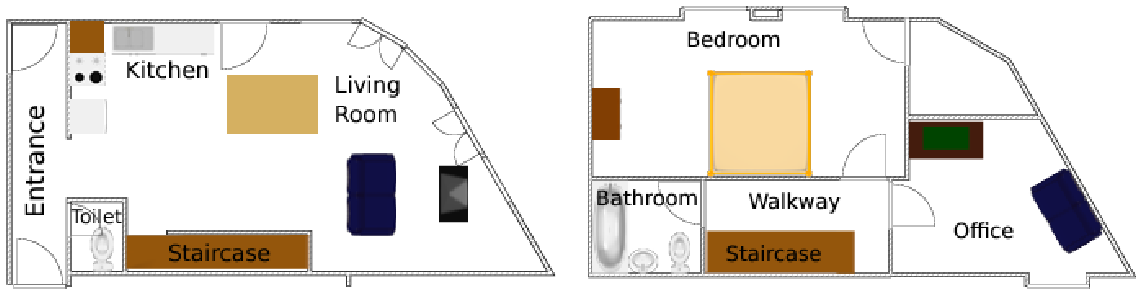


Figure 1: Smart Home Layout Used for Orange4Home Dataset Collection [37]

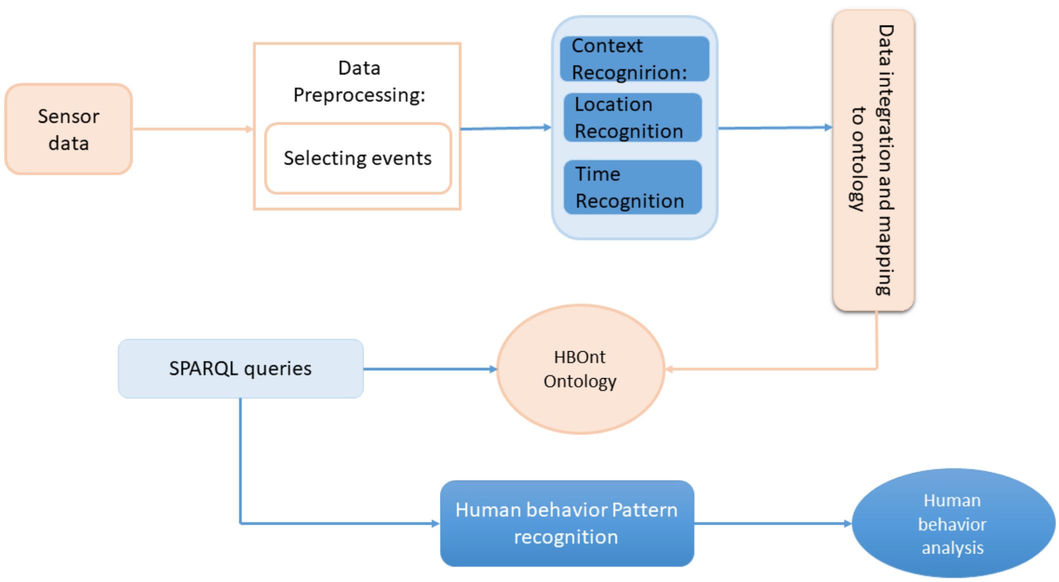


Figure 2: General architecture of the proposed method

- *Sensor* ($\mathscr{S}$): Represents the various types of sensors installed in smart homes:
  - *DoorSensor* ($\mathscr{S}_D$)
  - *MotionSensor* ($\mathscr{S}_M$)
  - *TemperatureSensor* ($\mathscr{S}_T$)

  Individuals are created to correspond to the actual sensors in the datasets for each type of sensor.
- *Object* ($\mathscr{O}$): Represents the different objects involved in the activities classified by the location the object can be found in (e.g., desktop, toothbrush).
- *Subject* ($\mathscr{S}_B$): Represents the various subjects performing the activities; in our datasets, we have a single subject.

These classes and subclasses are related using object and data properties, from which we cite:

- *hasActor* (*hasActivity* : $\mathscr{A} \rightarrow \mathscr{S}_B$): Each occurrence of the activities is linked to a specific *Subject*.
- *occursOn* (*occursOn* : $\mathscr{A} \rightarrow \mathscr{D}$): Each *Activity* is linked to a specific day and time of the day.
- *isMonitoredBy* (*isMonitoredBy* : $\mathscr{A} \rightarrow \mathscr{S}$): Each *Activity* is associated with one or more sensors in the location where it is monitored.
- *hasActivityLocation* (*hasActivityLocation* : $\mathscr{A} \rightarrow \mathscr{L}$): Each activity occurrence is associated with a specific *Location* in the smart home.

To capture the temporal aspects of activity occurrences, we define the data properties: *hasStartTime* (*hasStartTime* : $\mathscr{A} \rightarrow xsd:dateTime$) specifies the start time of each activity occurrence, and *hasEndTime* (*hasEndTime* : $\mathscr{A}_O \rightarrow xsd:dateTime$) specifies the end time of each activity occurrence. The XSD (XML Schema Definition), a World Wide Web Consortium recommendation, specifies how to describe an element formally. The xsd:dateTime is a Datatype of the XML Schema language definition. One way to conceptualize time values is as objects with integer-valued year, month, day, hour, and minute attributes.

For each row in the datasets, we create instances of the class *Activity* ($\mathscr{A}$) that capture contextual details (e.g., day, location, start time, and end time, object). Activity occurences ($\mathscr{A}_O$) are derived directly from the dataset's activity labels. Each occurrence of an activity is captured as an instance of *Activity* subclasses, with associated temporal data using the properties *hasStartTime*, *hasEndTime* and *hasActivityLocation*.

Sensors in the datasets are categorized into the subclasses *DoorSensor* ($\mathscr{S}_D$), *MotionSensor* ($\mathscr{S}_M$), and *TemperatureSensor* ($\mathscr{S}_T$). Each sensor is represented as an individual, with properties such as *SensorID*, *Location*, and *SensorType*. Locations ($\mathscr{L}$) represent different rooms in the smart home, and each sensor is linked to its respective location through the property *isInstalledIn*.

Each *Activity* type ($\mathscr{A}$) is linked to an *activityTime* ($\mathscr{D}$) and a *Location* ($\mathscr{L}$), and is associated with a *Subject* ($\mathscr{A}$) through the object properties *occursOn*, *hasActivityLocation*, and *hasActor* respectivly.

The formal relationships can be expressed as follows:

$$\forall a_0 \in \mathscr{A}, \exists d \in \mathscr{D}, \exists l \in \mathscr{L} \mid occursOn(a_0, d) \wedge hasActivityLocation(a_0, l) \tag{1}$$

$$\forall a_0 \in \mathscr{A}, \exists s \in \mathscr{S}_B \mid hasActor(a_0, s) \tag{2}$$

$$\forall a_o \in \mathscr{A} \mid hasStartTime(a_o, t_s) \wedge hasEndTime(a_o, t_e) \tag{3}$$

The data is converted into RDF triples, semantically encoding the connections between actions, their occurrences, and the contextual components in the smart home via the structured ontology mapping. This model facilitates the inference of activity patterns and behaviors, enabling sophisticated SPARQL queries. Figures 3 and 4 present an overview of the HBOnt-Ontology taxonomy. The detail box highlights part of the Stimulus-Sensor-Observation pattern in SSN [36]. To represent sensor values in observations, the *hasValue* property was added to the HBOnt ontology.

The *locatedAt* relation expresses proximity, for instance, *locatedAt* (Person, Place, Time Interval) indicates a person (Subject) being at a specific place during a given time interval. The Sensor concept includes two properties: *observes*, which describes sensor qualities (e.g., precision, resolution, response time), and *produces*, which describes the sensor's output (e.g., temperature, brightness, position). A temperature sensor (*TemperatureSensor*) has a unique identifier (Sensor ID), and the *locatedAt* property can infer a person's presence at a place during a time interval.

The Observation concept represents contextual data from sensors, with the *observed* property linking it to the Sensor concepts in the SSN ontology. Our robust taxonomy-based semantic model includes key contextual concepts such as *locatedAt*, *installedIn*, and *observed*. Physical devices like temperature and motion sensors are represented as *TemperatureSensor* and *MotionSensor*, respectively. Each sensor is uniquely identified by a Sensor ID, ensuring accurate inferences of a Subject's presence in a given space.

3.3 Human Behavior Analysis

Our SPARQL queries developed to examine human behaviors and their connections within the framework of cognitive psychology. These queries are designed to extract relevant information from the ontology to help understand why individuals engage in specific activities and the interactions between various activities, contexts, and motivations. Each query contributes to the broader analysis of cognitive processes associated with activity sequences while addressing a distinct aspect of human behavior.

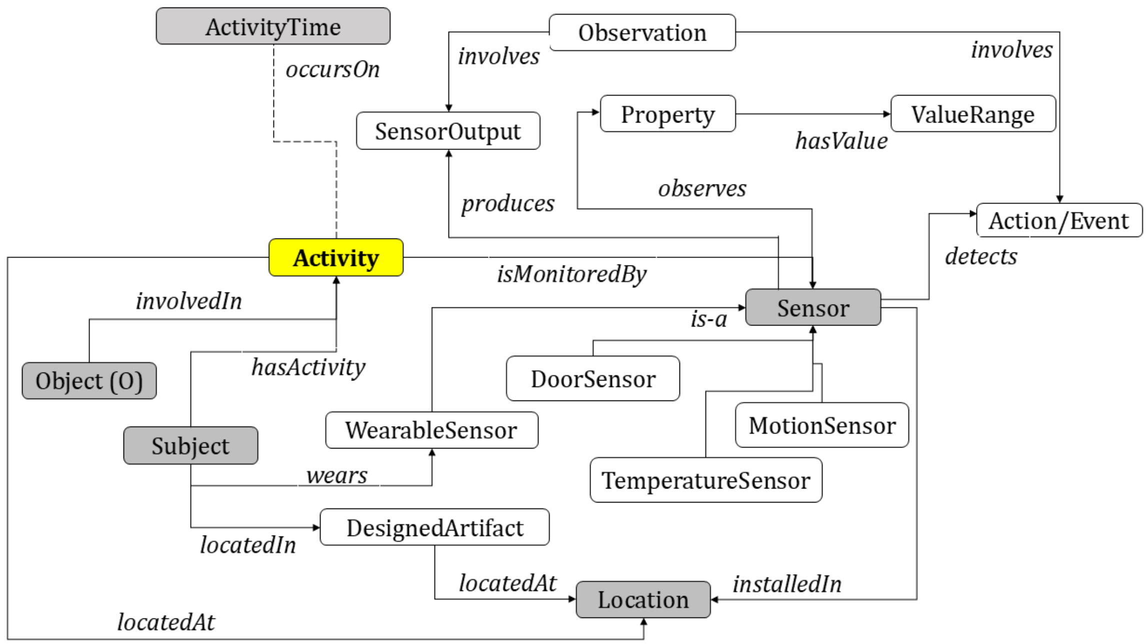


Figure 3: Main components of the HBOnt ontology to represent and organize knowledge about human activities.

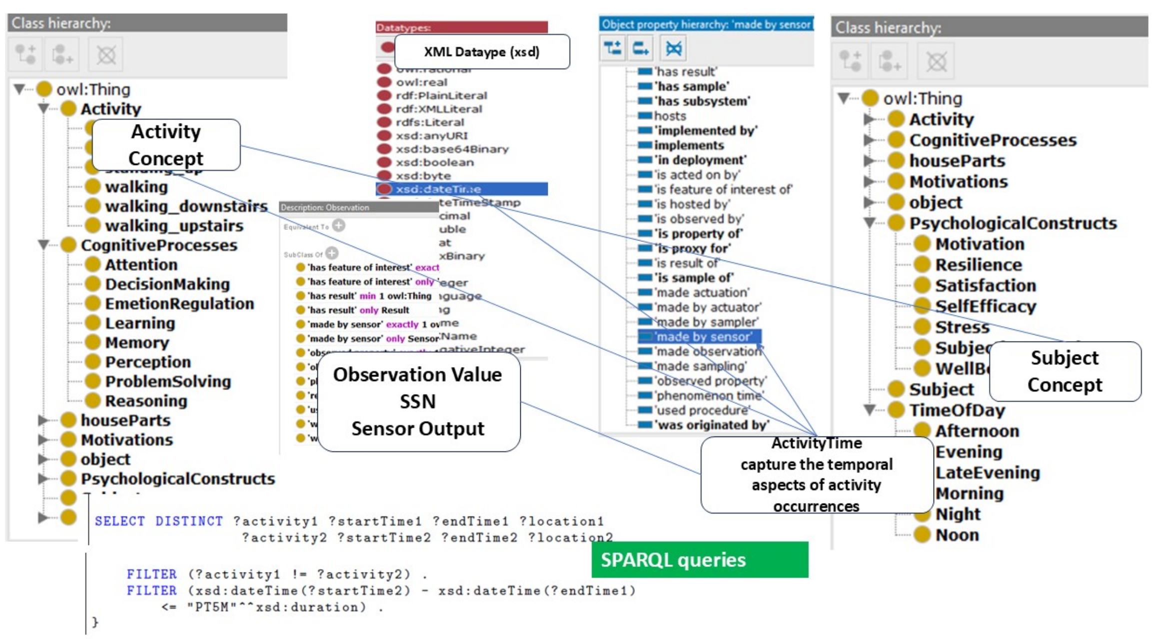


Figure 4: Structure of the ontology, adding semantic contextual information for each recorded activity occurrence, and modelling the relations of the concept Observation of the HBOnt ontology and the Subject and the Activity concepts.

Retrieve Activities of a Subject

```
SELECT DISTINCT ?activity ?startTime ?endTime
WHERE {
    <http://www.co-ode.org/ontologies/ont.owl#Mary> ex:hasActivity ?activity .
    ?activity ex:hasStartTime ?startTime ;
              ex:hasEndTime ?endTime .
}
```

This query returns specific tasks along with their start and end times for the subject. By identifying these behaviors, we can gain insights into the precise steps the individual took during a predetermined period. Understanding the sequence and duration of these activities is essential for analyzing cognitive patterns and behavioral motivations.

Filter Activities by Duration

```
SELECT DISTINCT
    (STRAFTER(STR(?activity), "http://www.co-ode.org/ontologies/ont.owl#") AS ?activityName)
    ?startTime ?endTime
WHERE {
    {
        SELECT ?activity ?startTime ?endTime
        WHERE {
            <http://www.co-ode.org/ontologies/ont.owl#Mary> ex:hasActivity ?activity .
            ?activity ex:hasStartTime ?startTime ;
                      ex:hasEndTime ?endTime .
        }
    }
    FILTER ((xsd:dateTime(?endTime) - xsd:dateTime(?startTime)) > "PT6M"^^xsd:duration) .
}
```

This query builds upon the previous one by restricting the activities to those lasting more than a specified period of time. By focusing on longer tasks, we can examine patterns that may indicate deeper cognitive engagement or preference.

Identify Activity Sequences with Contextual Locations

```

SELECT DISTINCT ?activity1 ?startTime1 ?endTime1 ?location1
                ?activity2 ?startTime2 ?endTime2 ?location2
WHERE {
    <http://www.co-ode.org/ontologies/ont.owl#Mary> ex:hasActivity ?activity1 .
    ?activity1 ex:hasStartTime ?startTime1 ;
               ex:hasEndTime ?endTime1 ;
               ont:hasActivityLocation ?location1 .

    <http://www.co-ode.org/ontologies/ont.owl#Mary> ex:hasActivity ?activity2 .
    ?activity2 ex:hasStartTime ?startTime2 ;
               ex:hasEndTime ?endTime2 ;
               ont:hasActivityLocation ?location2 .

    FILTER (?activity1 != ?activity2) .
    FILTER (xsd:dateTime(?startTime2) - xsd:dateTime(?endTime1) <= "PT5M"^^xsd:duration) .
}
```

This query retrieves the start and end times, as well as the locations of the activity pairs completed by the subject 'Mary.' By examining the temporal proximity of different activities and their surrounding contexts, we can infer potential linkages and reasons for the transitions between activities.

4 Experiments and Results

In this section, we evaluate the performance of our proposed framework, utilizing Protégé for ontology development and the GENA API in Java for implementation. The Human Behavior Ontology (HBOnt) was constructed from the dataset and queried using Apache Jena. The analysis, conducted on a machine with an Intel i5 CPU and 8 GB RAM, provides insights into human behaviors, particularly focusing on the sleep activity. By examining pre- and post-sleep activities, we uncover how daily routines and environmental conditions influence cognitive states, such as emotional health and stress management.

The study finds that regular sleep patterns are associated with cognitive strategies for stress management, aligning with existing research on the importance of sleep regularity. Additionally, the quality of sleep is linked to the nature of preceding activities, such as calming versus stimulating tasks, and the external environment, such as noise levels, reinforcing the cognitive psychology perspective that behavior and cognition are shaped by routines and surroundings.

SPARQL queries allow for detailed analysis of individual activity patterns, such as those of 'Mary' who shows a structured daily routine involving meal preparation, relaxation, and sleep. Query results reveal that activities like meal-preparation reflect personal values, such as creativity or care, while a preference for relaxation before bed suggests the importance of unwinding. The analysis of activity transitions further indicates a structured routine that balances rest and relaxation, highlighting the role of behavioral consistency in promoting well-being.

According to cognitive theory, the duration and timing of activities reveal underlying routines, physiological and mental needs, and responses to external stimuli. Unlike short impulsive acts, Query 1 screened for instances of the *Relax* activity, highlighting cases that show sustained focus or intentional rest, which can indicate deeper cognitive engagement.

- **Results for Query 1:**
  Activity: *Relax_2024-07-04_1*
  Start Time: *2024-07-04T09:29:23*
  End Time: *2024-07-04T09:34:05*
  Activity: *Relax_2024-07-06_1*
  Start Time: *2024-07-06T11:11:08*
  End Time: *2024-07-06T11:25:23*

By demonstrating cognitive recovery strategies, extended periods of relaxation align with the ideas of resource management and cognitive balance. As shown in these cases, longer or repeated rest intervals may indicate that the subject is maximizing its recovery after exertion.

The brief pauses between activities shown in Query 3 may indicate goal-driven behavior, where each activity aligns with a particular need or objective. This aligns with cognitive psychology's view of motivation as a structured action in response to external or internal stimuli.

- **Results for Query 3:**
  Start Time: *2024-07-06T11:11:08*
  End Time: *2024-07-06T11:25:23*
  Activity Name: *Relax_2024-07-06_1*
  Start Time: *2024-07-06T22:27:35*
  End Time: *2024-07-06T22:36:08*
  Activity Name: *Relax_2024-07-06_6*

The pattern of swiftly switching from *Relax* to other tasks, such as *Meal Preparation*, might indicate adherence to set schedules or routines linked to physiological requirements (like hunger), implying that some tasks are prioritized due to urgent, time-sensitive motives. This behavior aligns with motivation theories, which suggest that people balance leisure and obligations by performing specific tasks out of need or habit.

As shown with Query 3, shifts in position and activity provide information about behavior that adjusts to environmental stimuli.

- **Results for Query 3:**
  Start Time: *2024-07-06T18:29:36*
  End Time: *2024-07-06T18:41:18*
  Activity Name: *Relax_2024-07-06_2*
  Location:*Living*
  Start Time: *2024-07-04T16:36:10*
  End Time: *2024-07-04T17:06:00*
  Activity Name: *Meal_Preparation_2024-07-04_8*
  Location:*Kitchen*

In this instance, switching between tasks—for example, from *Relax* in one area to *Meal Preparation* in another—demonstrates cognitive flexibility. The ability to modify behavior in response to contextual signals or environmental resources is illustrated by this shift in tasks across locations, which shows awareness of and reactivity to the current environment. This flexibility is a key component of cognitive flexibility, where behavior is both goal-oriented and responsive to circumstances.

These examples illustrate a sample of our findings, demonstrating how ontology-based querying allows us to infer motivational states, evaluate behavior patterns, and detect abnormalities. Our ability to identify effective patterns, spot deviations that may indicate cognitive exhaustion, and recognize behaviors consistent with motivational theories is enhanced by the ontology framework.

Our ontology-based approach's flexibility in handling novel situations is one of its main benefits. Because ontologies are inherently adaptable, new concepts and relationships can be added with ease. This is essential for changing datasets and behavioral contexts. Our method can be expanded by adding additional classes or attributes in the ontology, providing scalability and adaptability, in contrast to traditional deep learning models that require costly retraining when fitted to new activities.

In HAR, interpretability is crucial, especially when behavior analysis is utilized to infer human intents. Because each conclusion is backed up by specific rules and connections, our ontology-based method provides great transparency and makes the reasoning process traceable and explicable. On the other hand, deep learning models are often referred to as 'black boxes,' having little interpretability concerning the underlying decision-making process. The openness of our approach facilitates the understanding of human behavior patterns, making it especially useful for situations where explainability is crucial.

The structure of our ontology models action sequences and their relationships, allowing for a more thorough explanation of human behavior. Our method can infer the motives underlying sequences of actions, unlike methods that only label individual activities (e.g., inferring "preparing to sleep" based on prior behaviors like "brushing teeth" and "changing clothes"). Because standard machine learning models typically lack the potential for such high-level thinking, this hierarchical and relational knowledge provides insights that are difficult to obtain.

A comparative overview of the main characteristics covered above is provided in Table 5, which also highlights the benefits of our ontology-based approach in HAR over deep learning-based models that are frequently employed in the literature.

The table 4 compares the query efficiency between the **CASAS Aruba** and **Orange4Home** datasets, focusing on how efficiently the ontology-based system can handle large amounts of data. For the **CASAS Aruba** dataset, which contains 11 activities, a total of **150 SPARQL queries** were executed. The system achieved an **average query time of 0.25 seconds**, with a **maximum query time of 0.35 seconds**. The query complexity for this dataset is classified as **medium**, indicating that the queries are moderately complex but that the system can handle them efficiently.

Table 4: Comparative Analysis of Query Efficiency (CASAS Aruba and Orange4Home)

Dataset	Number of Activities	Number of SPARQL Queries Run	Average Query Time (s)	Max Query Time (s)	Query Complexity
CASAS Aruba	11	150	0.25	0.35	Medium
Orange4Home	17	200	0.20	0.30	Medium

Table 5: Comparison of the proposed ontology-based approach with different types of deep learning models in HAR

Metric	Our Proposed Approach	Deep Learning Models
Accuracy of Inference	High correctness in composite activities	High in predefined labels [30]
Context Awareness	Detailed spatiotemporal context	Limited to training data context [31]
Flexibility and Adaptability	Easily extensible with new concepts	Requires retraining for new activities [32]
Interpretability	High, with transparent rules	Low, often a ”black box” [33]
Query Execution Efficiency	Rapid, supports real-time inference	Slower, may require preprocessing [34]
Behavioral Understanding	Hierarchical and sequence-based	Limited to labeled data [35]

On the other hand, the **Orange4Home** dataset, which involves 17 annonated activities, had **200 SPARQL queries** executed. The average query time was slightly shorter, at **0.20 seconds**, with a **maximum query time of 0.30 seconds**. Similarly to CASAS Aruba, the complexity of the query for Orange4Home is also classified as **medium**. Despite the higher number of activities, the query performance remains efficient.

Although the suggested method yields good results in general, a few practical considerations should be noted. The quality of the sensor data and the effectiveness of its adaptation to various real-world environments can influence its success. To align with the evolution of activity patterns, the ontology may require periodic updates. In environments that change quickly or are characterized by high dynamics, additional optimization could assist in preserving real-time performance. However, our findings are promising, both datasets have consistently exhibited efficient query times, with averages remaining low and peak times falling within an acceptable range. The system can handle moderately complex queries, suggesting that it is appropriate for real-time applications in smart environments, particularly those requiring rapid and reliable processing of large amounts of data.

5 Conclusion

The proposed structured human behavioural ontology has practical applications that allow a cohesive transformation of raw sensor data into a structured model that captures various actions' time, location, and events relationships. The ontological querying process based on SPARQL queries examines distinct aspects of humans and contributes to the broader analysis of cognitive processes and behaviours. Moreover, queries extract relevant information from the ontology to help understand why individuals engage in specific activities and the interactions between various activities, contexts, and motivations.

Relying on the CASAS Aruba and Orange4Home datasets, we demonstrated the efficiency of the proposed semantic framework through a series of experiments, showing a significant improvement over conventional methods in capturing the complexities of human behaviour in smart homes. In future work, we will enhance this framework by incorporating deep learning techniques to refine the accuracy of behavioural analysis. Additionally, exploring the development of personalized recommendation systems could improve quality of life by offering tailored insights based on individual activity patterns.

Acknowledgments

Not applicable

References

[1] R. Raghavendra, V. Singh, S. Arora, and M. V. Gupta, „Sensor-based human activity detection using a novel machine learning algorithm in home automation”, *Multidisciplinary Science Journal*, vol. 6, 2024.

[2] P. Kulurkar, C. K. Dixit, V. C. Bharathi, A. Monikavishnuvarthini, A. Dhakne, and P. Preethi, „AI based elderly fall prediction system using wearable sensors: A smart home-care technology with IOT”, *Measurement: Sensors*, vol. 25, p. 100614, 2023.

