PAPER ON AI BASED APPLICATION FOR REAL TIME APPLICATION

N. Bhaskar[1], Archana Patil[2] , Madhu Bandi[3] , S. Ramana[4], Anil Kumar Masimukku[5] , M.V. Ramana Murthy[6]

[1]Ramnath Guljarilal Kedia College Commerce , 3-1-336, Esmania bazar, Opp. Chadarghat bridge Hyderabad–500027

[2]Asst. Professor, Dept. of Computer Science & Engineering, Rishi MS Institute of Engineering and Technology for women, Hyderabad, Telangana, India.

[3]Independent researcher, 7903, Elm, Ave Apts #257, Rancho Cucamonga, CA 91730, U S A.

[4]Bhavan's Vivekananda , Dept. of Computer Science, Asst. Professor& Controller of Examinations , Sainikpuri, Secunderabad—500 094

[5]Independent Researcher, 10623 Canoe Dr , Coppell, TX-75019. U.S.A

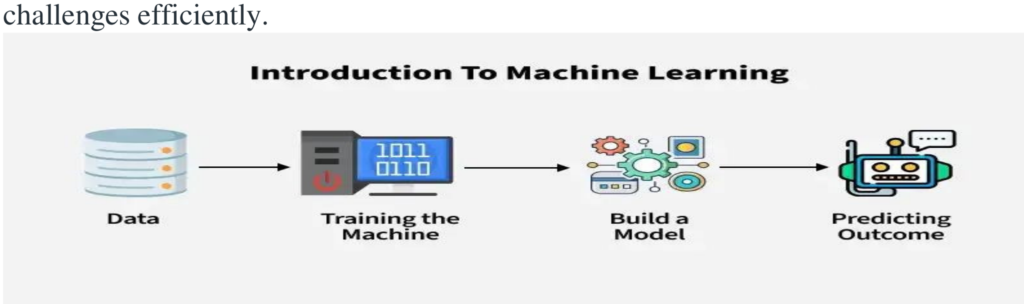
[6]Former Professor and Chairman In Computer Science and Mathematics, Osmania university, 7556, Covington Pl, Rancho Cuca Monga, California, 91730, U S A.

Abstract : In this study, a Machine Learning (ML) is implemented to soft computation of the Reconfigurable Horn Bowtie Dumbbell (RHBD) antenna at operating frequency range from 26 GHz to 29.5 GHz for 5G applications. An adaptive learning rate approach is used to build a ML model on a 5-layer system utilizing a simulated database of 180 RHBD antennas. In the training stage of a hybrid method that combines the advantages of particle swarm optimization (PSO) with a modified version of the gravitational search algorithm (MGSA), the architecture frame and hyper-parameters of the ML model are optimized. A precise electromagnetic analysis platform is used to simulate 180 RHBD antennas with varying geometrical properties in terms of the resonant frequency in order to create the database for training and testing the model. The ML model is tested and validated using a fabricated RHBD antenna operating at 27.5 GHz. Then, three PIN diodes are placed in the gaps of the reflectors located at the back of the antenna, and by changing the state of these PIN diodes, it can be noticed that they have a significant and direct effect on the radiation pattern, as they are able to change the beamwidth from 10.7° to 156.2°. The suggested antenna makes it easier to create dynamic radiation patterns that may be utilized to reconfigure the coverage area as required in accordance with the spatial-temporal user and traffic variations in high mobility environments.

Key words: Natural Language Processing, Decision Tree, Natural Language Processing, Fitting, unsupervised Learning, Algorithms for analysis.

Introduction:

Introduction to Machine Learning: What Is and Its Applications. Machine learning (ML) allows computers to learn and make decisions without being explicitly programmed. It involves feeding data into algorithms to identify patterns and make predictions on new data. **Machine learning is used in various applications, including image and speech recognition, natural language processing, and recommender systems.** **Machine Learning** algorithm learns from data, train on patterns, and solve or predict complex problems beyond the scope of traditional programming. It drives better decision-making and tackles intricate challenges efficiently.



1. **Solving Complex Business Problems**
Traditional programming struggles with tasks like image recognition, natural language processing (NLP), and medical diagnosis. ML, however, thrives by learning from examples and making predictions without relying on predefined rules.

Example Applications:

- Image and speech recognition in healthcare.
- Language translation and sentiment analysis.

2. Handling Large Volumes of Data
With the internet's growth, the data generated daily is immense. ML effectively processes and analyze this data, extracting valuable insights and enabling real-time predictions.

Use Cases:

- Fraud detection in financial transactions.
- Social media platforms like Facebook and Instagram predicting personalized feed recommendations from billions of interactions.

3. Automate Repetitive Tasks
ML automates time-intensive and repetitive tasks with precision, reducing manual effort and error-prone systems.

Examples:

- **Email Filtering**: Gmail uses ML to keep your inbox spam-free.
- **Chatbots**: ML-powered chatbots resolve common issues like order tracking and password resets.
- **Data Processing**: Automating large-scale invoice analysis for key insights.

Key words : Deep learning, Over Fitting, Unsupervised Learning, Algorithms for analysis, Natural Language Processing, Decision Tree, Natural Language Processing.

Analysis: Problem analysis of Artificial Intelligence and divided into different parts the following are elained herewith for your reference.

Step 1: Define the problem. Identify the core ML task and ask clarifying questions to determine the appropriate requirements and tradeoffs. (8 minutes)

Step 2: Design the data processing pipeline. Illustrate how you'll collect and process your data to maintain a high-quality dataset. (8 minutes)

Step 3: Create a model architecture. Come up with a suitable model architecture that would address the needs of the core ML task identified in Step 1. (8 minutes)

Step 4: Train and evaluate the model. Select a model and explain how you'll train and evaluate it. (8 minutes)

Step 5: Deploy the model. Determine how you'll deploy the model, how it will be served, and how to monitor it. (8 minutes)

Step 6: Wrap up. Summarize your solution and present additional considerations you would address with more time. (5 minutes)

Analysis:

Predictive ML and data

Data is the driving force of predictive ML. To make good **predictions**, you need data that contains **features** with predictive power. Your data should have the following characteristics:

- **Abundant**. The more relevant and useful examples in your **dataset**, the better your model will be.

- **Consistent and reliable**. Having data that's consistently and reliably collected will produce a better model. For example, an ML-based weather model will benefit from data gathered over many years from the same reliable instruments.
- **Trusted**. Understand where your data will come from. Will the data be from trusted sources you control, like logs from your product, or will it be from sources you don't have much insight into, like the output from another ML system?
- **Available**. Make sure all inputs are available at prediction time in the correct format. If it will be difficult to obtain certain feature values at prediction time, omit those features from your datasets.
- **Correct**. In large datasets, it's inevitable that some **labels** will have incorrect values, but if more than a small percentage of labels are incorrect, the model will produce poor predictions.
- **Representative**. The datasets should be as representative of the real world as possible. In other words, the datasets should accurately reflect the events, user behaviors, and/or the phenomena of the real world being modeled. Training on unrepresentative datasets can cause poor performance when the model is asked to make real-world predictions.

Predictive power

For a model to make good predictions, the features in your dataset should have predictive power. The more correlated a feature is with a label, the more likely it is to predict it.

Some features will have more predictive power than others. For example, in a weather dataset, features such as cloud coverage, temperature, and dew point would be better predictors of rain than monophase or day of week. For the video app example, you could hypothesize that features such as video description, length and views might be good predictors for which videos a user would want to watch.

Determining which features have predictive power can be a time-consuming process. You can manually explore a feature's predictive power by removing and adding it while training a model. You can automate finding a feature's predictive power by using algorithms such