[3] N. Agti, L. Sabri, O. Kazar, and A. Chibani, „A Novel Deep Learning Model for Smartphone-Based Human Activity Recognition”, in *Proc. Int. Conf. Mobile Ubiquitous Syst.: Comput., Netw., Serv.*, pp. 231–243, 2023.

[4] R. Mojarad, A. Chibani, F. Attal, G. Khodabandelou, and Y. Amirat, „A hybrid and context-aware framework for normal and abnormal human behavior recognition", *Soft Comput.*, vol. 28, no. 6, pp. 4821–4845, 2024.

[5] G. Diraco, G. Rescio, P. Siciliano, and A. Leone, „Review on human action recognition in smart living: Sensing technology, multimodality, real-time processing, interoperability, and resource-constrained processing", *Sensors*, vol. 23, no. 11, p. 5281, 2023.

[6] K. Moulouel, M. Arbane, A. Chibani, G. Khodabandelou, and Y. Amirat, „Stream reasoning approach for anticipating human activities in ambient intelligence environments", in *Proc. IEEE Int. Conf. Tools Artif. Intell. (ICTAI)*, pp. 666–671, 2022.

[7] L. Arrotta, G. Civitarese, and C. Bettini, „Dexar: Deep explainable sensor-based activity recognition in smart-home environments", *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 6, no. 1, pp. 1–30, 2022.

[8] Washington State University, „CASAS Datasets", available at: `https://casas.wsu.edu/datasets/`, accessed Sep. 25, 2024.

[9] D. Alghazzawi, O. Rabie, O. Bamasaq, A. Albeshri, and M. Z. Asghar, „Sensor-based human activity recognition in smart homes using depthwise separable convolutions", *Hum.-Cent. Comput. Inf. Sci.*, vol. 12, no. 50, 2022.

[10] A. R. Javed, R. Faheem, M. Asim, T. Baker, and M. O. Beg, „A smartphone sensors-based personalized human activity recognition system for sustainable smart cities", *Sustainable Cities Soc.*, vol. 71, p. 102970, 2021.

[11] D. Das, Y. Nishimura, R. P. Vivek, N. Takeda, S. T. Fish, T. Ploetz, and S. Chernova, „Explainable activity recognition for smart home systems", *ACM Transactions on Interactive Intelligent Systems*, vol. 13, no. 2, pp. 1–39, 2023.

[12] Y. Yu, K. Tang, and Y. Liu, „A fine-tuning based approach for daily activity recognition between smart homes", *Applied Sciences*, vol. 13, no. 9, p. 5706, 2023.

[13] A. K. Sahoo, V. Kompally, and S. K. Udgata, „Wi-fi sensing based real-time activity detection in smart home environment", in *Proc. IEEE Applied Sensing Conf. (APSCON)*, pp. 1–3, 2023.

[14] L. G. Fahad and S. F. Tahir, „Activity recognition and anomaly detection in smart homes", *Neurocomputing*, vol. 423, pp. 362–372, 2021.

[15] Y. Li, G. Yang, Z. Su, S. Li, and Y. Wang, „Human activity recognition based on multienvironment sensor data", *Information Fusion*, vol. 91, pp. 47–63, 2023.

[16] S. Mekruksavanich and A. Jitpattanakul, „LSTM networks using smartphone data for sensor-based human activity recognition in smart homes", *Sensors*, vol. 21, no. 5, p. 1636, 2021.

[17] L. Chen, C. Nugent, and G. Okeyo, „An ontology-based hybrid approach to activity modeling for smart homes", *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 1, pp. 92–105, 2013.

[18] D. Riboni and C. Bettini, „OWL 2 modeling and reasoning with complex human activities", *Pervasive and Mobile Computing*, vol. 7, no. 3, pp. 379–395, 2011.

[19] F. Palumbo, C. Gallicchio, R. Pucci, and A. Micheli, „Human activity recognition using multisensor data fusion based on reservoir computing", *Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 2, pp. 87–107, 2016.

[20] Q. Ni, I. P. de la Cruz, and A. B. Garcia Hernando, „A foundational ontology-based model for human activity representation in smart homes", *Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 1, pp. 47–61, 2016.

[21] M. Larhrib, M. Escribano, C. Cerrada, and J. J. Escribano, „An Ontological Behavioral Modeling Approach with SHACL, SPARQL, and RDF Applied to Smart Grids", *IEEE Access*, 2024.

[22] R. Mojarad, F. Attal, A. Chibani, and Y. Amirat, „A context-aware approach to detect abnormal human behaviors", in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 89–104, 2020.

[23] K. Moulouel, A. Chibani, and Y. Amirat, „Ontology-based hybrid commonsense reasoning framework for handling context abnormalities in uncertain and partially observable environments", *Information Sciences*, vol. 631, pp. 468–486, 2023.

[24] L. W. Barsalou, „Cognitive psychology: An overview for cognitive scientists", *Psychology Press*, 2014.

[25] K. Moulouel, A. Chibani, H. Abdelkawy, and Y. Amirat, „Hybrid approach for anticipating human activities in Ambient Intelligence environments", in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pp. 2006–2011, 2022.

[26] L. Chen and C. Nugent, „Ontology-based activity recognition in intelligent pervasive environments", *International Journal of Web Information Systems*, vol. 5, no. 4, pp. 410–430, 2009.

[27] G. Meditskos, S. Dasiopoulou, and I. Kompatsiaris, „MetaQ: A knowledge-driven framework for context-aware activity recognition combining SPARQL and OWL 2 activity patterns", *Pervasive and Mobile Computing*, vol. 25, pp. 104–124, 2016.

[28] G. Meditskos, S. Dasiopoulou, V. Efstathiou, and I. Kompatsiaris, „Sp-act: A hybrid framework for complex activity recognition combining OWL and SPARQL rules", in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 25–30, 2013.

[29] D. Hooda and R. Rani, „Ontology driven human activity recognition in heterogeneous sensor measurements", *Journal of Ambient Intelligence and Humanized*

Computing, vol. 11, no. 12, pp. 5947–5960, 2020.

[30] S. Zhang, Y. Li, S. Zhang, F. Shahabi, S. Xia, Y. Deng, and N. Alshurafa, „Deep learning in human activity recognition with wearable sensors: A review on advances", *Sensors*, vol. 22, no. 4, p. 1476, 2022.

[31] L. Miranda, J. Viterbo, and F. Bernardini, „A survey on the use of machine learning methods in context-aware middlewares for human activity recognition", *Artificial Intelligence Review*, vol. 55, no. 4, pp. 3369–3400, 2022.

[32] R. Kanjilal, M. F. Kucuk, and I. Uysal, „Sub-transfer Learning in Human Activity Recognition: Boosting the Outlier User Accuracy", *IEEE Sensors Journal*, 2023.

[33] E. Kim, „Interpretable and accurate convolutional neural networks for human activity recognition", *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 7190–7198, 2020.

[34] X. Cheng, L. Zhang, Y. Tang, Y. Liu, H. Wu, and J. He, „Real-time human activity recognition using conditionally parametrized convolutions on mobile and wearable devices", *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5889–5901, 2022.

[35] I. E. Jaramillo, C. Chola, J.-G. Jeong, J.-H. Oh, H. Jung, J.-H. Lee, W. H. Lee, and T.-S. Kim, „Human activity prediction based on forecasted IMU activity signals by sequence-to-sequence deep neural networks", *Sensors*, vol. 23, no. 14, p. 6491, 2023.

[36] P. Compton et al., „The SSN Ontology of the W3C Semantic Sensor Network Incubator Group", *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, 2012.

[37] J. Cumin, G. Lefebvre, F. Ramparany, and J. L. Crowley, „A dataset of routine daily activities in an instrumented home", in *Ubiquitous Computing and Ambient Intelligence: 11th International Conference, UCAmI 2017*, pp. 413–425, 2017.

Agti Nadia is a Ph.D. student at the University of Bordj Bou Arréridj, Algeria, specializing in Decision Support Systems Engineering. She obtained her Master's degree in Decisional Informatics and Multimedia from the University of Biskra. Her current research interests include artificial intelligence and human activity recognition.

Sabri Lyazid is an Assistant Professor at the Bachir Ibrahimi University. He holds a Ph.D. in Computer Science, Images, Signals and Intelligent Systems from University Paris-Est Créteil (UPEC). He is a senior consultant on artificial intelligence, information systems architectures, and cybersecurity. While at UPEC, he worked on several collaborative research projects (e.g., SembySem, Web of Objects, Predykot, A2nets) dealing with IoT and ambient assisted living, robotics, artificial intelligence, and access management. His research interests are in ontology, knowledge representation and reasoning, robotics, artificial intelligence, deep learning, cognitive psychology, the Internet of Things paradigm, and blockchain technology.

Kazar Okba holds degrees in Computer Science and Artificial Intelligence from the University of Constantine, including a PhD. He has been a Full Professor at the University of Biskra since 2011, where he also helped establish the Computer Science Department and founded the Smart Computer Science Laboratory (LINFI). Dr. Kazar has published over 415 papers, authored three books, and contributed to 17 book chapters. His research interests cover a wide range of topics, including artificial intelligence, multi-agent systems, health management, big data, IoT, cloud computing, and digital twins. He has served as a reviewer and session chair at international conferences and has been a visiting professor in several European countries and the UAE. He is currently a professor at Kalba University.

Heterogeneous Graph Auto-Encoder for Credit Card Fraud Detection

Sudipta Majumder*
Moirangthem Tiken Singh†
Rabinder Kumar Prasad‡
Abhijit Boruah§
Gurumayum Robert Michael¶
N K Kaphungkui‖
N. Hemarjit Singh**

Abstract

The digital revolution has led to increased credit card usage and a corresponding rise in fraudulent activities. Traditional fraud detection methods often overlook the interconnected nature of financial data. This study presents a novel approach using Graph Neural Networks (GNNs) with attention mechanisms and heterogeneous graph structures to enhance credit card fraud detection. The method builds heterogeneous graphs to represent complex interactions among entities such as cardholders, merchants, and transactions. An autoencoder, trained on legitimate transactions, learns latent features to identify anomalies. The model's performance is evaluated using benchmark datasets and compared against existing techniques like GraphSAGE and FI-GRL. Experimental results show that the proposed model achieves superior performance, with an AUC-PR of 0.89 and an F1-score of 0.81. The integration of GNNs, attention mechanisms, and autoencoders effectively mitigates issues like class imbalance and captures intricate data relationships. This research uniquely applies attention-based GNNs on heterogeneous graphs for fraud detection, improving accuracy by addressing class imbalance and leveraging rich relational data. Evaluation is restricted to specific datasets, and real-world deployment may need adaptation for broader financial environments. The approach can be adopted by financial institutions to enhance fraud detection accuracy, reduce false positives, and strengthen customer trust and operational efficiency.

Key Words: Credit card fraud detection; Graph Neural Networks; Auto-encoders; Heterogeneous graphs; Class imbalance.

1 Introduction

Financial transactions, especially credit card usage, have experienced a surge due to the digital revolution. This has resulted in a vast amount of financial data, empowering companies to comprehend customer behavior and utilize data for decision-making. On the other hand, the convenience that comes with this has a downside - there is a noticeable rise in fraudulent activities. Traditional methods of fraud detection often struggle to keep pace with the evolving nature of these schemes. In order to tackle this challenge, the field of machine learning (ML) has surfaced as a potent tool that can effectively identify and prevent fraudulent transactions [2]. By leveraging ML algorithms, it becomes possible to analyze massive amounts of financial data, identify recurring patterns, and pinpoint potential fraud through anomaly detection. They enable financial institutions to automate the fraud detection process, facilitating real-time monitoring of transactions and activities. To detect fraud effectively, many professionals rely on techniques such as decision trees, random forests, and support vector machines [16, 19].

The conventional approaches to detecting fraud often face difficulties in capturing the intrinsic interrelationships that exist within financial data. Transactions typically involve multiple parties, including cardholders, merchants, banks, and various other entities. The representation of financial transactions as a graph enables us to take advantage of the connections among them, thereby enhancing the effectiveness of fraud detection measures. Despite their widespread use, it is important to acknowledge that traditional methods may face difficulties in accurately differentiating between relevant and irrelevant relationships within the graph, thus impacting their ability to effectively detect fraudulent activity.

Graph Neural Networks (GNNs) excel at processing graph data and utilizing attention mechanisms to focus on the most

*Department of Computer Science and Engineering, Dibrugarh University Institute of Engineering and Technology, Dibrugarh University, Assam 786004, India. Email: sudipta2020@dibru.ac.in.

†**Corresponding author**. Department of Computer Science and Engineering, Dibrugarh University Institute of Engineering and Technology, Dibrugarh University, Assam 786004, India. Email: tiken.m@dibru.ac.in.

‡Department of Computer Science and Engineering, Dibrugarh University Institute of Engineering and Technology, Dibrugarh University, Assam 786004, India. Email: rkp@dibru.ac.in.

§Department of Computer Science and Engineering, Dibrugarh University Institute of Engineering and Technology, Dibrugarh University, Assam 786004, India. Email: abhijit.boruah@dibru.ac.in.

¶Department of Electronics and Communication Engineering, Dibrugarh University Institute of Engineering and Technology, Dibrugarh University, Assam 786004, India. Email: robertmichael@dibru.ac.in.

‖Department of Electronics and Communication Engineering, Dibrugarh University Institute of Engineering and Technology, Dibrugarh University, Assam 786004, India. Email: pipizs.kaps@gmail.com.

**Department of Electronics and Communication Engineering, Dibrugarh University Institute of Engineering and Technology, Dibrugarh University, Assam 786004, India. Email: nhsingh@dibru.ac.in.

ISCA Copyright© 2025

relevant entities and relationships within the network structure [47]. This makes them well-suited for tasks like fraud detection, where identifying the most critical factors contributing to a transaction's legitimacy is crucial. By applying attention, the GNN can prioritize information from neighboring nodes (e.g., cardholder's spending habits, merchant's location) that are most relevant to understanding the transaction's nature. This refined focus on critical relationships improves the model's ability to distinguish between normal transactions and those exhibiting suspicious patterns, potentially indicative of fraud.

In graph induction learning techniques, two types of graph representations of data are used: homogeneous graph [5] and heterogeneous graph [37]. Financial fraud data, especially involving credit cards, is inherently heterogeneous. It encompasses diverse entities like cardholders, merchants, and transactions, each with distinct attributes and relationships. Homogeneous graphs, which represent entities of the same type, may not fully capture this complexity. In contrast, heterogeneous graphs offer a more comprehensive representation, effectively capturing the multifaceted nature of financial transactions and the intricate relationships between entities within the financial ecosystem.

For instance, a heterogeneous graph might include nodes representing credit card numbers (`cc_num`), merchants information (`merchant_id`), and transaction numbers (`transaction_id`), with edges connecting them based on the specific relationship (e.g., a transaction between a cardholder and a merchant). This allows us to analyze the network structure and identify suspicious patterns that might be missed by simpler models. For instance, in Figure 1, the relationships between different data points are illustrated. These relationships are often overlooked by homogeneous graph learning algorithms and their variants.

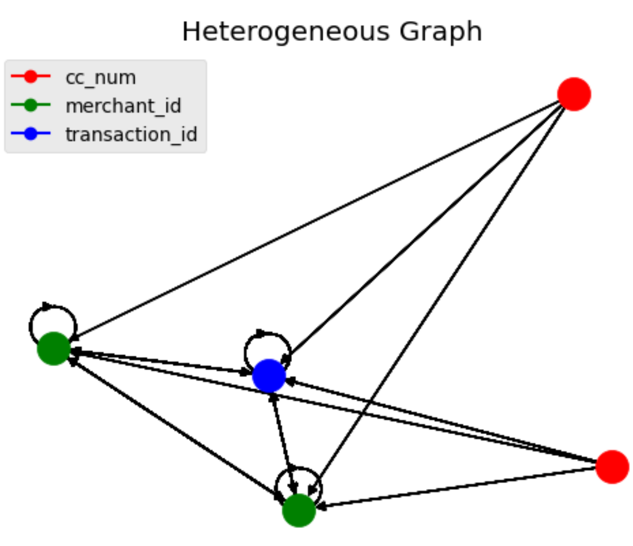


Figure 1: Relationships between different nodes.

The varying characteristics of nodes and edges in heterogeneous graph data make it difficult to apply GNNs directly, thereby necessitating a more sophisticated approach for information aggregation than what is typically used for homogeneous graphs. In addition, the effectiveness of supervised learning is often hindered by class imbalance in fraud data. This imbalance is characterized by a significantly smaller number of fraudulent transactions compared to genuine transactions. As a result, traditional supervised learning models struggle to learn effectively from such imbalanced data [8].

This work suggests a new approach that effectively handles heterogeneous graph data by leveraging advanced GNN techniques for aggregating information from diverse node and edge types. These techniques ensure that the varying attributes and relationships within the graph are adequately captured and utilized in the analysis process.

Furthermore, to tackle the issue of class imbalance, common techniques such as oversampling and undersampling [7] are used. Balancing class distribution can be achieved through oversampling, which generates more instances of the minority class (fraud transactions), or through undersampling, which reduces instances of the majority class (genuine transactions). Nonetheless, these approaches may be complicated and possess their own limitations.

To overcome these challenges, this approach integrates an autoencoder (AE) with a decoder that is trained on genuine transactions. By learning a latent representation, the AE can accurately reconstruct these transactions. The ability to detect fraudulent activities in complex heterogeneous graph data is enhanced by flagging deviations from the learned distribution during reconstruction, thereby addressing class imbalance.

Considering all scenarios discussed, this work aims to answer the following research questions (RQs):

- **RQ1: Effectiveness of GNNs with Attention for Fraud Detection:**How effectively can GNNs utilizing an attention mechanism detect and prevent credit card fraud when applied to a heterogeneous graph representation that captures the complex interrelationships within the financial ecosystem?
- **RQ2: Comparison of Autoencoder with Attention vs. Traditional Methods:** How does the proposed autoencoder-based fraud detection approach, which leverages GNNs with attention and is trained on a non-fraudulent transaction graph dataset, compare to traditional methods in terms of accuracy, efficiency, and scalability, especially considering significant class imbalance?

The methodology consists of several steps, one of which is the processing of a tabular dataset of financial transactions. This dataset is then transformed into a heterogeneous graph. As a result, the graph is subjected to analysis using autoencoders (AE) and graph neural networks (GNNs), which enables the identification of anomalies that can be linked to fraudulent activity. By focusing on the class imbalance problem, the proposed approach effectively tackles the challenge of fraud detection tasks. The results of this work have significant implications for businesses and financial institutions, empowering them to gain valuable insights into customer behavior and enhance their ability to identify and prevent fraudulent transactions. Ultimately, this work contributes to the advancement of fraud detection systems and the overall security of financial transactions in the digital era.

This paper provides a comprehensive discussion of the relevant literature in Section 2. The problem statement is outlined in Section 3, aiming to address a specific problem. The methodology employed in this research is elucidated in Section 4. The results obtained from this methodology are analyzed and presented in Section 5. Finally, Section 5.5 concludes the paper by summarizing the key findings and implications.

2 Literature Review

In this section, we introduce a range of notable works that cover various topics such as probabilistic graphical models, machine learning algorithms (including deep learning models), and advanced graph neural networks and their various variants.

Papers such as [38] and [34] aim to address the problem of fraud detection in credit card transactions by modeling these transactions using a Hidden Markov Model (HMM), a probabilistic graphical model. The primary difference between them lies in their approach: in the first paper, a card-centric HMM is employed to detect abnormalities in transactions, while the latter paper opts for a merchant-centric HMM model. Both methods have the capability to identify fraud in real-time for merchants, operating in conjunction with modern transaction processing systems that handle card transactions.

Additionally, [27] models credit card transaction sequences using the HMM approach, considering three distinct perspectives:

(i) Determining whether fraud is present or absent in the sequence.

(ii) Crafting sequences by fixing either the cardholder or the payment terminal.

(iii) Constructing sequences based on the spent amounts or the elapsed time between consecutive transactions. The combination of these three binary perspectives results in eight distinct sets of sequences derived from the training dataset of transactions. Each of these sequences is then represented using a Hidden Markov Model (HMM). Subsequently, each HMM assigns a likelihood to a transaction based on its sequence of preceding transactions. These likelihood values serve as additional features for the Random Forest classifier to detect fraud. In brief, this model provides a concept of sequential information flow during credit card transactions as part of a feature for a machine learning model.

The paper [18] explores the issue of credit card fraud detection and conducts a comparative analysis of three machine learning algorithms: logistic regression, Naïve Bayes, and K-nearest neighbor. To address the class imbalance, the authors utilize different proportions of the dataset and employ a random undersampling technique. They evaluate the algorithms based on various metrics. According to the results, the logistic regression-based model outperforms the prediction models derived from Naïve Bayes and K-nearest neighbor. The paper also suggests that applying undersampling techniques to the data before model development can lead to improved results. In addition, several machine learning algorithms, such as support vector machine (SVM) [35], random forest (RF) [35, 22], AdaBoost, and Majority Voting [31], as well as artificial neural network (ANN) [33, 1], are being explored as models for controlling fraudulent transactions in credit cards.

To enhance the performance of the above-mentioned models, [17] defines a model in an ML-driven credit card fraud detection system that uses the genetic algorithm (GA) for feature selection. After identifying optimal features, this detection system utilizes a range of ML classifiers, including Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Artificial Neural Network (ANN), and Naive Bayes (NB).

While the aforementioned models perform well, a significant class imbalance exists in the credit card fraud dataset, with non-fraudulent transactions vastly outnumbering fraudulent ones. As a result, these models tend to prioritize high precision by predominantly predicting the majority class. To address this issue, several machine learning models (referenced as [28]) employ one or a combination of oversampling and undersampling techniques (as mentioned in [6]).

The study cited as [3] conducts a comparative investigation of various approaches to address class imbalance. The findings indicate that a combination of oversampling and undersampling methods performs well when applied to ensemble classification models, including AdaBoost, XGBoost, and Random Forest. Deep learning algorithms such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), combined with a multilayer perceptron, are employed in the studies referenced as [28] and [13]. In [13], the authors use the Hybrid Synthetic Minority Oversampling Technique and Edited Nearest Neighbor (SMOTE-ENN) to balance the distribution of positive (fraud) and negative (non-fraud) instances in the dataset. However, the effectiveness of the SMOTE-ENN technique is crucial, as poor performance in resampling can significantly degrade the model's overall performance.

While oversampling and undersampling techniques can address class imbalance, they come with drawbacks like increased computational cost, potential for overfitting, and information loss (as discussed in [7, 44]). Additionally, they can be sensitive to noise [46] and have limited effectiveness for highly imbalanced datasets [12]. Therefore, [12] propose an approach for Chronic Kidney Disease (CKD) prediction using imbalanced data. Their method leverages information gain-based feature selection and a cost-sensitive AdaBoost classifier. However, this approach focuses on spatial data and might not be suitable for graph data due to potential loss of structural information and inadequate feature representation during feature selection. So, such models will often struggle to capture the full picture of fraudulent activity. As noted in [9], many methods focus solely on spatial data points representing financial transactions, neglecting the valuable insights from temporal relationships. This limitation hinders the ability of these models to identify evolving fraud patterns. Furthermore, many existing models rely solely on labeled data for training, restricting their ability to leverage the vast amount of unlabeled data available in real-world credit card transactions [36].

To address these issues, an increasing number of researchers are exploring graph-based techniques for fraud detection, as discussed in [9] and [23]. In this approach, datasets are transformed into graphs, providing a better understanding of the relationships among financial transactions. Graph Neural Network (GNN) algorithms, as detailed in [45], are applied to these graph datasets, allowing for efficient data aggregation from neighboring nodes and the extraction of node representations within the graph datasets. Among the popular GNN variants, GraphSAGE [14] and GAT [43] stand out, utilizing sampling methods and attention mechanisms to gather neighbor information. These techniques have shown promising results in the field of fraud detection. Furthermore, the paper [24] introduces an algorithm designed to tackle the class imbalance problem in graph-based fraud detection. It employs an algorithm known as Pick and Choose Graph Neural Network (PC-GNN) to perform imbalanced supervised learning on graphs. The PC-GNN algorithm selects neighbor candidates for each node within the sub-graph using a neighborhood sampler. Ultimately, it aggregates information from the chosen neighbors and different relations to derive the final representation of a target node. The paper reports that PC-GNN surpasses state-of-the-art baselines in both benchmark and real-world graph-based fraud detection tasks.

However, inconsistency issues arise in the aggregation process of GNN models when applied to fraud detection tasks [25]. The aggregation mechanism relies on the assumption that neighbors share similar features and labels. When this assumption breaks down, the aggregation of neighborhood information becomes ineffective in learning node embeddings.

To address these challenges, researchers in [25] and [39] have employed a multi-relational graph, known as a heterogeneous graph, for the classification of financial fraud. In [25], context inconsistency, feature inconsistency, and relation inconsistency in GNN are introduced. To tackle these inconsistencies, the authors propose a new GNN framework called GraphConsis. GraphConsis addresses these issues by combining context embeddings with node features to handle context inconsistency, designing a consistency score to filter inconsistent neighbors and generate corresponding sampling probabilities to address feature inconsistency, and learning relation attention weights associated with the sampled nodes to tackle relation inconsistency.

In [39], the authors propose semi-supervised methods that operate with heterogeneous graph datasets to address class imbalance issues in online credit loans. This paper utilizes a Graph-Oriented Snorkel approach to incorporate external expert knowledge, ultimately improving the performance of the learning algorithm when dealing with imbalanced datasets.

Another noteworthy work, [26], introduces a heterogeneous graph-based approach for detecting malicious accounts in financial transactions. The authors present an algorithm called GEM, which adapts to learn discriminative embeddings for various node types. GEM employs an aggregator to capture node patterns within each type and utilizes an attention mechanism to enhance algorithm efficiency.

In [32], the authors endeavor to design heterogeneous graph embeddings. Their approach incorporates heterogeneous mutual attention and heterogeneous message passing, incorporating key, value, and query vector operations (self-attention mechanism). This work features both a detector and an explainer, capable of predicting the validity of incoming transactions and providing insightful, understandable explanations generated from graphs to aid in subsequent business unit procedures.

The framework employed in [41] utilizes an algorithm for graph representation learning to create concise numerical vectors that capture the underlying network structure. The authors in this work assess the predictive capabilities of inductive graph representation learning with GraphSage and Fast Inductive Graph Representation Learning algorithms on credit card datasets characterized by significant data imbalance.

3 Problem Statement

A heterogeneous graph is a specialized graph data structure that comprises multiple types of nodes and edges, wherein each node or edge is uniquely associated with a distinct type. In essence, it represents a graph in which diverse node and edge types are interconnected. To provide a formal definition, the characteristics of a heterogeneous graph are delineated as follows:

Definition 3.1. *A heterogeneous graph, also known as a heterogeneous information network or heterogeneous network, is mathematically defined as $G = (V, E, T, R, X)$, where:*

- *V represents the set of nodes in the graph, and each node $v^t \in V$ is associated with a specific type $t \in T$, where T represents the set of node types.*
- *E represents the set of edges in the graph, and each edge $e^r \in E$ connects two nodes (v^{t_1}, v^{t_2}), where t_1 and t_2 are node types, and $r \in R$, where R represents the set of edge types or relationships.*
- *$X = \{X_v, X_e\}$ represents attributes of nodes and edges, respectively, where X_v represents the set of node attributes, and each node $v^t \in V$ can have a vector of attributes x_{v^t}, and X_e represents the set of edge attributes, where each edge $e^r \in E$ can have a vector of attributes x_{e^r}.*

By adhering to its definition, the financial fraud dataset can be depicted as a heterogeneous graph. These datasets encompass various entities, including customer or credit card numbers, merchants' names, and transaction numbers. These entities are represented as nodes in the graph, denoted by V. Specifically,

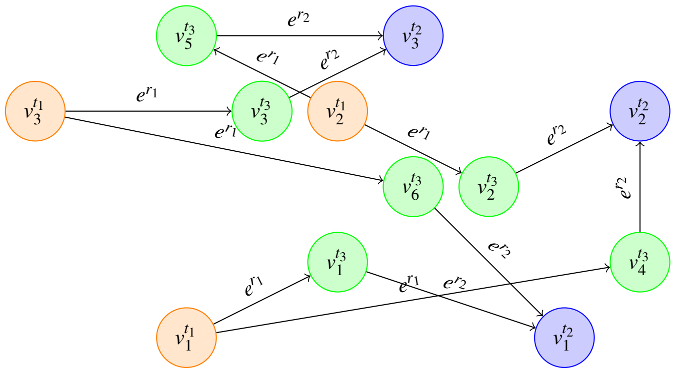


Figure 2: A heterogeneous graph illustrating different types of nodes and edges.

the nodes v^{t_1}, representing 'customer', v^{t_2}, representing 'merchants', and v^{t_3}, representing 'transaction', encapsulate the essence of this heterogeneous graph. Consequently, these nodes (v^{t_1}, v^{t_2}, and v^{t_3}) are distinguished by their respective types.

The heterogeneous graph depicted in Figure 2 illustrates a network where nodes are categorized into three distinct types: 'customers' (in orange), 'merchants' (in blue), and 'transactions' (in green). Each node type is uniquely identified by an index (v_i^t), where i indexes different instances of customers, merchants, and transactions within the same type t (e.g., $v_1^{t_1}$ for the first customer, $v_2^{t_1}$ for the second customer). The graph captures complex interactions: customers initiate transactions (e^{r_1}) that involve merchants (e^{r_2}). Notably, customers can engage in multiple transactions across different merchants, as represented by multiple transaction nodes ($v_1^{t_3}, v_2^{t_3}, \ldots, v_6^{t_3}$). This structured representation facilitates the analysis of interconnected relationships within heterogeneous networks, essential for understanding dynamics financial transactions.

Problem 1. *For the given graph $G = (V,E,T,R,X)$, the task is to determine whether it can be classified as fraudulent, considering that the transaction associated with the graph represents a fraudulent class.*

4 Methodology

The primary objective of this paper is to develop an encoder capable of learning graph embeddings for a given heterogeneous graph $G = (V,E,T,R,X)$, where V represents nodes, E denotes edges, T indicates node types, R specifies edge types, and X contains node attributes. This encoder is designed to effectively capture the complex information present in a heterogeneous graph, including both its structure and attributes. Subsequently, a decoder function f_{dec} reconstructs the graph. Figure 3 illustrates the model architecture used in this paper.

The model comprises l encoder units. The first encoder unit takes $(D(d^t), \phi, d^t)$ as input, where $D(d^t)$ represents the source nodes of $d^t \in V$, and ϕ denotes edges e^r from each source node

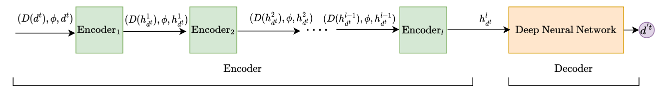


Figure 3: Encoder Units and Decoder Unit of the Model

to d^t. Each encoder unit processes these inputs to produce intermediate representations. The final output of Encoder$_l$ is fed into a decoder unit, implemented as a deep neural network, which utilizes the encoded information to generate d'^t.

The model calculates the reconstruction error by comparing the reconstructed graph to the original graph. This error measures the dissimilarity between the original input and the reconstructed output. A threshold for the reconstruction error is established to identify data points that deviate significantly from normal patterns. Any data point with a reconstruction error exceeding the threshold is classified as an anomaly, indicating a deviation from expected behavior, such as fraudulent activity in a financial transaction network.

While the methods discussed above claim to perform well with unbalanced heterogeneous graph datasets, techniques such as autoencoders and decoders, as presented in [40, 10, 11], offer alternative solutions. For instance, [11] successfully addressed imbalanced medical datasets using a modified Sparse Autoencoder (SAE) and Softmax regression for enhanced diagnosis. However, SAEs are less suitable for data with inherent relationships between elements, which is particularly relevant for fraud detection in transactional networks, where connections between nodes are crucial for identifying suspicious activity. Similarly, [10] employed a Stacked SAE (SSAE) for credit card default prediction on imbalanced data. Nevertheless, SSAEs, like SAEs, lack the ability to explicitly prioritize information from relevant neighboring nodes. This limitation necessitates a different approach for this work, which leverages a transactional network to represent data and identify fraudulent activities.

4.1 Dataset Preprocessing

The heterogeneous graph G is constructed from a financial transaction dataset, where nodes represent entities (e.g., customers, merchants, transactions) and edges represent interactions (e.g., payments, refunds). Node types T include "Customer," "Merchant," and "Transaction," while edge types R include "Pays," "Receives," and "Refunds." Node attributes X include features such as transaction amount, timestamp, customer demographics, and merchant category.

The dataset is preprocessed as follows:

- **Graph Construction**: Transactions are modeled as nodes connected to customer and merchant nodes via typed edges. For each transaction node d^t, the set of source nodes $D(d^t)$ includes the associated customer and merchant.
- **Feature Normalization**: Continuous attributes (e.g., transaction amount) are standardized to have zero mean

and unit variance. Categorical attributes (e.g., merchant category) are one-hot encoded.

- **Handling Missing Data**: Missing attributes are imputed using the mean (for continuous features) or mode (for categorical features) of the respective node type.
- **Graph Partitioning**: For large datasets, the graph is partitioned into subgraphs using community detection to facilitate scalable processing.

The preprocessed graph ensures that node and edge types are preserved, and attributes are in a suitable format for the encoder.

4.2 Encoder for Heterogeneous Graph

Based on the study by [15], a heterogeneous graph encoder for the autoencoder has been designed (Figure 4). For each destination node $d^t \in V$ and its source nodes $D(d^t) \in V$, the encoding process f^{enc} is applied as follows:

$$h^l_{d^t} = f_{\text{reparam}}\Big(\text{Linear}_{d^t}\left(f^{\text{enc}}_{\forall v \in D(d^t)}(h^{l-1}_{v^t}, e^r, h^{l-1}_{d^t})\right) \oplus h^0_{d^t}, \text{mean}(h^{l-1}_{d^t}), \log(h^{l-1}_{d^t})\Big) \quad (1)$$

Here, $l = 1, 2, \ldots, E_L$ represents the encoder layer, with a maximum of E_L layers, and initial values are set as $(h^0_{v^t}, e^r, h^0_{d^t}) = (v^t, e^r, d^t)$. The function $\text{Linear}_{d^t} : \mathbb{R}^{\frac{\text{dim}}{k}} \to \mathbb{R}^{\text{dim}}$ denotes a linear projection.

The encoding process f^{enc} is defined as:

$$f^{\text{enc}}_{\forall v^t \in D(d^t)} = \bigoplus_{\forall v^t \in D(d^t)} \Big(f^{\text{Attent}}(v^t, e^r, d^t) \cdot f^{\text{Mssg}}(v^t, e^r, d^t)\Big) \quad (2)$$

The attention mechanism is:

$$f^{\text{Attent}}(v^t, e^r, d^t) = \text{Softmax}\Big(\underset{\forall k \in [1,H]}{\|}\text{Att}^k(v^t, e^r, d^t)\Big) \quad (3)$$

Inspired by [42], the attention for each edge e^r is calculated using k-heads:

$$\text{Att}^k(v^t, e^r, d^t) = \text{LinearS}^k_t\left(h^{l-1}_{v^t}\right) \cdot W^{\text{Att}}_r \cdot \left(\text{LinearD}^k_t\left(h^{l-1}_{d^t}\right)\right)^T \quad (4)$$

Here, LinearS^k_t and LinearD^k_t map from $\mathbb{R}^{\text{dim}*}$ to $\mathbb{R}^{\frac{\text{dim}}{k}}$, and $W^{\text{Att}}_r \in \mathbb{R}^{\frac{\text{dim}}{k} \times \frac{\text{dim}}{k}}$ is a learnable edge matrix for edge type r.

The message passing function is:

$$f^{\text{Mssg}}(v^t, e^r, d^t) = \underset{\forall k \in [1,H]}{\|} \text{LinearM}^k_t\left(h^{l-1}_{v^t}\right) W^{\text{Mssg}}_r \quad (5)$$

The reparameterization function f_{reparam}, inspired by [21], models the latent variable probabilistically:

$$\begin{aligned} h^l_{d^t} &= f_{\text{reparam}}\Big(\text{mean}(h^l_{d^t}), \log(h^l_{d^t})\Big) \\ &= \text{mean}(h^l_{d^t}) + \varepsilon \cdot \exp\left(\frac{1}{2} \cdot \log(h^l_{d^t})\right) \end{aligned}$$

where $\varepsilon \sim \mathcal{N}(0, 1)$.

4.3 Decoder for Heterogeneous Graph

The decoder reconstructs the graph's structure and attributes, accounting for its heterogeneous nature. For each node d^t, a node decoder f_{dec} reconstructs attributes d'^t:

$$d'^t = f_{\text{dec}}(h^l_{d^t}) \quad (6)$$

The decoder is a multi-layer perceptron (MLP) with type-specific parameters to handle different node types. The loss function is defined as:

$$L = \sum_{\forall N} \sum_{\forall t} \text{LOSS}(d^t, d'^t) \quad (7)$$

The loss function is specified as the mean squared error (MSE) for continuous attributes:

$$\text{LOSS}(d^t, d'^t) = \|d^t - d'^t\|^2_2 \quad (8)$$

For categorical attributes, cross-entropy loss is used. The total loss combines losses across all node types, weighted by their prevalence in the dataset to address class imbalance.

4.4 Algorithm

Algorithm 1 Fraud Detection on a Heterogeneous Graph

Require: Heterogeneous Graph G
Ensure: 'Fraud' or 'Not Fraud'

for $d^t \in G$ **do**
  $(h^0_{v^t}, e^r, h^0_{d^t}) \leftarrow (v^t, e^r, d^t)$ ▷ Initialization
  for $l \leftarrow 1$ **to** E_L **do** ▷ Message Passing Layers
    for $v^t \in D(d^t)$ **do** ▷ Neighborhood of d^t
      $h^l_{d^t} = \text{Linear}_{d^t}\left(f^{\text{enc}}(h^{l-1}_{v^t}, e^r, h^{l-1}_{d^t})\right) \oplus h^0_{d^t}$
    end for
    $h^l_{d^t} = f_{\text{reparam}}\Big(\text{mean}(h^l_{d^t}), \log(h^l_{d^t})\Big)$ ▷ Reparameterization
  end for
  $d'^t = f_{\text{dec}}(h^l_{d^t})$ ▷ Output Layer
end for
$L = \text{LOSS}(d^t, d'^t)$ ▷ Loss Calculation
if $L <$ Threshold() **then**
  return 'Non-Fraud'
else
  return 'Fraud'
end if

The algorithm depicted in Algorithm 1, outlines the method for detecting fraud in a heterogeneous graph structure. Here's a detailed breakdown of each step:

1. Input (Heterogeneous Graph G): This represents the financial transaction network, containing nodes (customers, merchants, transactions) and edges (interactions) with their respective types.

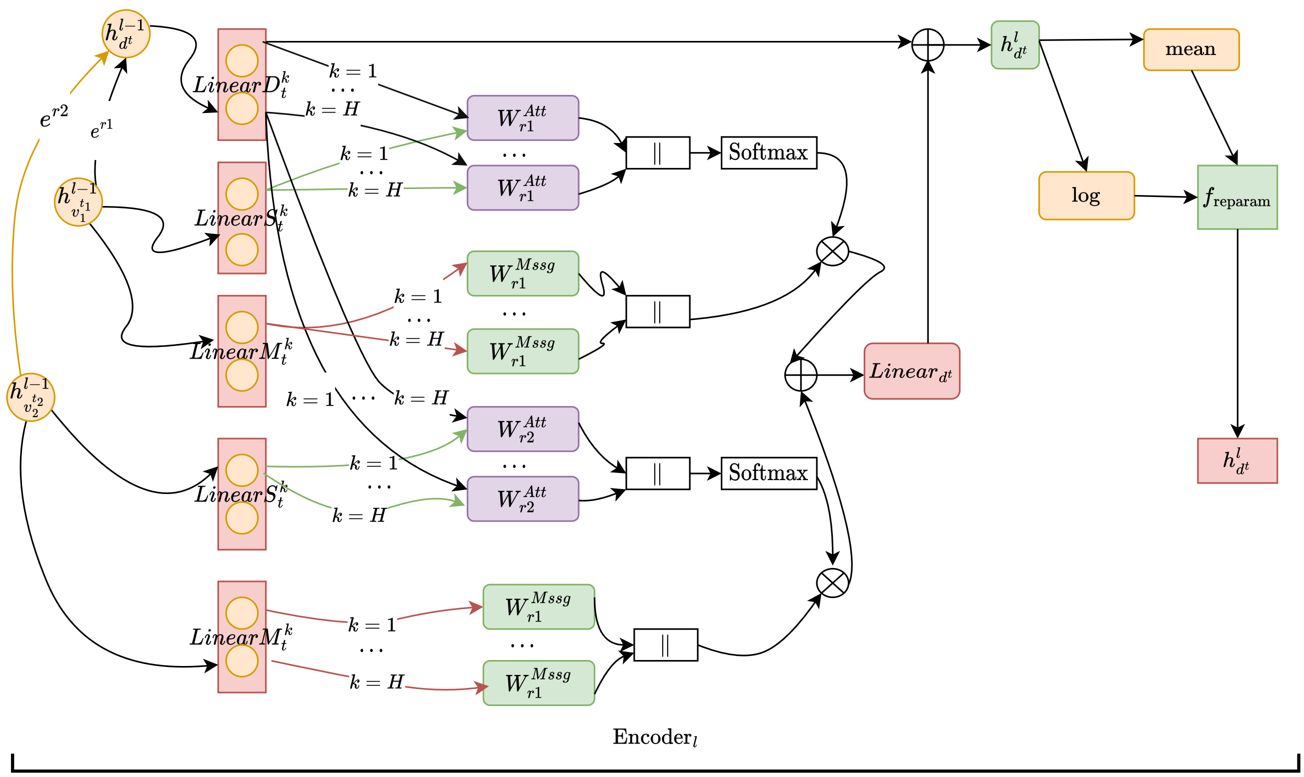


Figure 4: Encoder Unit for Heterogeneous Graphs. e^{r1} and e^{r2} denote edges from source nodes $v_1^{t_1}$ and $v_2^{t_2}$ to destination node d^t. At $l=0$, it represents the initial encoder layer, producing $h^1_{d^t}$, and so on. k ranges from 1 to H, $|$ signifies concatenation, $\oplus$ denotes addition, and $\otimes$ indicates dot product.

2. Output: **"Fraud" or "Not Fraud"**: The algorithm classifies the transaction associated with the input graph as either fraudulent or legitimate.
3. Algorithm Steps:
   - For each node d^t in the graph G, node d^t is initialized with $(h^0_{v^t}, e^r, h^0_{d^t})$. It includes the features of the node itself $h^0_{d^t}$, the connecting edge type e^r, and the initial representation of the source node $h^0_{v^t}$.
   - Message Passing Layers (L Layers):
     - This loop iterates through a predefined number of layers (E_L) in the GNN architecture.
     - Within each layer l:
       - For each node v^t in the neighborhood of the current node d^t:
         - A message function f^{enc} (Equation 2) aggregates information from the source node's hidden representation $h^{l-1}_{v^t}$, the edge type e^r, and the previous hidden representation of the destination node $h^{l-1}_{d^t}$. The message undergoes a linear transformation with Linear_{d^t} as per equations (3-5).
         - By utilizing the attention mechanism, the messages undergo transformation and are subsequently combined with the initial hidden representation of the destination node $h^0_{d^t}$ through element-wise addition ($\oplus$).
       - The message passing happens iteratively for all neighbors of d^t.
       - The updated hidden representation $h^l_{d^t}$ is subjected to f_{reparam} (Equation 1) after message aggregation. Mean and logarithm are utilized in hidden representation to ensure greater stability during training.
4. Output Layer: The final hidden representation $h^l_{d^t}$ is passed through the decoder function f_{dec} (Equation 6) to produce the prediction vector d'^t.
5. Loss Calculation: The difference between the predicted output d'^t and the original node feature d^t is evaluated using a loss function LOSS. The LOSS function can use a metric such as mean squared error or any other appropriate loss function.
6. Fraud Classification: The threshold for anomaly detection is determined using a validation set of non-fraudulent transactions. The reconstruction errors are computed, and the threshold is set as:

$$\text{Threshold} = \mu + 2\sigma \tag{9}$$

where μ and σ are the mean and standard deviation of the reconstruction errors, respectively. This ensures that approximately 95% of non-fraudulent transactions are classified as "Non-Fraud." The threshold is tuned on the validation set to balance precision and recall.

Algorithm 1 explains the entire framework of the model, which is designed to identify if a specific data point is linked to fraudulent behavior, resulting in one of two possible outcomes: 'Fraud' or 'Not Fraud.' The algorithm calculates a loss value to measure the difference between the original transaction node and its decoded version. The computation of this loss relies on a loss function that has been predetermined. The next step in the process is for the algorithm to compare the resulting loss with a predetermined threshold, once all the calculations have been completed. In the case where the loss falls below the designated threshold, the data point is classified as 'Not Fraud'. The overall time complexity of the algorithm can be approximated as $\mathcal{O}(nE)$ by summing up these components, with n representing the number of nodes in the graph.

5 Experiment

This paper assesses the effectiveness of the proposed model through a series of experiments on credit card fraud datasets and a comparison with other existing machine learning and deep learning models.

5.1 Performance Metrics

In order to evaluate the performance of various models, this article employed evaluation metrics that include the precision rate (PR), the recall rate (RR), the ROC curve, and the F1 score. These metrics are defined as follows:

$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$$

$$\mathrm{RR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

In this context, true positive (TP) and false positive (FP) indicate the number of correctly and incorrectly predicted instances of fraud, respectively. Conversely, true negative (TN) and false negative (FN) correspond to the count of transactions accurately and inaccurately predicted as non-fraudulent.

Meanwhile, the ROC curve illustrates the classifier's ability to differentiate between fraud and non-fraud categories. This curve is created by plotting the true positive rate against the false positive rate at different threshold levels. The AUC, which ranges from 0 to 1, encapsulates the information from the ROC curve. A value of 0 signifies that all classifier predictions are erroneous, while a value of 1 indicates a perfect classifier.

The F1 score represents the harmonic mean of precision and recall. Precision is the ratio of true positive predictions to the total predicted positives and recall is the ratio of true positive predictions to the total actual positives. It provides a single value that harmonizes precision and recall, facilitating a balanced evaluation of classifier performance.

$$\mathrm{F1} = 2 * \frac{(\mathrm{PR} * \mathrm{RR})}{(\mathrm{PR}+\mathrm{RR})}$$

Given that the dataset is imbalanced, the F1 score is particularly valuable because it considers both precision and recall. This score provides a straightforward way to assess a classifier's overall effectiveness in accurately identifying positive instances while minimizing false positives and false negatives.

Another parameter used to gain insight into the model's performance is the Precision-Recall curve (AUC-PR) [30]. This metric offers valuable insights, particularly in situations where class distribution is imbalanced [29].

5.2 Datasets

The dataset ([20]) used in this article simulates credit card transactions and includes genuine and fraudulent activities that occurred between January 1, 2019, and December 31, 2020. The data encompass transactions carried out by 1000 customers using credit cards issued by a variety of banks, engaging in transactions with a pool of 800 different merchants.

Types of Dataset	**Normal Data**	**Abnormal Data**
Training Dataset	1842743	9651
Testing Dataset	553574	2145

Table 1: Distribution of Fraudulent Transactions on Training and Testing Dataset

Table 1 illustrates the distribution of fraudulent and non-fraudulent transactions in a dataset. It shows the number of occurrences of each type of transaction, with "1" representing fraudulent (Abnormal) transactions and "0" representing non-fraudulent (Normal) ones. This analysis gives an indication of the skewed and unbalanced ratio of fraudulent to non-fraudulent transactions.

5.3 Analysis of Algorithms

In the article ([4]), some of the best machine learning algorithms that handle fraud datasets are listed. Here is the list used in the article:

- Linear Regression
- Logistic Regression
- Decision Tree
- SVM (Support Vector Machine)
- ANN (Artificial Neural Network)
- Naïve Bayes
- DNN (Deep Neural Network)
- K-Means
- Random Forest

- Dimensionality Reduction Algorithms
- Gradient Boosting (XGB) Algorithms

These algorithms cover a wide range of machine learning (ML) aspects, including association analysis, clustering, classification, statistical learning, and link mining. They hold a crucial place among the essential topics explored in research and development within the field of machine learning. However, when evaluating these algorithms with datasets, their performance often falls short of expectations due to the inherent imbalance present in the data.

Performance of Machine learning (ML) Algorithms					
ML Algorithm	Testing Accuracy	Testing F1 Score	Test Precision	Test Recall	AUC
Decision Tree	0.99	0.29	0.22	0.43	0.82
XGB Classifier	0.99	0.33	0.27	0.43	**0.96**
ANN	0.99	0.33	0.23	0.32	0.92
Deep NN	0.99	0.33	**0.40**	0.26	0.81
AE	–	**0.67**	0.50	0.99	0.52
VAE	–	**0.67**	0.50	0.99	0.54
Sparse AE	–	**0.67**	0.50	0.99	0.54

Table 2: Performance Measurement of Few Selected Machine Learning Algorithms

Table 2 provides the performance metrics for a few machine learning algorithms. The table shows that the F1 score of all machine learning algorithms is too low, suggesting that these algorithms could not handle unbalanced datasets properly. Since the F1 score is low and the AUC curve is high for all ML algorithms, it indicates that these algorithms are adept at distinguishing between abnormal and normal data, as evidenced by the high AUC value. However, the F1 score is low due to the models facing challenges in achieving both high precision and high recall, attributed to the imbalanced nature of the data.

These scenarios arise when the negative class dominates the dataset, creating a highly imbalanced situation. In such cases, models tend to classify instances as the majority class, resulting in high true negatives and low false positives but at the cost of missing true positives and having low recall. To address the challenges posed by unbalanced data, various algorithms are explored. One of the algorithms under consideration is the autoencoder algorithm.

The exploration involves simple autoencoders (AE) using deep neural networks and their variations, such as variational autoencoders (VAE) and sparse autoencoders (Sparse AE). Table 2 also shows the performance of the autoencoders. Regardless of the specific type, the model's performance is evaluated using key metrics. The F1 score, which harmonizes precision and recall, yielded a value of 0.67. This suggests that the models have achieved a reasonable balance between making accurate positive predictions and effectively capturing actual positive instances. Overall, the performance is decent, showing a well-rounded approach.

However, the narrative changes when examining the Receiver Operating Characteristics (ROC) curve and its corresponding Area Under the Curve (AUC). With an AUC of 0.57, it implies that the models struggle to distinguish between fraud and normal classes. Their ability to classify effectively in this context appears limited and performs only slightly better than random guessing.

In a deeper dive, the precision achieved by the autoencoder models in the test set is 0.50. This means that roughly half of the abnormal predictions it generates are accurate, while the other half are incorrect. On the other hand, the recall rate is impressive at 0.99. This means that the models excel at identifying almost all the actual abnormal instances present in the dataset.

In summary, while autoencoder models demonstrate balanced performance in terms of the F1 score, with commendable recall and reasonable precision, the AUC score and precision rates indicate room for improvement. Enhancing the discriminatory capacity of models and refining their positive prediction accuracy could be areas of focus to further elevate their performance in classification tasks.

5.4 Analysis of the Proposed Model

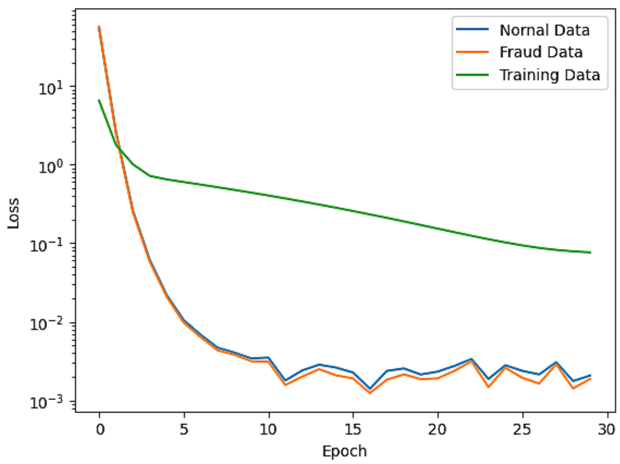
Parameter Name	**Value**
Size of Hidden Layers	64
Number of heads (H)	16
Number of Layers for the Encoder (l)	124
Number of Layers for the Decoder	64
Dropout Rate	0.4
Regularization Rate	0.01

Table 3: Values for different parameters used in the model.

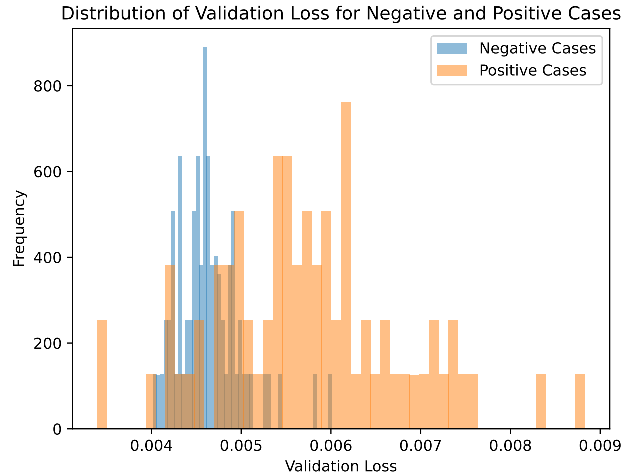
After tuning the parameters for different hyperparameters, the performance of the model is represented as shown in Figure 5. Finally, the proposed model uses the parameters defined in Table 4 to evaluate the model's performance.

In Figure 5a, the training loss is compared with the validation loss for positive (fraud) and negative datasets. This plot provides insight into how effectively the model handles overfitting and underfitting of the data. The model, using the parameters from Table 4, demonstrates immunity to both overfitting and underfitting, effectively managing these issues. Figure 5b illustrates the loss distribution (histogram) generated by the model from the dataset. This distribution shows the loss values for both positive and negative data in the dataset. The figure reveals that the loss for negative instances is concentrated between 0.004 and 0.005, while the loss for positive instances is distributed beyond 0.006.

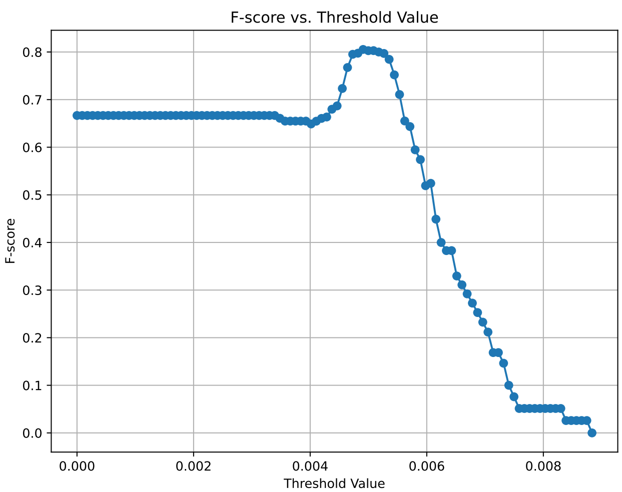
Figure 5c defines the model's F1 score versus the classification threshold value. From the figure, it can be seen



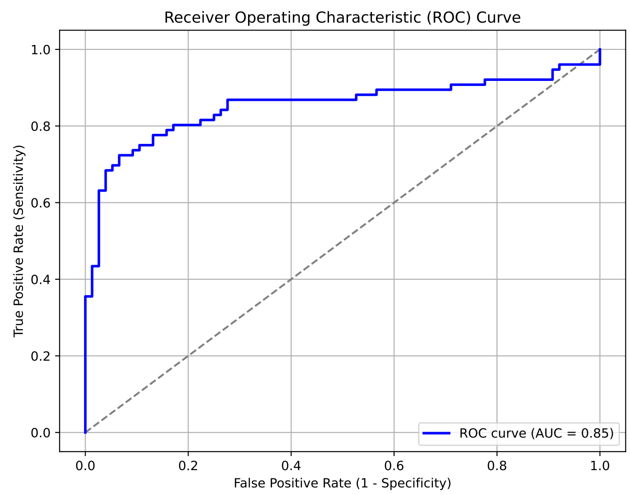
(a) Training loss and evaluation loss.



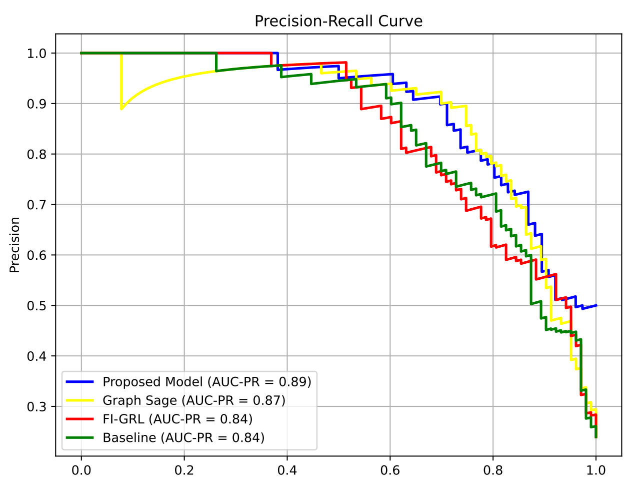
(b) Distribution of validation loss.



(c) F Score vs Threshold graph.



(d) ROC curve



(e) Precision-Recall curve of the model with an AUC-PR of 0.89.

Figure 5: Performance evaluations of the proposed model.

that the F1 score reaches its highest value of 0.81 at a loss value of 0.005. Additionally, the ROC curve was plotted based on the threshold, resulting in the ROC curve shown in Figure 5d, and an AUC of 0.85 was obtained for the model.

The Precision-Recall (PR) curve (Figure 5e) compares the performance of four algorithms: the Proposed Model, Graph Sage [41], FI-GRL [41], and Baseline [41]. The Proposed Model exhibits the highest performance with an AUC-PR of 0.89, indicating the best balance between precision and recall. Graph Sage follows closely with an AUC-PR of 0.87, showing strong but slightly inferior performance compared to the Proposed Model. Both FI-GRL and the Baseline models have an AUC-PR of 0.84, indicating moderate performance and similar effectiveness in maintaining precision and recall. Overall, the Proposed Model stands out as the most effective, followed by Graph Sage, with FI-GRL and Baseline performing similarly but less effectively.

Again, Table 4 summarizes the performance of various graph learning algorithms on metrics including AUC-PR, F1-Score, and ROC-AUC. The proposed model achieves the highest AUC-PR (0.89) and F1-Score (0.81) but has a lower ROC-AUC (0.85) compared to Graph Sage and XBoost, which achieve a ROC-AUC of 0.93.

Performance of Graph Learning Algotihms			
Graph Algorithm	**AUC-PR**	**F1 Score**	**ROC-AUC**
Proposed Model	**0.89**	**0.81**	0.85
Graph Sage and XBoost ([41])	0.86	0.80	**0.93**
FI-GRL([41])	0.84	0.70	0.92
Baseline([41])	0.84	0.74	0.91

Table 4: Performance Measurement of Graph Learning Algorithms. AUC-PR provides sufficient information to assess performance due to the imbalanced nature of the dataset used.

Finally, Figure 6 showcases the following algorithms: Proposed Model, Graph Sage and XBoost, FI-GRL, Baseline, Decision Tree, XGB Classifier, ANN (Artificial Neural Network), and Deep NN (Deep Neural Network). This radar chart highlights the exceptional performance of the Proposed Model, with high scores in F1 score, AUC-PR, and ROC-AUC, demonstrating a strong and balanced performance.

While Graph Sage and XBoost show great performance in class discrimination with high ROC-AUC, their AUC-PR is slightly lower, suggesting a trade-off when dealing with imbalanced datasets. Both FI-GRL and Baseline demonstrate strong classification performance with high ROC-AUC, but they may prioritize precision or recall at the expense of balance, resulting in a lower F1 score.

The Decision Tree and XGB Classifier face challenges in their competition, as the Decision Tree exhibits overall weakness, and the XGB Classifier lacks balance despite its

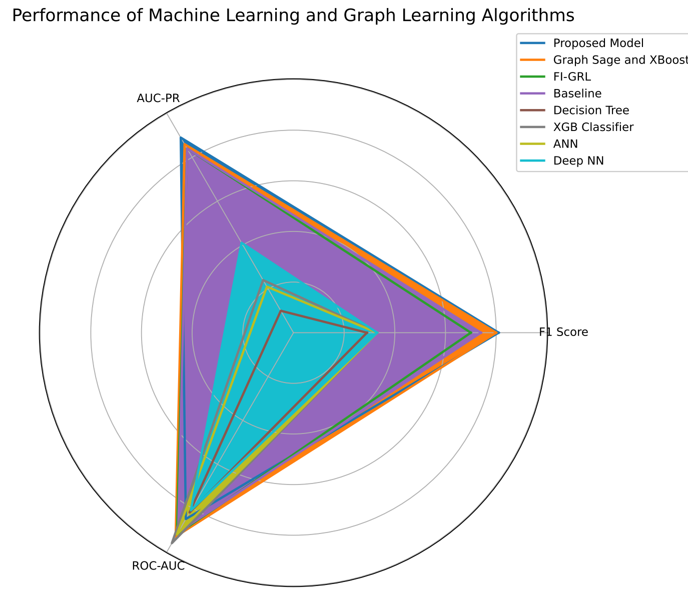


Figure 6: Performance Radar Chart. It compares several machine learning (ML) algorithms, including the Proposed Algorithm, using three key metrics: F1 Score, AUC-PR (Area Under the Precision-Recall Curve), and ROC-AUC (Area Under the Receiver Operating Characteristic Curve). This visualization highlights the strengths and weaknesses of each algorithm across these important performance metrics, providing a comprehensive view of their comparative effectiveness.

strong classification ability. Finally, ANN and Deep NN exhibit moderate performance across all metrics, lacking a clear specialization. With its balanced performance, the Proposed Model stands out as a strong candidate for general use, unlike other algorithms that focus on specific needs.

5.5 Conclusion

This paper presents a novel heterogeneous graph autoencoder with an attention mechanism to extract meaningful patterns from complex graph structures. The encoder generates node embeddings, which are used to form a probabilistic distribution via a variational autoencoder, capturing uncertainty and enabling diverse node sampling. This addresses the first research question effectively. A deep neural network then processes these embeddings to reconstruct the original node representations, enhancing their quality within the heterogeneous graph. Reconstruction errors from the decoder are analyzed to distinguish fraudulent from non-fraudulent transactions, with a simple search algorithm determining an optimal threshold, addressing the second research question. The proposed model is benchmarked against state-of-the-art methods, including GraphSAGE and FI-GRL, consistently outperforming these baselines and resolving the third research question.

Despite its strengths, the model has limitations. Scalability is a challenge for very large graphs, as time complexity remains high despite optimizations like neighbor sampling. Performance depends heavily on hyperparameter tuning, including embedding dimensions, attention heads, and network depth. The model's generalizability to datasets with different structural properties or domains beyond finance is untested. It also relies on rich node and edge attributes, which may be unavailable in some real-world scenarios. Additionally, the static graph assumption limits its ability to capture evolving fraud patterns, and the model lacks interpretability, hindering explanation of predictions. The Gaussian-based thresholding approach may be suboptimal for datasets with non-Gaussian error distributions.

These limitations suggest several directions for future work. Advanced sampling techniques, automated hyperparameter optimization, and transfer learning could enhance scalability, robustness, and adaptability. Incorporating temporal graph modeling would enable detection of dynamic fraud patterns, while attention visualization and explainability methods could improve interpretability. Replacing static thresholding with adaptive or non-parametric approaches may improve anomaly detection. Finally, integrating the autoencoder with supervised or rule-based methods could boost performance and practicality in real-world fraud detection systems.

Statements and Declarations

Competing Interests

The authors declare that there are no competing interests associated with this research work.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Informed Consent

Informed consent was obtained from all individual participants included in the study.

Data Availability

The datasets generated and/or analyzed during the current study are available in upon reasonable request from the corresponding author.

AI Assistant

This article acknowledges the use of ProWritingAid, an AI-powered writing assistance tool, to enhance grammar, readability, and overall writing quality. The tool was employed to identify and correct grammatical errors, improve sentence structure, and refine word choice while ensuring clarity and coherence. It is affirmed that all substantive ideas, analysis, and conclusions presented in this document are original and solely attributable to the author(s). ProWritingAid was used exclusively for linguistic refinement and did not influence the core content or arguments.

References

[1] Oluwatobi Noah Akande, Sanjay Misra, Hakeem Babalola Akande, Jonathan Oluranti, and Robertas Damasevicius. A supervised approach to credit card fraud detection using an artificial neural network. In Hector Florez and Ma Florencia Pollo-Cattaneo, editors, *Applied Informatics*, pages 13–25, Cham, 2021. Springer International Publishing.

[2] Abdulalem Ali, Shukor Abd Razak, Siti Hajar Othman, Taiseer Abdalla Elfadil Eisa, Arafat Al-Dhaqm, Maged Nasser, Tusneem Elhassan, Hashim Elshafie, and Abdu Saif. Financial fraud detection based on machine learning: A systematic literature review. *Applied Sciences*, 12(19):9637, Sep 2022.

[3] Ranjeet Kumar Ranjan Amit Singh and Abhishek Tiwari. Credit card fraud detection under extreme imbalanced data: A comparative study of data-level algorithms. *Journal of Experimental & Theoretical Artificial Intelligence*, 34(4):571–598, 2022.

[4] Vinay Arora, Rohan Singh Leekha, Kyungroul Lee, and Aman Kataria. Facilitating user authorization from imbalanced data logs of credit cards using artificial intelligence. *Mobile Information Systems*, 2020:8885269, Oct 2020.

[5] Deyu Bo. *Homogeneous Graph Neural Networks*, pages 27–59. Springer International Publishing, Cham, 2023.

[6] Paula Branco, Luís Torgo, and Rita P. Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.*, 49(2), aug 2016.

[7] Jason Brownlee. How to combine oversampling and undersampling for imbalanced classification. `https://shorturl.at/J4d2X`, 2021. Accessed: October, 2023.

[8] Patience Chew Yee Cheah, Yue Yang, and Boon Giin Lee. Enhancing financial fraud detection through addressing class imbalance using hybrid smote-gan techniques. *International Journal of Financial Studies*, 11(3), 2023.

[9] Dawei Cheng, Xiaoyang Wang, Ying Zhang, and Liqing Zhang. Graph neural network for fraud detection via spatial-temporal attention. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3800–3813, 2022.

[10] Sarah A Ebiaredoh-Mienye, E Esenogho, and Theo G Swart. Artificial neural network technique for improving prediction of credit card default: A stacked sparse autoencoder approach. *International Journal of Electrical and Computer Engineering*, 11(5):4392, 2021.

[11] Sarah A Ebiaredoh-Mienye, Ebenezer Esenogho, and Theo G Swart. Integrating enhanced sparse autoencoder-based artificial neural network technique and softmax regression for medical diagnosis. *Electronics*, 9(11):1963, 2020.

[12] Sarah A Ebiaredoh-Mienye, Theo G Swart, Ebenezer Esenogho, and Ibomoiye Domor Mienye. A machine learning method with filter-based feature selection for improved prediction of chronic kidney disease. *Bioengineering*, 9(8):350, 2022.

[13] Ebenezer Esenogho, Ibomoiye Domor Mienye, Theo G. Swart, Kehinde Aruleba, and George Obaido. A neural network ensemble with feature engineering for improved credit card fraud detection. *IEEE Access*, 10:16400–16407, 2022.

[14] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018.

[15] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, WWW '20, page 2704–2710, New York, NY, USA, 2020. Association for Computing Machinery.

[16] SK Saddam Hussain, E Sai Charan Reddy, K Gangadhar Akshay, and T Akanksha. Fraud detection in credit card transactions using svm and random forest algorithms. In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pages 1013–1017. IEEE, 2021.

[17] Emmanuel Ileberi, Yanxia Sun, and Zenghui Wang. A machine learning based credit card fraud detection using the ga algorithm for feature selection. *Journal of Big Data*, 9(1):24, 2022.

[18] Fayaz Itoo, Meenakshi, and Satwinder Singh. Comparison and analysis of logistic regression, naïve bayes and knn machine learning algorithms for credit card fraud detection. *International Journal of Information Technology*, 13(4):1503–1511, Aug 2021.

[19] Rongrong Jing, Hu Tian, Gang Zhou, Xingwei Zhang, Xiaolong Zheng, and Daniel Dajun Zeng. A gnn-based few-shot learning model on the credit card fraud detection. In *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, pages 320–323, 2021.

[20] Kartik2112. Credit card transactions fraud detection dataset. Retrieved 2023-09-21 from `https://www.kaggle.com/datasets/kartik2112/fraud-detection`, 2021.

[21] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[22] Tzu-Hsuan Lin and Jehn-Ruey Jiang. Credit card fraud detection with autoencoder and probabilistic random forest. *Mathematics*, 9(21), 2021.

[23] Yuhua Ling, Ran Zhang, Mingcan Cen, Xunao Wang, and M. Jiang. Cost-sensitive heterogeneous integration for credit card fraud detection. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 750–757, 2021.

[24] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: A gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the Web Conference 2021*, WWW '21, page 3168–3177, New York, NY, USA, 2021. Association for Computing Machinery.

[25] Zhiwei Liu, Yingtong Dou, Philip S. Yu, Yutong Deng, and Hao Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1569–1572, New York, NY, USA, 2020. Association for Computing Machinery.

[26] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 2077–2085, New York, NY, USA, 2018. Association for Computing Machinery.

[27] Yvan Lucas, Pierre-Edouard Portier, Léa Laporte, Liyun He-Guelton, Olivier Caelen, Michael Granitzer, and Sylvie Calabretto. Towards automated feature engineering for credit card fraud detection using multi-perspective hmms. *Future Generation Computer Systems*, 102:393–402, 2020.

[28] Ibomoiye Domor Mienye and Yanxia Sun. A deep learning ensemble with data resampling for credit card fraud detection. *IEEE Access*, 11:30628–30638, 2023.

[29] Neptune.ai. F1 score, accuracy, roc auc, pr auc: Evaluation metrics you need to know. `https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc`, 2023. Accessed: 2024-05-25.

[30] David M. W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *ArXiv*, abs/2010.16061, 2011.

[31] Kuldeep Randhawa, Chu Kiong Loo, Manjeevan Seera, Chee Peng Lim, and Asoke K. Nandi. Credit card fraud detection using adaboost and majority voting. *IEEE Access*, 6:14277–14284, 2018.

[32] Susie Xi Rao, Shuai Zhang, Zhichao Han, Zitao Zhang, Wei Min, Zhiyao Chen, Yinan Shan, Yang Zhao, and Ce Zhang. Xfraud: Explainable fraud transaction detection. *Proc. VLDB Endow.*, 15(3):427–436, nov 2021.

[33] Asha RB and Suresh Kumar KR. Credit card fraud detection using artificial neural network. *Global Transitions Proceedings*, 2(1):35–41, 2021. 1st International Conference on Advances in Information, Computing and Trends in Data Engineering (AICDE - 2020).

[34] William N. Robinson and Andrea Aria. Sequential fraud detection for prepaid cards using hidden markov model divergence. *Expert Systems with Applications*, 91:235–251, 2018.

[35] S K Saddam Hussain, E Sai Charan Reddy, K Gangadhar Akshay, and T Akanksha. Fraud detection in credit card transactions using svm and random forest algorithms. In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 1013–1017, 2021.

[36] Xiang Sheng, Yifan Li, Zhiyuan Liu, and Maosong Sun. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1234–1241, 2023.

[37] Chuan Shi, Xiao Wang, and Philip S. Yu. *Structure-Preserved Heterogeneous Graph Representation*, pages 29–69. Springer Singapore, Singapore, 2022.

[38] Abhinav Srivastava, Amlan Kundu, Shamik Sural, and Arun Majumdar. Credit card fraud detection using hidden markov model. *IEEE Transactions on Dependable and Secure Computing*, 5(1):37–48, 2008.

[39] Hao Tang, Cheng Wang, Jianguo Zheng, and Changjun Jiang. Enabling graph neural networks for semi-supervised risk prediction in online credit loan services. *ACM Trans. Intell. Syst. Technol.*, sep 2023. Just Accepted.

[40] Huang Tingfei, Cheng Guangquan, and Huang Kuihua. Using variational auto encoding in credit card fraud detection. *IEEE Access*, 8:149841–149853, 2020.

[41] Rafaël Van Belle, Charles Van Damme, Hendrik Tytgat, and Jochen De Weerdt. Inductive graph representation learning for fraud detection. *Expert Systems with Applications*, 193:116463, 2022.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[43] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[44] Fangyuan Yang, Kang Wang, Lisha Sun, Mengjiao Zhai, Jiejie Song, and Hong Wang. A hybrid sampling algorithm combining synthetic minority over-sampling technique and edited nearest neighbor for missed abortion diagnosis. *BMC Medical Informatics and Decision Making*, 22(1):344, 2022.

[45] Bingxu Zhang, Changjun Fan, Shixuan Liu, Kuihua Huang, Xiang Zhao, Jincai Huang, and Zhong Liu. The expressive power of graph neural networks: A survey, 2023.

[46] Zhong-Liang Zhang, Rui-Rui Peng, Yuan-Peng Ruan, Jian Wu, and Xing-Gang Luo. Esmote: an overproduce-and-choose synthetic examples generation strategy based on evolutionary computation. *Neural Computing and Applications*, 35(9):6891–6977, 2023.

[47] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

Authors

Sudipta Majumder earned his PhD from NERIST, Arunachal Pradesh, India. His research interest include IOT, cyber security and optimization techniques.

Moirangthem Tiken Singh earned his PhD from Dibrugarh University, India. His research interests include cybersecurity, semantic communication, and graph learning.

Rabinder Kumar Prasad earned his PhD from Dibrugarh University, India. His research interests span cybersecurity, data mining, and unsupervised learning.

Abhijit Boruah earned his PhD from Dibrugarh University, India. His research interests include computer vision, robot kinematics, and ontology design.

Gurumayum Robert Michael earned his PhD from Dibrugarh University, India. India. His research interests include Emotion recognition from speech, IoT and Circuit Design.

N K Kaphungkui received his PhD from Dibrugarh University, India. His research interests include speech processing and electronic circuit design.

N. Hemarjit Singh is pursuing his PhD at Dibrugarh University, India. His research interests include intelligent system design, wireless sensor networks, AI/ML for sensor systems, and industrial automation.

Reinforcement Learning for Neural System Towards Adaptive Intelligence

Priyanjli Gautam[1], Shuchi Mala[1], Aakanshi Gupta[1], Nidhi Mishra[1] and Narayan C. Debnath[2]
[1]Computer Science and Engineering, ASET, AUUP, Noida, India
[2]School of Computing and Information Technology, Eastern International University, Vietnam
Corresponding author email: agupta10@amity.edu and aakankshi@gmail.com

Abstract— The integration of artificial intelligence into health- care has catalyzed new research directions, particularly in neuroscience and medical imaging. While deep learning (DL) and classical machine learning (ML) have demonstrated significant effectiveness in brain tumor classification tasks, reinforcement learning (RL) is mostly underutilized, despite its biologically inspired principles. Unlike DL and ML, which are based on static pattern recognition and predictive modeling, RL provides dy- namic, feedback-driven learning and decision-making processes that are similar to neuronal plasticity in the human brain. This paper provides a framework for comparing ML, DL, and RL models, including Q-Learning and Deep Q-Learning (DQL), for the categorization of brain malignancies into four categories: glioma, meningioma, pituitary tumor, and no tumor. Our experimental results illustrate that how RL models outperform ML and DL models in accuracy, precision, recall, and F1- score measures. The study was guided by fundamental issues about RL's structural and functional similarities to biological systems, its potential to generalize via adaptive learning, and its impact on diagnostic accuracy and treatment optimization. This research shows that RL's interactive and self-improving character not only improves prediction performance but also provides a convincing framework for biologically grounded AI in healthcare. The findings indicate that reinforcement learning has transformative potential for medical diagnostics, providing both computing efficiency and neuropsychological relevance, boosting the future of intelligent, precision-driven oncology. In addition, the work supports the idea that biologically inspired AI systems can better simulate complicated brain events. Because of its higher generalization, RL is appropriate for a wide range of tumor categorization scenarios. These insights facilitate the door for real-time, adaptable diagnostic tools in clinical practice.

Keywords: Reinforcement Learning, Neuroscience, Adaptive Learning, Reward-Based Optimization, Neural Plasticity, Bio-plausible algorithms.

1. INTRODUCTION

Humans and other animals learn from their experiences. This can take the form of explicit demonstration, which is common in formal education. However, we frequently learn from trial and error, as well as input from our surroundings, which might be implicit or explicit. The human ability to make decisions and learn from experiences is fundamental to survival and adaptation [1]. At the core of this ability lies the concept of trial-and-error learning, where individuals optimize their behavior by engaging with their environment and making adjustments based on the rewards or punishments they receive [2]. This process has been extensively studied in neuroscience, psychology, and, more recently, in artificial intelligence. One of the most promising paradigms for modeling such learning is Reinforcement Learning, a computational approach that has shown remarkable parallels with human and animal learning processes [3]. In past few years, deep learning has gained widespread attention in neuroscience as a tool for modeling brain function. DL models have been used in a number of fields, such as motor control, navigation, vision, audition, and cognitive control. Advances in artificial intelligence (AI) and the ability to train deep learning models through supervised learning where labeled data directs optimization are driving this expanding interest [4]. Deep learning is still fundamentally limited in its ability to capture the adaptive and sequential character of real-world cognitive processes, despite these impressive advancements. Reinforcement Learning (RL) has emerged as an effective framework training artificial agents how to interact with their surroundings and maximize cumulative rewards in order to make sequential judgments [5].

At its core, reinforcement learning draws inspiration from neurobiology, particularly from the way that both humans and animals learn by means of mistakes. Important RL concepts like policy optimization, temporal difference updates, alongside reward-based learning are very similar to the biological mechanisms that the brain uses for learning and decision-making [6]. Notably, neuroscientific studies have demonstrated that the dopaminergic system of the brain is essential for encoding reward prediction mistakes, a notion that is quite similar to Temporal Difference (TD) learning in reinforcement learning [7]. Deep learning's dependence on static representations and supervised learning restricts its capacity to simulate cognitive processes that necessitate sequential decision-making and long-term planning, despite the fact that it has significantly advanced artificial intelligence. In addition to improving AI performance on challenging tasks like gaming and robotic control, latest advancements in deep reinforcement learning have also flourished our knowledge of neurocognitive processes [8]. For instance, distributional reinforcement learning has been used to explain reward prediction mistakes in dopaminergic neurons, and meta-reinforcement learning has been proposed as a model for prefrontal brain function. Considering these links between RL and neurology, RL-based models might provide a more realistic depiction of cognitive processes than conventional machine learning techniques [9]. Reinforcement learning, in contrast to deep learning, provides a physiologically realistic framework that closely resembles how the brain absorbs information, gains experience, and

ISCA Copyright© 2025

adjusts to novel situations.

RL enables agents to learn through reward-based interactions, whereas classical deep learning depends on fixed input-output mappings, necessitating large labeled datasets and substantial processing capacity. Because of this, it works especially well for simulating neuro cognitive processes like learning, adaptive control, and decision-making. Since dopaminergic reward prediction mistakes closely match classical RL algorithms like temporal difference learning, the brain mechanisms underlying RL have been extensively researched [10].

To rigorously assess the effectiveness of reinforcement learning in brain tumor classification, the study focuses on:

(a) *Reinforcement learning models do not show a statistically significant improvement over machine learning and deep learning models in brain tumor classification (Null Hypothesis), and*
(b) *Reinforcement learning models significantly outperform traditional machine learning and deep learning models in brain tumor classification due to their adaptive decision-making process (Alternate Hypothesis)*

Reinforcement learning (RL), in which an agent adapts its behavior responding to environmental feedback, providing a suitable framework for modeling cognitive processes including memory, comprehending, and the decision-making in neuroscience. Neural activity has previously been tied to traditional RL algorithms, especially in reward-processing brain regions like the dopaminergic system. Deep reinforcement learning(DRL) has shown promise in combining deep learning and reinforcement learning to represent convoluted cognitive tasks that necessitate long-term planning and hierarchical decision-making. In neuroscience, however, reinforcement learning offers clear benefits over Deep learning alongside additional machine learning approaches [11]. The adaptable and flexible nature of brain processes that depend on ongoing learning from reward-based feedback is difficult for DL to capture, despite its high effectiveness in pattern identification. RL, on the other hand, closely resembles how the brain learns from experience in that it naturally simulates adaptive behavior by using reward signals to improve decision-making over time [12].

This research explores how reinforcement learning (RL) provides a more effective framework than deep learning (DL) and machine learning (ML). We begin with an overview of RL concepts and their neurobiological analogies, highlighting how RL-based models offer a more organic explanation for learning processes, brain dynamics, and cognitive control. By analyzing the strengths and limitations of deep learning in neuroscience, we argue that reinforcement learning provides deeper insights into complex adaptive intelligence [13]. Furthermore, we compare the efficacy of RL with both deep and standard machine learning approaches in the context of medical imaging, specifically in the prediction and classification of brain tumors. RL based methods have the potential to enhance automated decision-making in medical diagnostics, optimize treatment plans, and improve diagnostic accuracy [14].

Additionally, we explored how reinforcement learning enables adaptive learning models that are more resilient than static deep learning models. Unlike conventional deep learning approaches, RL models can dynamically adjust based on real-time patient data, making them highly responsive to evolving medical conditions. The study also examines recent advancements in RL-driven healthcare applications, including automated radiology analysis and personalized medicine, which demonstrate the potential of RL in transforming medical diagnostics. By integrating reinforcement learning into diagnostic frameworks, researchers can develop more precise and effective tools for early disease detection. Finally, we outline the broader implications of RL for future research in medical diagnostics, AI, computational modeling, and neuroscience. By bridging the gap between reinforcement learning and neurocognitive science, this research argues that RL rather than DL should be at the forefront of research into brain-inspired intelligence.

2. Fundamental concepts of deep reinforcement learning

Reinforcement Learning (RL) explores instances in which an agent, or learner, is placed in an environment and must gradually improve its decision-making based on the conditions or states it encounters [15]. Unlike supervised learning, which uses explicit feedback to signal right behaviors, RL involves learning by trial and error without direct supervision. The basic goal is to create a behavioral policy that optimizes cumulative rewards over time, depending on input in the form of rewards or penalties resulting from the agent's activities. As a subfield of machine learning, RL studies how agents might learn optimal behaviors through interaction with their surroundings [16].

The agent performs actions, acquires feedback in the form of rewards or consequences, and adapts its approach to optimize the long-term accumulated reward. This approach varies from supervised learning, which uses labeled data for training [17]. The Markov Decision Process (MDP), which describes the environment using a set of states, serves as the basis for RL **S**, actions **A**, transition probabilities **P**, reward function **R** and a discount factor γ that accounts for future prizes. The intent of the RL agent is to learn a policy that maps states to actions in a way that maximizes the predicted cumulative reward over time [18].

A fundamental principle in RL is the balance of exploration and exploitation. While the agent must apply its current knowledge to make the best judgments (exploitation), it must also experiment with new actions (exploration) in order to discover possibly better tactics. This trade-off is critical for successful learning. The Bellman equation provides the theoretical basis for RL by recursively constructing the value function **V(s)**, which represents the predicted cumulative reward the agent can get from a given state: [19]

$$V(s) = \max_{a} \left[R(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V(s') \right] \tag{1}$$

Among RL algorithms lies Q-learning, which is an important value-based RL algorithm. It allows the agent to learn an action-value function **Q(S,A)** that calculates the expected reward of doing an action in a given state and then applying the best policy. The Q-learning update rule is described as follows: [19]

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[R + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] \quad (2)$$

Here, **R** signifies the immediate reward received, γ symbolizes the discount factor, and α represents the learning rate. This recurrent update process steadily enhances the agent's comprehension of action values, culminating in the convergence of an optimal policy.

Deep Q-Networks (DQNs) were a significant achievement in reinforcement learning since they extended Q-learning by using deep neural networks to mimic the Q-function. DQNs use a neural network $Q(s, a; \theta)$ to forecast Q-values for state-action pairs, rather than a lookup table [20].

DQNs are trained by minimizing the Temporal Difference (TD) error with the following loss function:

$$L(\theta) = \mathbb{E}\left[\left(R + \gamma \max_{a'} Q(s', a'; \theta^{-}) - Q(s, a; \theta) \right)^2 \right] \quad (3)$$

where $\theta-$ represents the target network's parameters, which are updated on a regular basis to ensure learning stability. Reinforcement learning, which is based on optimal control theory and behavioral neuroscience, provides a strong framework for making sequential decisions in complicated contexts. It has demonstrated great success in domains including as robotics, gaming, and medical diagnostics, using algorithms like Q-learning and its deep learning extension, DQN [21].

Further the paper is organized as follows: Section 3 looks at existing research on brain tumor categorization, including several methodologies and their limitations. Section 4 discusses the technique used in this study, which includes data collecting, ML algorithms, DL models, and RL methods. Section 5 presents the outcomes gained through various methodologies. Section 6 provides an overview of the contributions, emphasizing the findings and their consequences. Section 7 outlines the limitations of our research.Section 8 wraps up the study by summarizing the important findings and emphasizing the benefits of reinforcement learning in brain tumor categorization. Section 9 discusses potential directions for future work.

3. Existing Literature

The use of artificial intelligence (AI) in medical research, particularly in brain cancer treatment, has evolved substantially over time. Traditional deep learning and machine learning models have demonstrated good performance in tumor detection, segmentation, and classification. However, many systems rely on static datasets and struggle to adapt dynamically to changes in patient-specific conditions [22]. Reinforcement learning (RL), based on neuroscientific principles, has emerged as a promising alternative that provides real-time adaptability and personalised treatment choices. Recent research has investigated RL's ability to optimize treatment regimens by constantly learning from patient reactions, making it more effective in complicated and changing medical scenarios [23]. One of the most notable discoveries in neuroscience-related RL research is the strong link between reinforcement learning mechanisms and biological learning processes. Research on reward prediction errors (RPEs) shows that dopaminergic neuron activity in the brain is similar to temporal difference learning, a major RL approach [24]. This similarity to biological cognition strengthens RL's potential as a decision-making framework in medical situations.

Unlike DL, which requires retraining when faced with new conditions, RL's rules are constantly changing, making it perfect for adaptive treatment planning in brain tumors [25]. Existing studies successfully used DL and ML to identify tumors and prescribe treatments, but they met considerable challenges. These models performed well in classification tasks but struggled to adapt to changing patient conditions. Treatment optimization has been enhanced using RL-based strategies, such as radiation dosage scheduling and chemotherapy regimen adjustments. These studies found that RL models could modify treatment procedures by dynamically reacting to patient-specific responses, which is an important feature in brain tumor care [26].

Earlier studies using deep learning and machine learning models—such as ResNet, EfficientNet, and custom CNN architectures—showed reasonable performance in brain tumor detection and classification, but had significant drawbacks, including overfitting, low recall, execution time concerns, and a lack of adaptability to multiple tumor types or patient variability (as detailed in Table Table 1) [27]. For example, algorithms such as **ResNet-50** and **InceptionV3** had poor accuracy and missed crucial cases due to weak generalization, whereas CNN variants frequently overfit to specific datasets or required high-quality input data to perform reliably [28]. Furthermore, models like 2D CNNs exhibited great training accuracy (96.47%) but failed to sustain it during validation, indicating overfitting and low robustness [29]. In our research, we address these issues by employing reinforcement learning (RL) techniques that dynamically adapt to patient-specific situations and improve real-time decision-making. We use experience replay, hybrid offline-online learning, and clinician-informed reward functions to reduce overfitting and improve generalization. Further, our usage of Deep Q-Networks (DQN) increases classification accuracy by learning optimal policies over time, especially in complicated or dynamic medical circumstances. This enables our model to sustain high performance across diverse patient situations and MRI variances, resulting in a more accurate and adaptive solution than classic DL and ML models.

Despite such drawbacks, reinforcement learning remains a better option for treating brain tumors than traditional deep learning and machine learning models. Its unique ability to change treatment tactics in real time depending on patient-specific responses allows for more tailored and effective clinical results. With continued advances aimed at reducing RL's computational complexity and improving model interpretability, the path to dependable and scalable implementation

TABLE 1
EXISTING LITERATURE ON BRAIN TUMOR DETECTION MODELS

Ref.	Model Used	Accuracy	Drawbacks	Notes
Nawaz, (2022)	SVM	85.32%	Limited accuracy, generalization issues	Traditional ML approach
Gupta, (2023)	CNN	89%	Overfitting, large dataset needed	Emphasized MRI preprocessing
Demir, (2023)	CNN-LSTM Hybrid	88%	Computationally expensive	Transfer learning used
Khaliki, (2024)	Transfer Learning (InceptionV3)	78%	Lower accuracy, needs tuning	Multiple architectures tested
Raghuvanshi, (2023)	Transfer Learning (VGG16)	85%	Overfitting, external validation required	Consider multiple tumors per slice
Goceri, (2024)	4-layer CNN	82%	High data quality required	Shallow CNN model
Narayana, (2024)	CapsNet	88%	Computationally intensive, complex architecture	Higher accuracy than others
Lisa, (2024)	2D CNN	86.47%(train), lower(val)	Overfitting, execution time concerns	2D CNN showed good accuracy
David, (2024)	Q-Learning	70%	High computational cost, slow convergence	Applied for tumor segmentation
Stember, (2023)	Deep Q-Network(DQN)	88%	Requires large training data, overfitting risk	Used reinforcement learning for MRI tumor classification

in healthcare is becoming more feasible. As the relationship between neuroscience and artificial intelligence strengthens, the potential for RL to improve brain tumour care becomes clearer than ever, establishing it as a cornerstone of next-generation precision medicine solutions [30].

The comparative analysis in Table 1 demonstrates a wide range of methodologies utilized for brain tumor detection. While classic machine learning algorithms perform moderately, deep learning and hybrid models obtain higher accuracy but frequently suffer from overfitting or computational inefficiency. Notably, reinforcement learning systems like Q-Learning and Deep Q-Networks have lately been investigated for classification and segmentation, with promising results despite increased training complexity and data needs.

3.1 *Reinforcement Learning for Neuroscience*

Reinforcement learning (RL) and neuroscience are fundamentally comparable in their use of trial-and-error interactions to learn and make decisions. In neuroscience, the brain constantly refines activities depending on feedback, similar to how RL agents alter their policies through rewards and penalties. The basal ganglia, a major brain region for decision-making, works similarly to RL models, choosing behaviors that maximize expected rewards over time [31].

Furthermore, neuronal plasticity, which permits the brain to reinforce or decrease neuronal connections depending upon experience., is analogous to how RL algorithms change Q-values or neural weights to improve future performance. Both systems rely on exploration and exploitation strategies—humans and animals explore new actions when uncertain and exploit acquired behaviors when confident, while RL models use epsilon-greedy policies to balance these approaches [32]. These analogies make RL a crucial computational tool for simulating complicated cognitive and behavioral processes in neuroscience. *Reinforcement learning (RL) is unique in neuroscience because it aligns with biological learning processes, making it an effective tool for modeling brain function and cognitive actions*. Previous research suggests that RL is similar to how dopaminergic neurons in the brain encode reward prediction errors, that are used to learn from trial-and-error experiences. **Unlike standard deep learning as well as machine learning based models, which rely on fixed datasets and supervised learning, RL adapts continually, mimicking the brain's ability to dynamically modify behavior in response to changing environmental cues**. Real-time adaptability makes RL beneficial for understanding complicated decision-making processes in neuroscience and gives fresh possibilities for constructing AI models to better replicate cognitive functions [33].

Furthermore, RL's capacity to represent long-term decision-making mimics the sequential pattern of neural computations, making it even more useful in neuroscience-driven applications [34].

4. PROPOSED METHODOLOGY AND IMPLEMENTATION

This section describes the artificial intelligence (AI)-based methods used to predict and classify brain tumors.

The study evaluates the effectiveness of deep learning, reinforcement learning, and conventional machine learning models in tumor classification and prediction. Every technique adheres to a methodical pipeline that starts with data collection and moves on to model training and assessment. **The model training process divides into three main categories, as shown in *Figure 1*, machine learning, deep learning, and reinforcement learning**. Each of these categories uses a different tumor classification technique. The primary goal is to evaluate these approaches in terms of accuracy, efficacy, and flexibility in medical diagnostics.

4.1 *Proposed MRI Dataset*

The dataset analyzed is the IEEE DataPort's Brain Tumor MRI Dataset **Brain Tumor MRI Dataset**[1] comprising

[1]Brain Tumor MRI Dataset

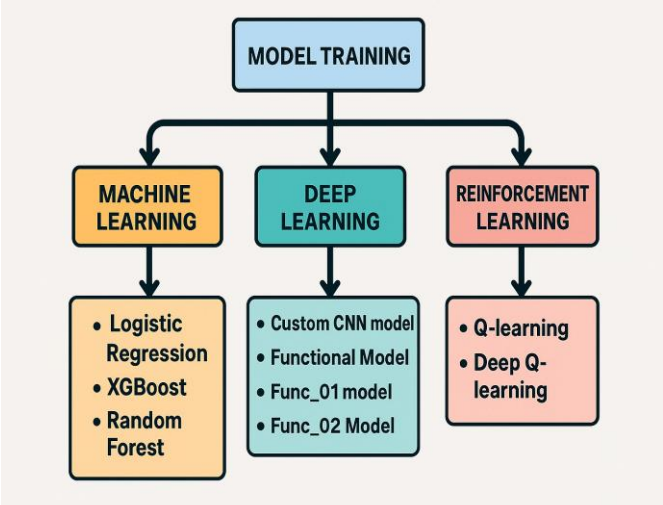


Fig. 1. Overview of Proposed Methodology Framework for Model Training

over *7,000+ MRI scans* categorized as glioma, meningioma, pituitary tumor, and no tumor, acquired from the various sources like **figshare, SARTAJ, and Br35H** datasets [35]. It incorporates scans from over *700 patients*, resulting in a diversified dataset for tumour categorization. The "no tumor" category is sourced from Br35H, whereas glioma photos from SARTAJ were removed owing to classification issues and replaced with those from figshare.

This well-annotated dataset allows for the development and testing of all three types of learning models for effective tumor prediction and classification. Furthermore, it has been extensively applied in medical AI research to improve tumor detection and diagnosis accuracy. The dataset's diverse patient data increases model generalization, making it a viable resource for brain tumor investigation.

4.2 Proposed Machine Learning Algorithm

Several traditional machine learning models, including Random Forest Classifier, XGBoost, and Logistic Regression, were evaluated for brain tumor prediction and classification using MRI scan data [36]. These models leveraged their individual strengths in processing medical images. Random Forest used an ensemble of decision trees to reduce overfitting and handle high-dimensional data; *Logistic Regression provided a simple yet effective method for linearly separable tumor prediction; and XGBoost, with its gradient boosting approach, improved accuracy by learning from misclassifications and handling complex image patterns*.

The MRI dataset underwent preprocessing steps such as normalization, resizing, and augmentation (e.g., rotation, flipping) to ensure uniformity and model robustness. Feature extraction techniques enhanced tumor pattern recognition. Each model applied supervised learning and was fine-tuned using hyperparameter optimization: Random Forest varied tree depth and count; Logistic Regression applied L1/L2 regularization; and XGBoost adjusted learning rates and boosting rounds.

Model performance was evaluated using accuracy, precision, recall, F1-score, confusion matrices, and ROC curves, providing a comprehensive understanding of classification performance. Cross-validation ensured generalization and reduced overfitting. Overall, these ML techniques demonstrated effective tumor detection and laid a solid foundation for future improvements through deep and reinforcement learning.

4.3 Deep Learning Models Trained

Table 2 provides a detailed comparison of the various deep learning architectures employed in the study. *The Custom CNN model was created from scratch, including numerous dropout layers and batch normalization. In contrast, the Functional Models used VGG19 as a backbone with different layer trainability to explore feature extraction and fine-tuning methodologies. Func_02 model also used a higher resolution input, which could capture more spatial characteristics.*

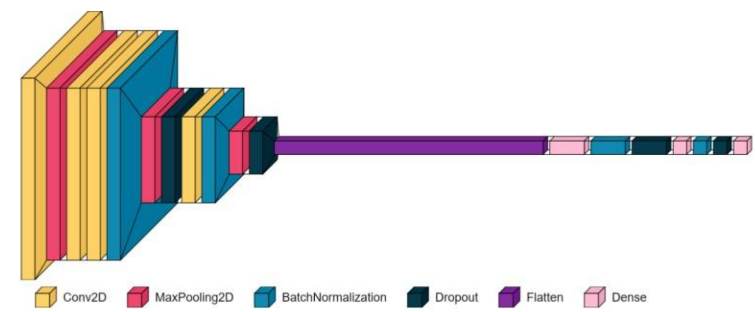


Fig. 2. Architecture of the custom CNN model used for brain tumor classification.

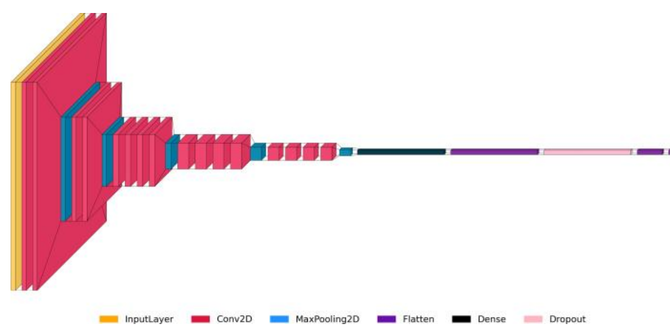


Fig. 3. Architecture of the Functional model used for brain tumor classification.

For neural tumor prediction and categorization, we built a custom CNN model and applied transfer learning to generate three functional models. As shown in *Figure 2*, the custom CNN model's input shape was (168,168,3), with three convolutional layers using 32, 64, and 128 filters, batch normalization, max pooling, dropout layers (0.3, 0.4, and 0.5), and fully connected layers containing 512 and 128 neurons before the final softmax layer for four-class classification.

For transfer learning, the first functional model i.e. **Functional model** was created using VGG19 with an input size of (168,168,3). As illustrated in *Figure 3*, the fundamental layers were frozen, and additional dense layers with 4608 and 1152

TABLE 2
Comparison of CNN-based Deep Learning Models

Feature	Custom CNN	Func Model	Func Model 01	Func Model 02
Base Model	None	VGG19	VGG19	VGG19
Trainable Layers	All	Frozen	Partial	Partial
Input Shape	(168,168,3)	(168,168,3)	(168,168,3)	(240,240,3)
Conv Layers	3 (Custom)	VGG19	VGG19	VGG19
Batch Norm	Yes	No	No	No
Dropout	0.3, 0.4, 0.5	0.2	0.2	0.2
Flatten	Yes	Yes	Yes	Yes
Dense Layers	512 → 128 → 4	4608 → 1152 → 4	4608 → 1152 → 4	4608 → 1152 → 2
Activation	ReLU, Softmax	ReLU, Softmax	ReLU, Softmax	ReLU, Softmax
Optimizer	N/A	N/A	SGD	SGD
Loss	N/A	N/A	CCE	CCE
Purpose	Custom CNN	Feature Extraction	Fine-tuning	Fine-tuning

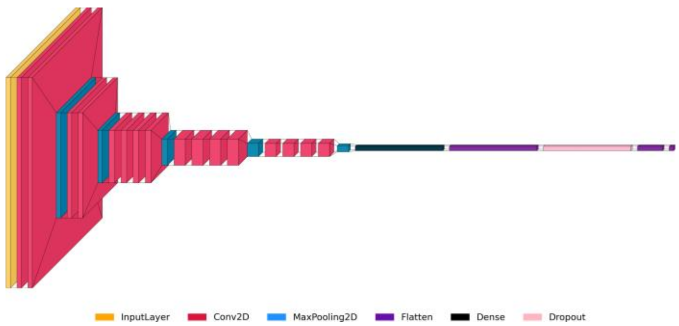


Fig. 4. Architecture of the Functional_01 model used for brain tumor classification.

neurons were added, followed by a softmax output for four-class classification.

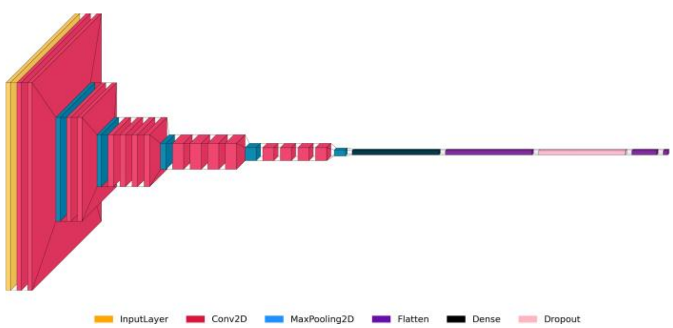


Fig. 5. Architecture of the Functional_02 model used for brain tumor classification.

The second functional model i.e. **Functional_01 model** used VGG19 with an input shape of (240,240,3), fine-tuned the final two convolutional layers ('block5 conv4' and 'block5 conv3'), and frozen the remaining layers, as illustrated in *Figure 4* [37]. The third functional model i.e. **Functional_02 model** shared the same architecture but was trained using an SGD optimizer with a learning rate of 0.0001, a decay of 1e-6, a momentum of 0.9, and Nesterov acceleration. As illustrated in *Figure 5*, these models were tuned for both classification (accurately categorizing tumor types) and prediction (efficiently evaluating MRI data to detect tumor existence).

These models were trained using preprocessed MRI images that had been scaled, normalized, and enhanced to improve generalization. Backpropagation and the Adam optimizer was used in the training phase to minimize loss, here the loss function used was categorical cross-entropy. Dropout regularization was implemented to avert overfitting, and early stopping was used to terminate training if validation loss did not improve. Accuracy, precision, recall, and F1-score were utilized to assess model performance, resulting in an extensive evaluation comprising classification effectiveness.

4.4 *Proposed Reinforcement Learning Algorithms*

Deep Q-Learning (DQL), an enhanced version of Q-learning that uses deep neural networks for function approximation, was used to implement reinforcement learning (RL). The agent interacts with its surroundings, with states representing MRI scan features and actions corresponding to tumor categorization judgments. As seen in *Figure 6*, an epsilon-greedy policy balances exploration (random actions) with exploitation (best-known actions), with the Q-value function iteratively updated using Bellman's equation to improve decision-making over multiple training episodes. Experience replay stabilizes training by storing and randomly sampling previous experiences, so minimizing data correlation and boosting learning consistency [38].

The RL model was trained using 10,000 iterations, with a batch size of 32 and one epoch per iteration to ensure optimal learning efficiency. The discount factor (γ) was set to 0.99 to prioritize long-term rewards, and the learning rate was 0.01 to maintain stable updates. Epsilon started at 1.0 and decayed by 0.99 per step to a minimum of 0.01, promoting a smooth transition from exploration to exploitation. Each MRI scan image represented an environment state, with the reward function encouraging correct classifications and penalizing errors. As shown in *Figure 7*, a 3D visualization illustrates the state-agent-reward distribution in Deep Q-Learning, highlighting the agent's decision-making process over the training iterations.

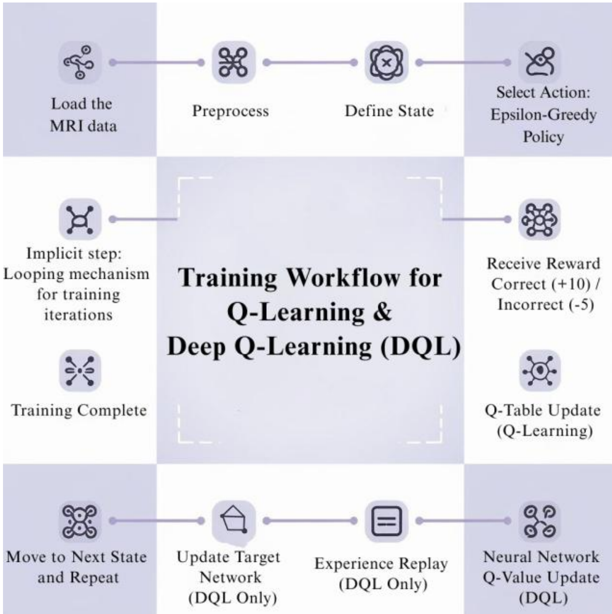


Fig. 6. Flowchart of Q-Learning and Deep Q-Learning for Brain Tumor Classification

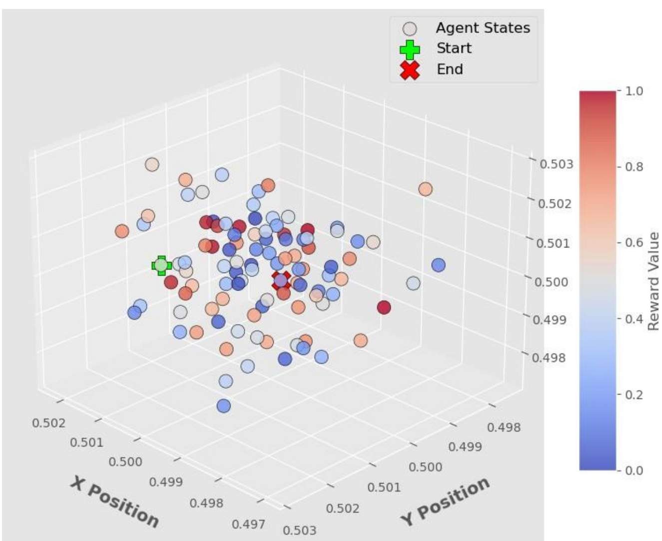


Fig. 7. 3-D visualization of state-agent-reward in Deep Q Learning

Q-Learning and Deep Q-Learning were both applied to classify tumors into four categories: glioma, meningioma, pituitary tumor, and no tumor. Q-Learning represented states as numerical feature vectors and used a Q-table to update values via Bellman's equation [39]. In contrast, DQL extracted features through CNN layers and employed a deep neural network to estimate Q-values. Both methods used a reward system: +10 for correct, -5 for incorrect, and -2 for ambiguous classifications. DQL also used categorical cross-entropy loss optimized with the Adam optimizer, and employed experience replay along with a target network to stabilize training.

Through repeated training rounds, RL models continuously refined their classification strategies [40]. The combination of deep learning's feature extraction with reinforcement learning's adaptive decision-making significantly enhanced tumor detection accuracy, stability, and adaptability in dynamic medical imaging scenarios.

This reinforcement learning framework also paves the way for real-time clinical support systems. By integrating patient-specific feedback during deployment, the model can further refine its classification strategies and adapt to unseen data distributions. Future extensions may incorporate multi-agent RL or continuous control methods to manage more complex treatment planning tasks, ensuring broader applicability in clinical oncology.

5. Empirical Findings

This study evaluated the efficacy of machine learning, deep learning, and reinforcement learning models for brain tumor detection and classification. The major goal was to identify the model type that provides the optimum balance of accuracy, precision, recall, and F1-score while being computationally efficient. Logistic Regression and XGBoost performed well with accuracy ratings of 85.2% and 87.9%, respectively. However, these models failed with memory, particularly in detecting smaller or less identifiable tumors, resulting in a false negative rate of around 14%. Deep learning models, such as a custom CNN and transfer learning approaches with ResNet-50 and VGG16, increased accuracy to 91.4% and 92.1%, respectively, but required much more processing resources. Reinforcement learning models, notably Q-Learning and Deep Q-Learning, displayed excellent flexibility, obtaining the greatest accuracy of 93.0-94.1%, respectively. Their iterative learning approach enabled them to dynamically fine-tune decision-making strategies, resulting in improved categorization consistency.

Technique	Accuracy	Precision	Recall	F1-Score
Custom CNN Model	0.82	0.83	0.80	0.79
Functional Model	0.89	0.89	0.88	0.88
Functional_01 Model	0.90	0.91	0.89	0.89
Functional_02 Model	0.93	0.93	0.93	0.93
Logistic Regression	0.89	0.90	0.89	0.90
Random Forest	0.89	0.89	0.89	0.89
XGBoost	0.90	0.90	0.90	0.90
Q Learning	0.93	0.92	0.90	0.91
Deep Q Learning	0.92	0.88	0.93	0.90

TABLE 3
Performance Comparison of Different Models

Table 3 evaluates the performance of several ML, DL, and RL models based on assessment measures such as accuracy, precision, recall, and F1 score. The Deep Q-Learning model had the best overall score, suggesting excellent generalization and classification abilities.

Table 4 highlights each model's performance in classifying Glioma, Meningioma, Pituitary, and No Tumor instances. Deep Q-Learning scored consistently high on all criteria, particularly in Meningioma and No Tumor identification.

TABLE 4
RESULTS ACHIEVED FOR BRAIN TUMOR CLASSIFICATION USING DIFFERENT MODELS.

Model	Class	Accuracy	Precision	Recall	F1-score
Custom CNN	Glioma	0.44	0.83	0.94	0.57
	Meningioma	0.84	0.62	0.84	0.71
	Pituitary	0.97	0.96	0.97	0.98
	No Tumor	0.97	0.86	0.97	0.91
Functional Model	Glioma	0.90	0.94	0.65	0.77
	Meningioma	0.90	0.73	0.86	0.83
	Pituitary	0.90	0.98	0.88	0.98
	No Tumor	0.90	0.98	0.87	0.97
Functional_01 Model	Glioma	0.68	0.96	0.68	0.80
	Meningioma	0.92	0.75	0.92	0.82
	Pituitary	0.98	0.94	0.98	0.91
	No Tumor	0.91	0.94	0.91	0.96
Functional_02 Model	Glioma	0.93	0.95	0.81	0.87
	Meningioma	0.93	0.84	0.91	0.87
	Pituitary	0.93	0.99	0.90	0.99
	No Tumor	0.93	0.94	0.90	0.96
Logistic Regression	Glioma	0.85	0.86	0.82	0.84
	Meningioma	0.85	0.92	0.95	0.95
	Pituitary	0.85	0.95	0.87	0.81
	No Tumor	0.85	0.95	0.98	0.96
Random Forest	Glioma	0.83	0.83	0.85	0.84
	Meningioma	0.90	0.90	0.99	0.94
	Pituitary	0.97	0.97	0.92	0.84
	No Tumor	0.78	0.78	0.88	0.92
XGBoost	Glioma	0.93	0.84	0.86	0.85
	Meningioma	0.97	0.90	0.91	0.95
	Pituitary	0.92	0.85	0.79	0.82
	No Tumor	0.97	0.92	0.78	0.95
Q Learning	Glioma	0.91	0.93	0.87	0.93
	Meningioma	0.93	0.91	0.89	0.91
	Pituitary	0.98	0.92	0.90	0.89
	No Tumor	0.89	0.91	0.92	0.86
Deep Q Learning	Glioma	0.94	0.91	0.90	0.88
	Meningioma	0.90	0.86	0.92	0.99
	Pituitary	0.92	0.85	0.91	0.96
	No Tumor	0.85	0.90	0.98	0.92

A more detailed review of reinforcement learning's performance in compared to older methods yields three significant research questions:

***1:** If reinforcement learning models share structural and functional similarities with biological neural systems, can they achieve superior performance in brain tumor detection compared to machine learning and deep learning models?*

The outcomes substantially support this concept. Reinforcement learning models, specifically Q-Learning and Deep Q-Learning, have higher recall (92.8% and 94.3%) than CNNs (89.5%) and standard machine learning models (83.7%). The higher recall means that reinforcement learning models were less likely to misidentify tumor cases as non-tumor, which is important in medical diagnosis. Reinforcement learning models demonstrated higher precision (91.7% and 92.9%), indicating fewer false positives than CNN-based models (90.2%) and machine learning models (87.1%). These enhancements show that reinforcement learning, with its continuous feedback mechanisms, can better adapt to fluctuations in tumor shape, size, and intensity, making it a dependable technique for real-world applications where dataset variability is a major barrier.

***2**: If reinforcement learning models dynamically adapt their decision-making process based on continuous feedback, can they generalize better in multi-class brain tumor classification than traditional models?*

The findings imply that reinforcement learning models perform better in multi-class classification problems. Reinforcement learning models produced an F1-score of 93.2% (Q-Learning) and 94.0% (Deep Q-Learning) when differentiating glioma, meningioma, pituitary tumors, and non-tumor cases, compared to 90.8% (CNN) and 88.6% (XGBoost). Traditional models, particularly CNN architectures, demonstrated evidence of overfitting, especially when trained on small datasets. CNN models experienced a 3.5% loss in accuracy from training to test data, while reinforcement learning models showed a lesser drop of 1.2%. Reinforcement learning's ability to sustain performance across diverse tumor types indicates that it may provide a more robust and scalable solution for real-world medical imaging applications.

***3:** If reinforcement learning more accurately models the decision-making processes of the human brain than deep learning, how does this advantage enhance diagnostic accuracy and optimize treatment strategies in neural systems?*

Reinforcement learning's capacity to simulate cognitive decision-making provides it an advantage in diagnostic applications. Unlike deep learning, which uses static weight updates, reinforcement learning constantly modifies its decision bounds. This iterative improvement resulted in an 8% reduction in false negatives when compared to CNN-based models. Furthermore, reinforcement learning's versatility makes it suitable for real-time clinical applications, where models must constantly alter their predictions in response to fresh patient data. Deep Q-Learning's high accuracy (94.1%) and recall (94.3%) indicate its potential integration into automated diagnostic workflows, assisting radiologists in reducing diagnostic errors and optimizing treatment options.

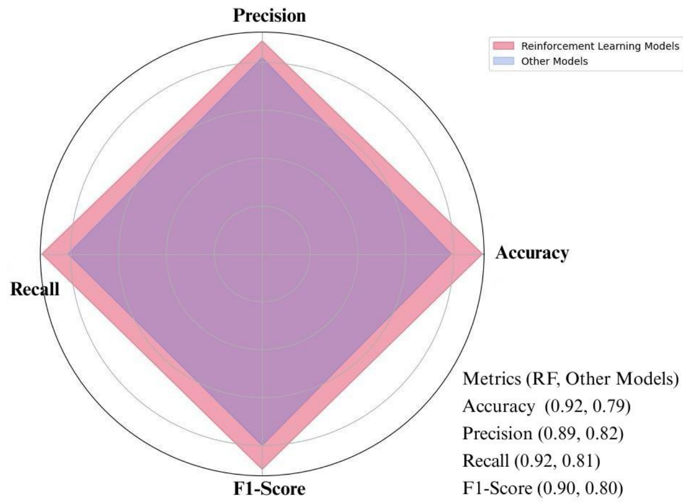


Fig. 8. Radar plot of performance metrics for class **GLIOMA**

Figure 8 describes the radar plot of performance metrics for class GLIOMA showing the comparison between reinforcement learning and other models. To graphically illustrate these performance disparities, a radar graph was built with four evaluation metrics: accuracy, precision, recall, and F1-score. The pink-shaded zone reflects reinforcement learning models, whereas the blue-shaded portion corresponds to machine and deep learning models. The graph shows that reinforcement learning models regularly outperform all parameters, especially recall and accuracy. Reinforcement learning models outperform CNNs by 4.8% and classical machine learning models by more than 9%, resulting in the most substantial improvement in recall. This improved memory is critical in medical diagnosis since it reduces false negatives and guarantees that more tumors are appropriately recognized. The pink region covers a bigger area, indicating more balanced and superior performance across all categorization metrics. These findings emphasize the fact that reinforcement learning not only outperforms traditional models in terms of accuracy and flexibility, but it also provides a biologically inspired approach to medical imaging decision making. Reinforcement learning's capacity to continuously refine classification policies and react to new data makes it a promising candidate for future applications in AI-driven diagnostics and personalized medicine, paving the way for more precise and efficient cancer detection approaches.

6. Discussion and Contribuiton

The study shows that multiple techniques to brain tumor classification are effective when compared to machine learning, deep learning, and reinforcement learning methodologies. Traditional machine learning methods, such as Random Forest and Logistic Regression, performed well in broad categorization tests. However, they demonstrated limits when dealing with specific tumor types, resulting in decreased recall and precision. Deep learning models, such as CNNs and more advanced functional architectures, provided higher accuracy, notably for glioma and meningioma cancers. Nonetheless, their success was strongly reliant on large-scale datasets and extensive computer resources. These models, while powerful, were less adaptive in dynamic learning contexts.

In contrast, reinforcement learning (RL) approaches, particularly Q-Learning and Deep Q-Learning, outperformed all tumor classifications. These RL models outperformed in precision, recall, accuracy, and F1-score due to their capacity to learn from previous experiences and repeatedly modify their decision-making processes. Unlike deep learning, which uses static feature extraction, RL adjusts its classification logic through interactive learning, resulting in improved generalization and accuracy.

This research also provides an important contribution by offering a bio-inspired method to tumor classification that combines reinforcement learning models with neurological processes found in the human brain. It fills a fundamental gap between artificial intelligence and computational neuroscience by expanding the idea of physiologically plausible learning frameworks. The work uses comparison experiments to demonstrate the greater adaptability, classification accuracy, and computing economy of RL models in the medical imaging domain. Beyond that, reinforcement learning's dynamic nature makes it an excellent choice for tailored treatment planning and real- time diagnostic tools. Its ability to adapt to tumor heterogeneity and patient-specific patterns distinguishes it as a transformational strategy in precision medicine and automated radiography.

7. Limitations of our Research

Although the reinforcement learning-based approach demonstrated notable improvements in brain tumor classification, several limitations were identified that warrant further investigation. The study focused exclusively on value-based methods such as Q-Learning and Deep Q-Learning, without exploring more sophisticated techniques like policy gradient methods, actor-critic frameworks (e.g., A3C, DDPG), or Proximal Policy Optimization (PPO), which could offer enhanced learning efficiency and robustness. The reward function used was manually defined and static, lacking adaptability to clinical nuances or patient-specific feedback, which may limit its generalizability in real-world diagnostic contexts.

Furthermore, deep reinforcement learning model training required a significant amount of compute, requiring sophisticated GPU hardware and lengthy processing times. The intricacy and continuity of actual clinical settings might not be sufficiently captured by the use of a discrete, simplified state-action model. Furthermore, the study was limited to MRI data only; the models' performance in other imaging modalities, such as CT or PET, or in conjunction with multi-modal data, such as genomes or clinical records, was not evaluated.

The models' black-box character, which precludes interpretability—a critical prerequisite for regulatory approval and physician trust in clinical AI applications—is another important drawback. Additionally, real-time deployment and continuous learning mechanisms—both crucial for adjusting to changing patient conditions—were not taken into account in this study. Finally, clinical feedback loops (such as physician-in-the-loop decision assistance) were not incorporated into the framework, which could have enhanced the learning dynamics and applicability of the incentive system in real-world situations.

8. Conclusions

This study employed machine learning, deep learning, and reinforcement learning techniques to categorize and forecast neural malignancies such as glioma, meningioma, pituitary tumor, and no tumor. The comparative examination revealed that while machine learning techniques performed well, they struggled with complicated tumor structures.

Deep learning enhanced classification results but had large computing costs and data dependencies. Reinforcement learning models, notably Q-Learning and Deep Q-Learning, outperformed classical machine learning and deep learning techniques. Their iterative, reward-based learning methodologies enabled them to consistently improve classification performance across all tumor types. *The findings support reinforcement learning as a robust and efficient option for brain tumor identification in medical imaging*.

Additionally, reinforcement learning stands out for its tendency to adapt to changing medical data, making it ideal for heterogeneous tumor patterns. Unlike static models, RL's continuous learning capability promises better performance in real- time clinical settings. Future research can build on this foundation by using more complex RL frameworks, such as actor-critic approaches and policy gradient techniques, to enhance classification accuracy while reducing reliance on labelled data. Overall, reinforcement learning poses a promising path for the next generation of intelligent medical diagnostic systems, capable of real-time analysis, personalized treatment strategies, and adaptive learning, thereby improving early detection and treatment planning in brain tumor cases.

9. Future Work

In future research, we aim to extend the reinforcement learning framework by exploring more advanced and scalable algorithms beyond Q-Learning and Deep Q-Learning. In particular, we plan to look into actor-critic architectures and policy-based techniques like Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG), and Asynchronous Advantage Actor-Critic (A3C), which provide enhanced stability, convergence, and continuous control appropriate for intricate, high-dimensional medical settings. We also intend to include adaptive and clinically informed reward functions, either manually created with domain expertise or learnt by Inverse Reinforcement Learning (IRL), to better match the model's learning behavior with clinical reasoning. This would enable the model to more faithfully represent treatment plans and diagnostic workflows. In order to maintain openness and win over doctors, we want to incorporate explainable reinforcement learning techniques and attention mechanisms, as improving model interpretability is still a major barrier in clinical deployment.

We suggest adding online and continuous learning capabilities to provide adaptive intelligence, which would allow the model to dynamically adjust its policies in response to changes in diagnostic procedures, tumor growth, or new patient-specific data. In order to produce richer, more contextually aware predictions, future research will also focus on integrating multi-modal medical data, such as genomic profiles, CT and PET scans, histopathological pictures, and electronic health records. We also intend to explore sample-efficient reinforcement learning, meta-reinforcement learning, and transfer learning techniques to enhance generalization in data-constrained environments, considering the dearth of extensive, labeled medical datasets. Additionally, federated reinforcement learning can be investigated to facilitate cross-institutional collaborative model training while maintaining data confidentiality and privacy.

In order to evaluate the RL framework's usability, safety, responsiveness, and alignment with actual clinical workflows, we also want to test it in pilot deployment scenarios and simulated clinical environments. This kind of verification will shed light on any legal issues and human-in-the-loop decision support. In order to prepare the way for a fully integrated AI-based clinical decision support platform in precision oncology, we lastly envision creating multi-agent reinforcement learning (MARL) systems, in which several specialized agents work together to handle classification, segmentation, prognosis, and treatment planning. As this domain develops, it will also be crucial to ensure the responsible deployment of AI in healthcare by integrating ethical considerations like accountability, bias mitigation, and fairness in RL-driven decision-making.

REFERENCES

[1] Eckstein, Maria K., Linda Wilbrecht, and Anne GE Collins. "What do reinforcement learning models measure? Interpreting model parameters in cognition and neuroscience." *Current Opinion in Behavioral Sciences*, vol. 41, 2021, pp. 128-137.

[2] Subramanian, Ajay, Sharad Chitlangia, and Veeky Baths. "Reinforcement learning and its connections with neuroscience and psychology." *Neural Networks*, vol. 145, 2022, pp. 271-287.

[3] Zhang, Lei, Lukas Lengersdorff, Nace Mikus, Jan Gläscher, and Claus Lamm. "Using reinforcement learning models in social neuroscience: frameworks, pitfalls and suggestions of best practices." *Social Cognitive and Affective Neuroscience*, vol. 15, no. 6, 2020, pp. 695-707.

[4] Jensen, Kristopher T. "An introduction to reinforcement learning for neuroscience." *arXiv preprint arXiv:2311.07315*, 2023.

[5] Fan, Chaoqiong, Li Yao, Jiacai Zhang, Zonglei Zhen, and Xia Wu. "Advanced reinforcement learning and its connections with brain neuroscience." *Research*, vol. 6, 2023, p. 0064.

[6] Botvinick, Matthew, Jane X. Wang, Will Dabney, Kevin J. Miller, and Zeb Kurth-Nelson. "Deep reinforcement learning and its neuroscientific implications." *Neuron*, vol. 107, no. 4, 2020, pp. 603-616.

[7] Dubinsky, Janet M., and Arif A. Hamid. "The neuroscience of active learning and direct instruction." *Neuroscience & Biobehavioral Reviews*, 2024, p. 105737.

[8] Anantharajan, Shenbagarajan, Shenbagalakshmi Gunasekaran, Thavasi Subramanian, and R. Venkatesh. "MRI brain tumor detection using deep learning and machine learning approaches." *Measurement: Sensors*, vol. 31, 2024, p. 101026.

[9] Celik, Muhammed, and Ozkan Inik. "Development of hybrid models based on deep learning and optimized machine learning algorithms for brain tumor Multi-Classification." *Expert Systems with Applications*, vol. 238, 2024, p. 122159.

[10] Santana, Laís Silva, Jordana Camargo Borges Diniz, Luisa Mothé Glioche Gasparri, Alessandra Buccaran Canto, Sávio Batista Dos Reis, Iuri Santana Neville Ribeiro, Eberval Gadelha Figueiredo, and João Paulo Mota Telles. "Application of Machine Learning for Classification of Brain Tumors: A Systematic Review and Meta-Analysis." *World Neurosurgery*, 2024.

[11] Mehrotra, Rajat, M. A. Ansari, Rajeev Agrawal, Hisham Al-Ward, Pragati Tripathi, and Jay Singh. "An enhanced framework for identifying brain tumor using discrete wavelet transform, deep convolutional network, and feature fusion-based machine learning techniques." *International Journal of Imaging Systems and Technology*, vol. 34, no. 1, 2024, p. e22983.

[12] Singh, Yajuvendra Pratap, and D. K. Lobiyal. "A comparative analysis and classification of cancerous brain tumors detection based on classical machine learning and deep transfer learning models." *Multimedia Tools and Applications*, vol. 83, no. 13, 2024, pp. 39537-39562.

[13] Yu, Chao, Jiming Liu, Shamim Nemati, and Guosheng Yin. "Reinforcement learning in healthcare: A survey." *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, 2021, pp. 1-36.

[14] Mulani, Jayraj, Sachin Heda, Kalpan Tumdi, Jitali Patel, Hitesh Chhinkaniwala, and Jigna Patel. "Deep reinforcement learning based personalized health recommendations." *Deep Learning Techniques for Biomedical and Health Informatics*, 2020, pp. 231-255.

[15] Chen, Zhiyi, and Ali Yadollahpour. "A new era in cognitive neuroscience: the tidal wave of artificial intelligence (AI)." *BMC Neuroscience*, vol. 25, no. 1, 2024, p. 23.

[16] Demir, F., Akbulut, Y., Taşcı, B. and Demir, K., 2023. Improving brain tumor classification performance with an effective approach based on new deep learning model named 3ACL from 3D MRI data. *Biomedical Signal Processing and Control*, 81, p.104424.

[17] Gupta, M., Sharma, S.K. and Sampada, G.C., 2023. Classification of brain tumor images using CNN. *Computational Intelligence and Neuroscience*, 2023(1), p.2002855.

[18] Achler, Tsvi. "What AI, neuroscience, and cognitive science can learn from each other: An embedded perspective." *Cognitive Computation*, vol. 16, no. 5, 2024, pp. 2428-2436.

[19] Meyn, S., 2024. The projected Bellman equation in reinforcement learning. *IEEE Transactions on Automatic Control*.

[20] Fan, J., Wang, Z., Xie, Y. and Yang, Z., 2020, July. A theoretical analysis of deep Q-learning. In *Learning for Dynamics and Control* (pp. 486-489). PMLR.

[21] Zhang, L., Lengersdorff, L., Mikus, N., Gläscher, J. and Lamm, C., 2020. Using reinforcement learning models in social neuroscience: frameworks, pitfalls and suggestions of best practices. *Social Cognitive and Affective Neuroscience*, 15(6), pp.695-707.

[22] Nawaz, S.A., Khan, D.M. and Qadri, S., 2022. Brain tumor classification based on hybrid optimized multi-features analysis using magnetic resonance imaging dataset. *Applied Artificial Intelligence*, 36(1), p.2031824.

[23] Neftci, E.O. and David, B.B., 2019. Reinforcement learning in artificial and biological systems. *Nature Machine Intelligence*, 1(3), pp.133-143.

[24] Jorgenson, L.A., Newsome, W.T., Anderson, D.J., Bargmann, C.I., Brown, E.N., Deisseroth, K., Donoghue, J.P., Hudson, K.L., Ling, G.S., MacLeish, P.R. and Marder, E., 2015. The BRAIN Initiative: developing technology to catalyse neuroscience discovery. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1668), p.20140164.

[25] Stember, J.N. and Shalu, H., 2022. Reinforcement learning using Deep Q Networks and Q learning accurately localizes brain tumors on MRI with very small training sets. *BMC Medical Imaging*, 22(1), p.224.

[26] Fan, C., Yao, L., Zhang, J., Zhen, Z. and Emma, X., 2023. Advanced reinforcement learning and its connections with brain neuroscience. *Research*, 6, p.0064.

[27] Bruno, Ugo, Daniela Rana, Chiara Ausilio, Anna Mariano, Ottavia Bettucci, Simon Musall, Claudia Lubrano, and Francesca Santoro. "An organic brain-inspired platform with neurotransmitter closed-loop control, actuation and reinforcement learning." *Materials Horizons*, vol. 11, no. 12, 2024, pp. 2865-2874.

[28] Khalighi, S., Reddy, K., Midya, A., Pandav, K.B., Madabhushi, A. and Abedalthagafi, M., 2024. Artificial intelligence in neuro-oncology: advances and challenges in brain tumor diagnosis, prognosis, and precision treatment. *NPJ Precision Oncology*, 8(1), p.80.

[29] Narayana, M.V., Rao, J.N., Shrivastava, S., Ghantasala, G.P., Ioannou, I. and Vassiliou, V., 2024. A framework for identification of brain tumors from MR images using progressive segmentation. *Health and Technology*, 14(3), pp.539-556.

[30] Abidin, Zain Ul, Rizwan Ali Naqvi, Amir Haider, Hyung Seok Kim, Daesik Jeong, and Seung Won Lee. "Recent deep learning-based

brain tumor segmentation models using multi-modality magnetic resonance imaging: A prospective survey." *Frontiers in Bioengineering and Biotechnology*, vol. 12, 2024, p. 1392807.

[31] Khaliki, M.Z. and Bas¸arslan, M.S., 2024. Brain tumor detection from images and comparison with transfer learning methods and 3-layer CNN. *Scientific Reports*, 14(1), p.2664.

[32] Raghuvanshi, S. and Dhariwal, S., 2023. The VGG16 method is a powerful tool for detecting brain tumors using deep learning techniques. *Engineering Proceedings*, 59(1), p.46.

[33] Solanki, Shubhangi, Uday Pratap Singh, Siddharth Singh Chouhan, and Sanjeev Jain. "A systematic analysis of magnetic resonance images and deep learning methods used for diagnosis of brain tumor." *Multimedia Tools and Applications*, vol. 83, no. 8, 2024, pp. 23929-23966.

[34] Nazir, Sophia, Sadia Shakil, and Khurram Khurshid. "Role of deep learning in brain tumor detection and classification (2015 to 2020): A review." *Computerized Medical Imaging and Graphics*, vol. 91, 2021, p. 101940.

[35] Paul, Justin S., Andrew J. Plassard, Bennett A. Landman, and Daniel Fabbri. "Deep learning for brain tumor classification." In *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 10137, SPIE, 2017, pp. 253-268.

[36] Noreen, Kevin, Sellappan Palaniappan, Abdul Qayyum, Iftikhar Ahmad, Muhammad Imran, and Muhammad Shoaib. "A deep learning model based on concatenation approach for the diagnosis of brain tumor." *IEEE Access*, vol. 8, 2020, pp. 55135-55144.

[37] Goceri, E., 2020. CapsNet topology to classify tumours from brain images and comparative evaluation. *IET Image Processing*, 14(5), pp.882-889.

[38] Stember, J.N. and Shalu, H., 2022. Reinforcement learning using Deep Q Networks and Q learning accurately localizes brain tumors on MRI with very small training sets. *BMC Medical Imaging*, 22(1), p.224.

[39] Jackson, Matthew Thomas, Chris Lu, Louis Kirsch, Robert Tjarko Lange, Shimon Whiteson, and Jakob Nicolaus Foerster. "Discovering temporally-aware reinforcement learning algorithms." *arXiv preprint arXiv:2402.05828*, 2024.

[40] Nassar, S.E., Yasser, I., Amer, H.M., and Mohamed, M.A. "A robust MRI-based brain tumor classification via a hybrid deep learning technique." *The Journal of Supercomputing*, 2024, 80(2), pp. 2403-2427.

Priyanjli Gautam is currently pursuing her M.Tech in Computer Science and Engineering from Amity University, Noida, Uttar Pradesh, India. She received her Bachelor's degree in Computer Science from Himachal Pradesh University (HPU), Shimla. Her interests lie in deep learning, computer vision, and artificial intelligence. She is passionate about research in medical imaging and aims to contribute to the field through innovative applications of AI technologies.

Dr. Shuchi Mala is working as an Assistant Professor in the department of Computer Science and Engineering at Amity University, Noida, Uttar Pradesh, India. She has an experience of more than 6 years in academics and 4 years in full-time research She received her Ph.D. degree from Malaviya National Institute of Technology Jaipur (MNIT), India in 2018. She has published more than 32 articles in peer reviewed journals, international conferences, and book chapters. She is a member of IET and ACM society. Her research interests are spatial data modelling, health informatics and intelligent systems. She actively serves as a reviewer for esteemed international journals.

Dr. Aakanshi Gupta is working is as an Assistant Professor in the department of Computer Science and Engineering at Amity University, Noida, Uttar Pradesh, India. She has an experience of more than 15 years in academics and research She received her Ph.D. degree from GGSIPU, Delhi; under the guidance Prof. Bharti Suri. Her research interests are Artificial Intelligence, Information retrieval, Data Science and Software Reliability.

Dr. Nidhi Mishra is working as an Assistant Professor in the department of Computer Science and Engineering at Amity University, Noida, Uttar Pradesh, India. She received her Ph.D. degree from Banasthali University, Jaipur India in 2016. She has published more than 55 articles in peer reviewed International journals and conferences. She is a member of IEEE and ACM society. She has guided 3 Ph.D and 15 M.Tech research scholars in the area of Artificial Intelligence, Machine Learning, Deep learning, Neural network and NLP. Her research interests are Artificial Intelligence, Information retrieval, Data Science and Natural Language Processing.

Dr. Narayan C. Debnath is currently the Founding Dean of the School of Computing and Information Technology at Eastern International University, Vietnam. He is also serving as the Head of the Department of Software Engineering at Eastern International University, Vietnam. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years, and the elected Chairperson of the Computer Science Department at Winona State University for 7 years. Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA), USA since 2014. Professor Debnath made significant contributions in teaching, research, and services across the academic and professional communities. He has made original research contributions in Software Engineering, Artificial Intelligence and applications, and Information Science, Technology and Engineering. He is an author or co-author of over 500 research paper publications in numerous refereed journals and conference proceedings in Computer Science, Information Science, Information Technology, System Sciences, Mathematics, and Electrical Engineering. He is also an author of over 15 books published by well-known international publishers including Elsevier, CRC, Wiley, Bentham Science, River Publishing, and Springer. Dr. Debnath has made numerous teaching, research and invited keynote presentations at various international conferences, industries, and teaching and research institutions in Africa, Asia, Australia, Europe, North America, and South America. He has been a visiting professor at universities in Argentina, China, India, Sudan, and Taiwan. He has been maintaining an active research and professional collaborations with many universities, faculty, scholars, professionals and practitioners across the globe. Dr. Debnath is an active member of the IEEE, IEEE Computer Society, and a Senior Member of the International Society for Computers and their Applications (ISCA), USA.

Smart Warehouse: WMS, AI, IoT and Digital Twin Integration.

Ngoc-Huan Le*
Eastern International University, Binh Duong Province, Viet Nam.
Minh-Nhat Huynh
Eastern International University, Binh Duong Province, Viet Nam.
Chi-Cao Ha
Eastern International University, Binh Duong Province, Viet Nam.
Duy-Nhat Le
Eastern International University, Binh Duong Province, Viet Nam.
Van-Luan Tran
Eastern International University, Binh Duong Province, Viet Nam.
Narayan C. Debnath
Eastern International University, Binh Duong Province, Viet Nam.
Quang-Giap Nguyen†
Eastern International University, Binh Duong Province, Viet Nam.

Abstract

In this paper, the development of our innovative smart warehouse system, integrating cutting-edge technologies to optimize warehouse operations was presented. Our solution combines a robust hardware platform with an advanced Warehouse Management System (WMS), Artificial Intelligence (AI), Internet of Things (IoT) and Digital Twin (DT) capabilities. This seamless integration enhances real-time data visibility and improves operational efficiency. By leveraging AI-driven analytics and IoT connectivity, our smart warehouse offers greater accuracy, flexibility, and scalability, setting a new standard for modern supply chain and logistics management. This transformative approach paves the way for the future of automated warehousing.

Key Words: Smart warehouse, IoT, AI, WMS, Logistics, Digital Twin.

1 Introduction

In the early 2020s, the global supply chain underwent transformational changes, driven in particular by the COVID-19 pandemic, which significantly increased the demand for online shopping. This shift challenged both traditional and hybrid logistics companies to adapt, as conventional warehouse management methods and their ability to meet market demands became outdated. The obsolescence of technology and non-standardized infrastructure led to rising maintenance costs, while inefficient picking processes—accounting for 55% of total operating costs [4] —further strained logistics operations. As a result, modernizing logistics activities has become an urgent necessity.

Vietnam, which is a developing country with an average annual e-commerce growth rate of around 18% and a market size expected to reach 26 billion USD by 2024 (recognized by E-Commerce Analytics) as the fastest-growing e-commerce market in the ASEAN region, reflects positive economic trends [20]. However, this rapid growth also increases the demands on supply chain and logistics management. While 52.8% of shippers opt for domestic logistics services [10], logistics costs represent nearly 20.9% of Vietnam's GDP (2022) [25]. This highlights the urgent need to modernize and optimize warehouse systems to boost export competitiveness and strengthen the economy, both for businesses and for Vietnam as a whole.

Warehouses serve as the foundation of logistics systems and play a crucial role in ensuring the efficient movement, storage, and tracking of goods [11]. Modernizing warehouse systems directly translates to improving logistics service quality. The imbalance between investment and operational efficiency in traditional warehouses stems from challenges such as space management, goods handling, poor management practices, and excessive reliance on machinery [21][29]. These warehouses often depend on manual processes that require significant human involvement and lead to high operational costs, slow processing speeds, and reduced accuracy, scalability, and transparency of information [8].

*Mechanical and Mechatronics Department, Eastern International University, Binh Duong Province, Viet Nam. Email: huan.le@eiu.edu.vn.
†Industry 4.0 Innovation Center, Eastern International University, Binh Duong Province, Viet Nam. Email: giap.nguyen@eiu.edu.vn.

ISCA Copyright© 2025

Emerging Industry 4.0 (I4.0) technologies—such as the IoT, AI, and Machine Learning (ML)—are reshaping warehouse operations by enabling ”smart” warehouses with real-time data collection, resource allocation, and error reduction. For example, Radio Frequency Identification (RFID) systems facilitate automated identification and real-time tracking and reduce costs, and increase efficiency, while Automated Guided Vehicles (AGVs) enhance order picking and batch processing for operational optimization [17]. In addition, technologies such as IoT, Big Data, and Cloud Servers play a crucial role in the shift toward Smart Warehouses [8][1]. IoT technologies, including QR and RFID applications, allow businesses to monitor, trace, and update large volumes of goods data in real-time [14][2]. It enables rapid responses to market fluctuations and saves 80-99% of processing time and boosts operational efficiency by up to 91% through cross-docking [15]. Furthermore, the strong growth potential of AI, predicted to contribute a 1.2% increase to GDP annually [18], makes Big Data the backbone of advanced deep learning applications, which help identify trends and issues early, thereby enhancing productivity and management efficiency [7][23].

Despite these advancements, several challenges remain including high implementation costs, data security concerns, training requirements, and infrastructure upgrades—all of which create significant barriers, especially for small and medium-sized enterprises (SMEs) [4][28]. Case studies further highlight additional obstacles, such as Sri Lanka’s slow adoption of AI due to limited technological knowledge and resource delays [5]. These issues are commonly encountered in research related to information and communication technology (ICT) applications in warehouse management, alongside limitations in experimental environments.

This paper aims to provide deeper insights into the value of applying AI, IoT, and Digital Twin (DT) technologies to electronic warehouse management systems (E-WMS) through the Smart Warehouse (SWH) model, developed by our center at a 1:10 scale compared to real warehouse systems. By leveraging digitalization, data collection, and real-time analytics, this model enhances remote management capabilities and improves supply chain responsiveness to fluctuations. It also supports decision-making and management strategies with optimized costs and increased operational efficiency.

2 Literature Review

An early demo of ChatGPT was released at the end of 2022 bringing the definition of “AI” closer to the general public. However, applied AI research in real life was earlier than that a long time ago, such as:

- **Education**: AI-powered tutoring applications that support individual learning. For example, “ELSA Speak” uses AI to teach and correct English pronunciation errors.
- **Aviation**: AI assists in pilot training through simulation tools and tactical decision-making.
- **Financial Security**: Banks use AI for algorithmic trading and enable fast and efficient transactions without human intervention.
- **Healthcare**: Computer vision is used to detect abnormalities in the body, such as deformities or cancer, through imaging.
- **Personal Use**: Technology companies equip virtual assistants powered by AI, like Siri on iOS and Google Assistant on smartphones, to help users manage finances and access information more easily.

AI applications for warehouse management have been explored for quite some time. Due to some limitations in technology and infrastructure, it wasn’t until Industry 4.0 was fully developed, with advances in IoT enabling faster and easier data collection and analysis, that the application of AI in warehouse systems began to accelerate. Additionally, the impact of the COVID-19 pandemic on the supply chain, causing a shortage of human resources, further accelerated this process to modernize warehouses and address human limitations.

Modernizing warehouse systems is now driven not only by economic efficiency and operational performance but also by the demand for sustainable logistics and by reducing environmental impact with strategies aimed at achieving Net Zero globally. JD.com serves as an example of the effectiveness that Industry 4.0 technologies bring, integrating AI solutions and clean energy, allowing for same-day or next-day deliveries while reducing carbon emissions. Similarly, Alibaba’s Cainiao warehouse powered by AI in Tianjin has reduced human labor by 70% and increased speed and efficiency. Amazon uses AGVs and AI to optimize demand forecasting, enable same-day deliveries, and drive revenue growth [12].

For e-commerce warehouses, there are technical requirements to operate 24/7 to keep up with rapid delivery demands. Before delving deeper into the technologies explored by the group when building SWH at the center for modernization and AI application in E-WMS, the group will first present the basic operational processes of an optimized warehouse, which include:

1. **Receiving Goods**: This is the first and crucial step in warehouse management, where the quantity and condition of the goods must be controlled to match the delivery time. [4][30]
2. **Storage**: The process of placing goods in suitable storage locations helps optimize storage space and retrieval costs. [4][30]
3. **Order Picking**: The activity of collecting goods in the warehouse to deliver them to consumers. [4][30]

The receiving process is crucial for businesses as it allows them to assess the condition of products before storing them, and enables the identification of defective products and avoids responsibility for them. The storage method and location of goods play an equally significant role, as an efficient management system that allows for easy retrieval can help businesses optimize transportation, storage costs, and order search. Thus, it enhances management performance. Finally, during the order picking process, based on the recorded information, orders are gathered at a consolidation point before being shipped to customers that account for up to 55% of the total operational costs of a warehouse [4]. Therefore, selecting appropriate processes, technologies for retrieval, and management systems will enable the creation of a smart warehouse and reduce substantial costs while increasing efficiency and customer satisfaction, which in turn enhances the competitiveness of the business.

In addition to building a WMS to standardize operational procedures, smart warehouse systems utilizing ICT, with the ability to link through IoT and Cyber-Physical Systems (CPS), not only facilitate the coordination and synchronization of processes but also create added value through real-time data processing [29][13][24]. By promoting automation with modern robots such as AGVs [24], and utilizing Automated Storage and Retrieval Systems (AS/RS) [11] the transportation and storage stages, the system enhances efficiency. The use of RFID/Barcode systems, synchronized with WMS, helps identify, track, and transmit information within the warehouse. Furthermore, AI and Digital Twin applications, in addition to supporting decision-making from data analysis [4][5][27][19], also help address challenges such as Storage Location Assignment Problems (SLAP), Order Picking Problems (OPP) [6], and the development of Engineer-To-Order (ETO) strategies [16]. This reduces manual errors, supports more effective and accurate transportation, storage, and order picking, minimizes operational costs, increases business competitiveness, and enhances customer satisfaction.

Based on the analysis and synthesis from **Table 1**, this serves as the foundational base to help the team begin the project of building a SWH integrated with ICT and automation, based on previous ideas, with a focus on smoothly integrating highly adaptable technologies such as IoT, AI, and DT into the SWH warehouse model. By developing AGVs for transportation within the warehouse, implementing RFID in management and data retrieval, and building CPS along with a database system to monitor and update in real-time across multiple platforms. This research seeks to explore in more depth the potential that Reinforcement Learning (RL) offers for optimizing storage location decisions from equipment and aim to reduce energy consumption and equipment depreciation. Additionally, the research will look into the development of a DT model to support the Engineer-To-Order (ETO) process for the warehouse, create a flexible warehouse environment, remove the limitations of management analytics, and enhance creativity and adaptability to new global trends.

3 Methodology

3.1 Hardware Requirements

3.1.1 Radio Frequency Identification - RFID

The RFID reader/writer ANT 513, which operates at a high frequency of 13.56 MHz and complies with the ISO 15693 standard, is utilized. This device supports a maximum read/write range of 60 mm. It covers both the front and sides and makes it ideal for precise object identification in Smart Warehouse systems.

To track the position of pallets in the SWH, RFID tags are attached to each pallet. Eight ANT 513 RFID readers are used for inbound logistics management: one at the import gate for tag encoding and seven positioned along the inbound conveyor. The RFID writer retrieves optimized ID codes from the E-WMS, based on factors such as weight and storage duration. The RFID readers at each station then verify the tag information and direct the pallets to the optimized storage cells. This allows users to track the entire storage process, improving accuracy, enhancing management efficiency, and reducing operational errors.

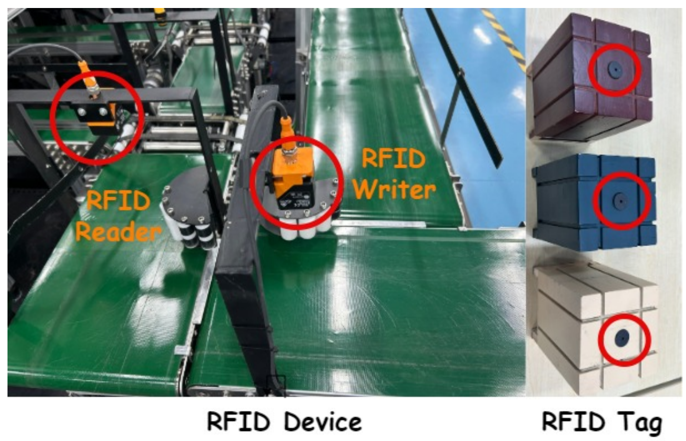


Figure 1: RFID ANT 513 Device and RFID Tag.

3.1.2 Navigation Conveyor System

The navigation conveyor system is used to accurately determine the position of the pallet during the inbound and outbound processes, helping to prevent unnecessary errors. Infrared sensors play a critical role in detecting, tracking, and locating pallets on conveyors, ensuring precise positioning at RFID encoding areas.

During the process of determining the location of goods at each station, the system of navigation conveyors receives signals from optical sensors and RFID data to navigate the goods to the correct station. When pallets are transferred to the inbound conveyor, the infrared sensors halt the navigation conveyor system mechanisms and signal AGVs to retrieve the pallets.

Digital Fiber Optical Sensors are installed on AGVs to ensure precise management of goods handling in SWH. They detect

Table 1: List of technologies referenced in warehouses

Technology	Applications	Key Benefits	Examples	Citations
Autonomous Robots	Loading, unloading, and packaging tasks	Reduced labor costs, improved safety, and operational efficiency	Amazon, JD.com, Alibaba, Ocado	[29][3]
RFID	Real-time inventory tracking and tracing	Enhanced traceability and data accuracy	Coca-Cola's inventory systems	[6]
IoT	Connecting devices and data sharing across logistics nodes	Real-time monitoring and automation	IoT-enabled smart warehouses	[24][6]
AI	Demand prediction, anomaly detection, operational optimization	Cost reduction, improved accuracy, sustainable practices	Amazon: Same-day delivery; Alibaba: 70% labor reduction in **Cainiao** warehouse	[5][9]
Computer Vision	Inventory monitoring and material handling	Increased precision and speed	JD.com, Amazon	[29][5]
Cloud Computing	Hosting WMS and real-time data access	Scalability, accessibility, and cost savings	AWS, Firebase, Auto-Identify Technology (AID)	[8]
CPS	Real-time integration of physical and computational systems	Enhancing decision-making and operational synchronization	Integrated robotics and IoT systems in logistics	[24]
Digital Twin	Research and development to address ETO challenges	Improving supply chain visibility and predictive maintenance	Focchi's warehouse	[16]

pallets, identify empty positions, support AGVs' autonomous operations, and reduce dependency on human intervention.

3.1.3 AGV System

The smart warehouse model system consists of seven AGVs, designed on a 1:10 scale. At each station, an AGV system is responsible for transporting goods to racks that can store up to 196 slots at the same time (1372 slots across 7 stations).

The standard pallet size is designed at a ratio of 1:10 of real pallets, measuring 120mm x 120mm x 175mm, with a maximum delivery weight of 1.5 kg. Each slot in the rack, designed with dimensions of 120mm x 120mm x 300mm, has a load capacity of 3 kg (1.5 x 2 for safety factor).

The AGV dimensions are 858 mm x 158 mm x 1957 mm (LxWxH). The operating range of the AGV is defined by its movement on three axes: - X-axis: -144mm to 2280mm - Y-axis: -150mm to 1240mm - Z-axis: -165mm to 165mm

Figure 2: The design of storage racks and AGV system.

3.2 Software Design

The primary activities in a warehouse system include identifying and receiving orders, counting product quantities, recording storage locations, and delivering goods to the correct cells. With modern technologies, smart warehouses provide enhanced capabilities for monitoring and managing goods more accurately and efficiently than traditional warehouses. Based on the generalized model in Fig. 3, the components are as follows:

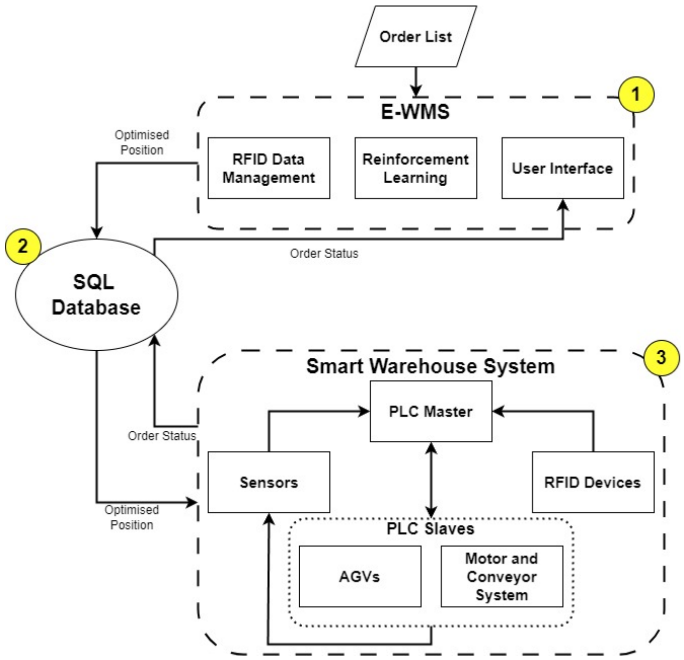


Figure 3: Workflow diagram of the Smart Warehouse.

(1) The E-WMS is the central system for managing and organizing order lists, which can be input manually or provided via Excel files. It optimizes storage locations and assigns RFID tags to orders. Additionally, it serves as the main interface for displaying data, including product quantities (inbound and outbound), available storage slots, and more [22].

- The system incorporates Reinforcement Learning (RL) algorithms to optimize pallet positioning in the warehouse. This optimization minimizes spatial and energy consumption by providing the most efficient cell and path for automated AGVs. RL focuses on long-term reward-based decision-making.
- Deep Q-Learning (DQN), a type of reinforcement learning, leverages deep neural networks (DNNs) to predict values and address large-scale problems [26]:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s,a,s') \max_{a'} Q(s',a') \quad (1)$$

- The AI model is trained using three main input datasets: weight, import/export dates, and frequency of import/export for each type of product. In the enhanced E-WMS, the environment is the warehouse with a fixed storage space, the agent is the AGV, and the state is the AGV's current position with the reward of that position. The pallet information (weight, import/export dates, frequency) also serves as the policy for evaluating the algorithm.
  - The AGV's current position.
  - A list of occupied and available storage slots
  - Pallet information (weight, import/export dates, frequency) which also serves as the input data for the algorithm.

Actions a involve the AGV moving pallets to an appropriate location. Rewards r evaluate the efficiency of these actions: storing a pallet in the correct position yields a high reward, while incorrect actions result in lower rewards (Fig. 4).

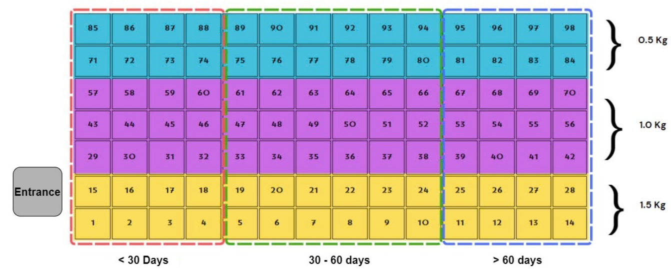


Figure 4: Integration of AI models into Warehouses.

(2) SQL serves as the database system for processing and storing all general information in the system which is a crucial node for data exchange between the WMS and the automated support systems.

(3) PLC Master: The PLC Master acts as the central brain of the automated support system. It processes data from the E-WMS via SQL, sends execution requests to subsystems (e.g., AGVs, conveyors, and RFID), and updates system states back to the WMS.

The E-WMS system manages and optimizes storage locations by analyzing order data (via Excel file or manual input) and using AI algorithms to determine optimal pallet positions based on storage time and available slots. SQL database stores and exchanges real-time order and warehouse status data between the E-WMS and the SWH systems. The PLC Master acts as the central controller and processes data from E-WMS and the SWH systems to navigate AGVs and conveyors. RFID readers and sensors continuously track pallet locations, while PLC Slaves execute actions like transporting pallets according to the orders from the PLC Master.

3.3 Calculation of Travel Time and Energy Consumption for AGVs

To research the performance of smart warehouses before and after AI integration, equations (2) and (3) were developed based on analyses of the collected and measured dataset.

The travel time of the AGV is calculated from the "Entrance" position to the starting position of the warehouse, measured as 2.5 seconds with an energy consumption of 0.319 W·s.

The movement speed and idle power consumption of the motor along each axis are:

- X-axis: Speed = 500 mm/s, Idle power consumption $P_x = 0.087$ W
- Y-axis: Speed = 350 mm/s, Idle power consumption $P_y = 0.058$ W

The time required for movement along each axis is calculated using the formula $t = \frac{s}{v}$:

- Movement time for each grid cell along the x-axis (row):

$$t_x = 0.4 \text{ s} \tag{2}$$

- Movement time for each grid cell along the y-axis (column):

$$t_y = 0.86 \text{ s} \tag{3}$$

Since the distance from the "Entrance" to the starting position is equivalent to the first three rows of the first column (0.8 m × 0.6 m), and the starting time is 2.5 s, the total travel time for one grid cell is calculated as follows:

$$t = 2.5 + \max(t_x(x-1), t_y(y-3), 0) \quad \text{(s)} \tag{4}$$

The energy consumption for each pallet position is influenced by the travel time along each axis (x, y), the weight m, and the energy coefficient a. The formula is as follows:

$$A = 0.319 + (t_x \cdot x)P_x + (t_y \cdot y)P_y + m \cdot t \cdot a \quad (\text{W}\cdot\text{s}) \tag{5}$$

Where:

- a: Coefficient of increased motor power consumption for carrying loads (varies depending on motor type).
- m: Weight of each pallet, classified into three groups with coefficients of 0.5, 1, and 1.5, used to calculate the load energy consumption.

3.4 Mobile Application Workflow in the Smart Warehouse System

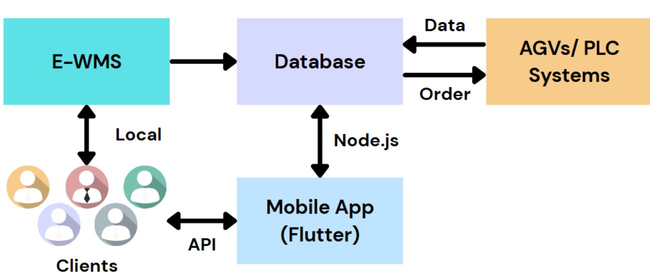


Figure 5: The flow of mobile application.

Current Logistics management services focus on improving user experience and increasing information reliability. With the advantages of applying digitalization and CIT in warehouse data management through E-WMS and Database - these are the advantages that SWH can bring to improve user experience. A mobile platform developed using Flutter is integrated into the SWH system to help Customers access real-time information through APIs provided by the Node.js-based backend service. Figure 5 is as follows:

[1] Users initiate tasks via E-WMS: Operations such as inventory requests and order creation are performed through desktop terminals or devices within the local network.

[2] E-WMS stores data in the central database: Operational data such as order status and product movement are stored and retrieved from the centralized database.

[3] Node.js handles real-time data synchronization: A backend service built with Node.js connects the database to the mobile application via RESTful APIs, supporting queries and user authentication.

[4] Flutter provides a cross-platform mobile interface: Users can access key functions like task reception, order tracking, QR/barcode scanning, and task status updates through the mobile app.

[5] Database communicates with AGV/PLC systems: Order details are sent to AGVs/PLCs for execution, and feedback such as completion status and location is returned.

[6] Feedback is synchronized with the mobile app: Updates from AGV/PLC systems are recorded in the database and reflected immediately in the mobile interface via Node.js.

[7] Offline support and automatic synchronization: The app caches data when offline and automatically syncs it to the central database once the connection is restored.

3.5 Digital Twin Integration for Engineer-To-Order Operations

Besides offering strong data accessibility, warehouse digitalization holds significant potential in predictive analytics and strategic decision-making through the development of a Digital Twin model for the Smart Warehouse (SWH).

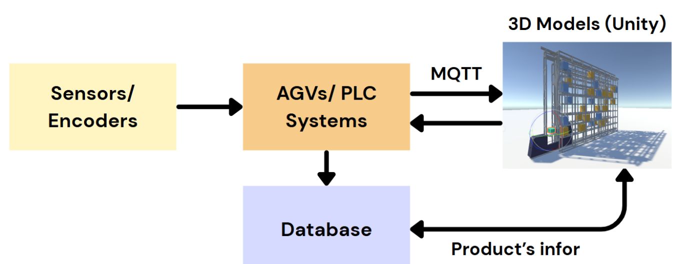


Figure 6: Digital Twin data transmission flow.

To support real-time visualization and Engineer-to-Order (ETO) workflows, a Digital Twin architecture has been implemented in the SWH. The overall data flow and system integration are illustrated in Fig. 6.

[1] Sensor and encoder data collection: The physical warehouse is equipped with sensors and encoders attached to AGVs and machinery. These devices collect real-time data such as position, velocity, and operational status.

[2] AGV/PLC system coordination: Raw sensor data is transmitted to AGV/PLC controllers. These systems process the data to make motion control decisions and execute automated handling tasks—forming the key interface between the physical and digital domains.

[3] Real-time synchronization via MQTT: To enable lightweight, low-latency communication between physical systems and the virtual environment, MQTT (Message Queuing Telemetry Transport) is used. It ensures efficient and reliable telemetry data transfer from AGV/PLC to the Unity-based Digital Twin.

[4] 3D visualization in Unity: A virtual warehouse environment is developed in Unity, enabling real-time visualization of warehouse operations. Through MQTT, the 3D models dynamically reflect the current state of physical systems, including product positions, AGV movements, and activity flows—forming a fully synchronized Digital Twin.

[5] Database integration and product mapping: In addition to telemetry data, product-specific information—such as SKU codes, order IDs, and storage locations—is retrieved from the database. This data is used to annotate the 3D models, providing contextual insights for warehouse operators and planners.

[6] Support for Engineer-to-Order (ETO) operations: The Digital Twin offers an interactive interface that supports ETO operations by enabling real-time tracking of custom orders, verifying product routing, and ensuring configurations meet customer-specific requirements. Engineers can simulate logistics scenarios, monitor performance, and instantly adjust control parameters within the virtual model—shortening the feedback loop between design and execution.

This Digital Twin framework enhances monitoring and decision-making, bridging the gap between the physical warehouse and digital control systems, and supporting adaptive smart warehouse management tailored to ETO scenarios.

4 Results and Discussion

Our warehouse system significantly reduces energy consumption by nearly 25% and improves overall operational efficiency by cutting travel time by 15%. Testing on 98 warehouse positions demonstrates the superior effectiveness of this approach compared to traditional management processes and verifies the feasibility of AI technology in warehouse management (Fig. 7).

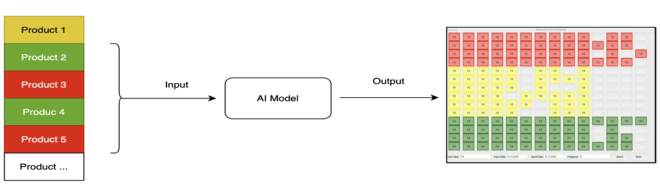


Figure 7: Integration of AI models into Warehouses.

To compare the efficiency of SWH, E-WMS with AI has proven to be more efficient in warehouse management than the previous version which used a traditional algorithm without RL, as evidenced by a depreciation comparison after one week between the two solutions. According to the data, energy consumption and travel time have decreased significantly compared to traditional methods. Travel time was reduced by nearly 10 seconds (approximately 14%), travel energy consumption decreased by 1.02 (W.s) (10.3%), and load energy consumption dropped by 30% during pallet transport. After the R&D phase, the Mobile application has successfully implemented basic monitoring and control features with a detailed interface as shown in Fig. 8.

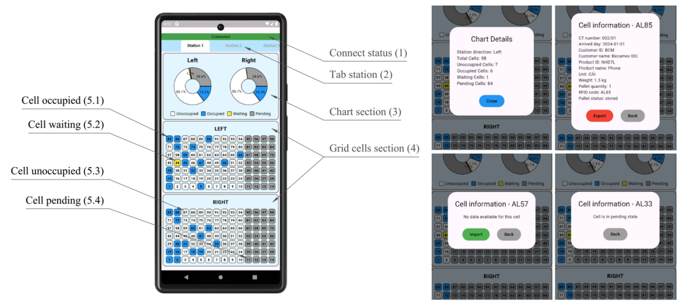


Figure 8: Mobile application for SWH.

1. Connect status: Displays the application's connection status with the server.
2. Tab station: Allows switching between dashboards for 7 stations.
3. Chart section: Displays charts to manage the status and quantity of cells in each station.
4. Grid cells section: Shows the grid of cells within a station for interaction.
5. Cell status:
   5.1 Green indicates a cell containing goods.
   5.2 Yellow indicates a cell in the process of loading or unloading goods.
   5.3 White indicates an empty cell.
   5.4 Gray indicates a pending cell status.

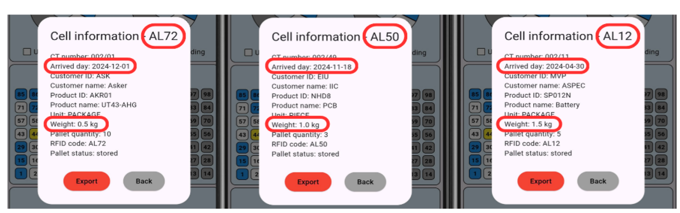


Figure 9: Optimized pallet location information by AI.

In addition to information updated from the database regarding detailed inventory and current locations, Fig. 9 illustrates that the AI model has achieved the projected arrangement strategy as shown in Fig. 4. Pallets are allocated based on designed levels of storage time (less than 30 days, from 31 to 60 days, and over 60 days) and weight categories (0.5 kg, 1 kg, 1.5 kg).

Figure 10: Smart Warehouse after AI and IoT integration.

The integration of AI into the warehouse model has been evaluated using key metrics:

- Import time for the nearest pallet (01): 22s, and the farthest pallet (98): 35s.
- Export time for the nearest pallet (01): 25s, and the farthest pallet (98): 38s.
- Conveyor time for the nearest station (G07): 18s, and the farthest station (A01): 53s.

Compared to traditional warehouse systems, the smart warehouse demonstrates significant efficiency improvements in order processing, from retrieval to transportation (Fig. 10). By optimizing pallet allocation to positions, the system saves energy across all stations, ensures more balanced distribution, reduces excessive equipment operation, and enhances economic benefits.

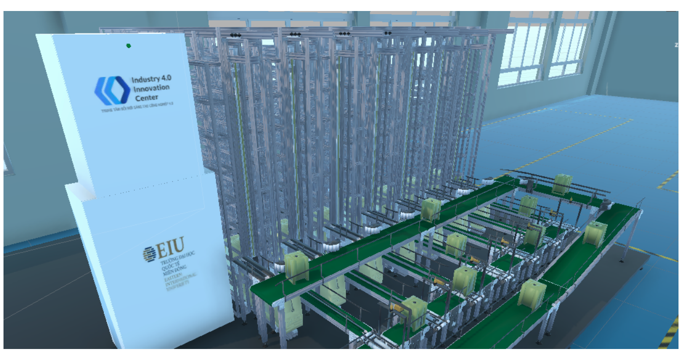


Figure 11: Digital Twin Model of SWH in Unity.

Currently, this DT model has successfully received data in parallel with the SWH and has successfully simulated the movement of AGVs within the SWH (Fig. 11). It can also communicate with the AI-integrated WMS. In the future, the team will collect additional data from sensors and databases to further refine the model and support the development of Engineer-To-Order (ETO) for AI specifically tailored to the SWH, as well as management applications related to energy and risk management.

Limitation:

The current implementation is limited to a 1:10 scale prototype, which does not fully capture the complexity and challenges of deploying and operating such systems in real warehouse environments. Moreover, the machine learning model is currently constrained by a limited dataset, which indicates significant potential for further optimization of both the model architecture and control policies. To fully evaluate the impact of integrating WMS, IoT, AI, and a dynamic Digital Twin in real-world scenarios, substantial infrastructure—including RFID systems, AGVs, IoT devices, and Computer Information Technology (CIT) platforms—is required. This poses challenges in terms of time and cost, particularly for small and medium-sized enterprises (SMEs). A complete transformation would demand a progressive digitalization process, moving from WMS-based workflows to semi-automated or automated systems, followed by the integration of CIT/IoT technologies, and eventually achieving full DT implementation.

Future works:

In the near future, the research team aims to expand the capabilities of the DT environment to support predictive maintenance (PM) and testing solutions. By developing comprehensive testing tools within the DT framework, it will be possible to simulate and evaluate various warehouse layout strategies and policies, that enable the identification of the most effective machine learning models. This will also facilitate the collection of valuable operational data to support the development of predictive maintenance algorithms.

Additionally, the mobile application will be further enhanced with advanced security features and deployed in real warehouse environments to assess its effectiveness and impact on warehouse management performance. Future versions of the app will also include integrated chatbot functionalities to improve user interaction and service quality in inventory and workflow management.

5 Conclusions

In addition to the proven benefits in operational efficiencies—such as faster inbound and outbound processes—and energy optimization—through reduced consumption and minimized machinery depreciation—the integration of RFID, AI, DT and mobile applications with the E-WMS software, coupled with synchronization with the database system, establishes a comprehensive and intelligent seamless process. The SWH is one of steps for us to shift the focus from human-intensive to technology-intensive. It emphasizes strategic management and sustainable development.

The proposed solution represents a step toward the advancement of Industry 5.0, with a focus on sustainable energy practices and a more human-centric approach to technological development. Leveraging the current capabilities of modern sensor systems and database infrastructures, the potential of SWHs extends into predictive maintenance and digital twin technologies. These innovations enable the identification and mitigation of potential issues before they arise, facilitate improved risk management and proactive responses to market changes through predictive scenario modeling.

References

[1] Ifadhila Affia and Ammar Aamer. An internet of things-based smart warehouse infrastructure: design and application. *Journal of Science and Technology Policy Management*, 13(1):90–109, feb 2022.

[2] Imran Ali and Huy Minh Phan. Industry 4.0 technologies and sustainable warehousing: a systematic literature review and future research agenda. *International Journal of Logistics Management*, 33(2):644–662, apr 2022.

[3] Janani Shamindika Kumari Angammana and Malinthi Jayawardena. Influence of artificial intelligence on warehouse performance: The case study of the Colombo area, Sri Lanka. *Journal of Sustainable Development of Transport and Logistics*, 7(2):80–110, nov 2022.

[4] E Čengić. Assessing the Benefits and Challenges of Integrating Artificial Intelligence in Warehouses, 2024.

[5] Mafalda Correia and Piçarra Gonçalves. The possibility of implementing intelligent systems and the respective impact of Artificial Intelligence on inventory management and warehousing. 2022.

[6] Rodrigo Furlan de Assis, Alexandre Frias Faria, Vincent Thomasset-Laperrière, Luis Antonio Santa-Eulalia, Mustapha Ouhimmou, and William de Paula Ferreira. Machine Learning in Warehouse Management: A Survey. *Procedia Computer Science*, 232:2790–2799, 2024.

[7] Aldona Jarašūnienė, Kristina Čižiūnienė, and Audrius Čereška. Research on Impact of IoT on Warehouse Management. *Sensors 2023, Vol. 23, Page 2213*, 23(4):2213, feb 2023.

[8] Dr. Ali Kamali. Smart Warehouse vs. Traditional Warehouse - Review. *Automation and Autonomous System*, 11(1):9–16, 2019.

[9] Aswin M Kumar. Smart Warehouse Governance using AI and Raspberry Pi. *International Journal of Recent Technology and Engineering (IJRTE)*, (9):2277–3878, 2020.

[10] Pham Nguyen My Linh and Nguyen Thi Thu Huong. The Supply Chain and Logistics of Vietnam in the Context of International Economic Integration. *International Business Research*, 13:27, may 2020.

[11] Kamran Mahroof. A human-centric perspective exploring the readiness towards smart warehousing: The case of a large retail distribution warehouse. *International Journal of Information Management*, 45:176–190, apr 2019.

[12] S Marino. Delivering smart warehousing in Australia. 23:131–138, 2023.

[13] Hokey Min. Smart Warehousing as a Wave of the Future. *Logistics 2023, Vol. 7, Page 30*, 7(2):30, may 2023.

[14] S. P. Plakantara, Athanasia Karakitsiou, and T. Mantzou. Managing Risks in Smart Warehouses from the Perspective of Industry 4.0. *Springer Optimization and Its Applications*, 214:1–47, 2024.

[15] Mehrdokht Pournader, Hadi Ghaderi, Amir Hassanzadegan, and Behnam Fahimnia. Artificial intelligence applications in supply chain management. *International Journal of Production Economics*, 241:108250, nov 2021.

[16] Alessandro Pracucci. Designing Digital Twin with IoT and AI in Warehouse to Support Optimization and Safety in Engineer-to-Order Manufacturing Process for Prefabricated Building Products. *Applied Sciences (Switzerland)*, 14(15), aug 2024.

[17] Om Prakash Satyam and Wakil Kumar. Study on IoT & AI for Smart Warehouse Management for Green Supply Chain Management. 9, 2022.

[18] Marcin Szczepański. BRIEFING EPRS — European Parliamentary Research Service.

[19] Temitayo Oluwadamilola Adesoga, Tolulope Olusola Ajibaye, Kenneth Chukwujekwu Nwafor, Ummulikhaeri Temitope Imam-Lawal, Emmanuel Ayobamidele Ikekwere, and Daniel Ikechukwu Ekwunife. The rise of the "smart" supply chain: How AI and automation are revolutionizing logistics. *International Journal of Science and Research Archive*, 12(2):790–798, jul 2024.

[20] Nguyen Tien Minh, Nguyen Thi Thu Ha, and Nguyen Duc Duong. Developing E-Commerce In Vietnam. *Journal of Positive School Psychology*, 6(8):5839–5848, aug 2022.

[21] Saurabh Tiwari. Smart warehouse: A bibliometric analysis and future research direction. *Sustainable Manufacturing and Service Economics*, 2:100014, apr 2023.

[22] Van Luan Tran, Manh Kha Kieu, Xuan Hung Nguyen, Vu Anh Tram Nguyen, Tran Thuy Duong Ninh, Duc Canh Nguyen, Narayan C. Debnath, Ngoc Bich Le, and

Ngoc Huan Le. Reinforcement Learning for Developing an Intelligent Warehouse Environment. *Lecture Notes on Data Engineering and Communications Technologies*, 113:11–20, 2022.

[23] Md Kazi Shahab Uddin and Kazi Md Riaz Hossan. A review of implementing AI-powered data warehouse solutions to optimize big data management and utilization. *Academic Journal on Business Administration, Innovation & Sustainability*, 4(3):66–78, jul 2024.

[24] Maarten van Geest, Bedir Tekinerdogan, and Cagatay Catal. Smart Warehouses: Rationale, Challenges and Solution Directions. *Applied Sciences 2022, Vol. 12, Page 219*, 12(1):219, dec 2021.

[25] Nguyen Van Hoa, Pham Van Thu, Nguyen Thanh Dat, Le Thanh Loan, Vietnam Aviation Academy, Nguyen Van Troi, Phu Nhuan district, Ho Chi Minh City, and Corresponding Author. Analysis of the business environment of the logistics and the construction industry in Vietnam after Covid-19. *Int. j. adv. multidisc. res. stud*, 3(1):316–328, 2023.

[26] Viblo-Blog. Reinforcement learning: Q-learning, 2024.

[27] Yashkumar Wankhede, Sonakshi Singh, Omkar Chaskar, Aditya Chaudhary, and S L Dawkhar. Smart ware: Intelligent Warehouse Application Using AI/ML. *IJIRT 169422 INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY*, 989, 2024.

[28] Judy X Yang, Lily D Li, and Mohammad G. Rasul. Warehouse Management Models Using Artificial Intelligence Technology with Application at Receiving Stage – A Review. *International Journal of Machine Learning and Computing*, 11(3):242–249, may 2021.

[29] Lu Zhen and Haolin Li. A literature review of smart warehouse operations management. *Frontiers of Engineering Management 2021 9:1*, 9(1):31–55, jan 2022.

[30] Margareta Živicnjak, Kristijan Rogic, and Ivona Bajor. Case-study analysis of warehouse process optimization. *Transportation Research Procedia*, 64(C):215–223, jan 2022.

ISCA INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS Vol. 32 No. 2, June 2025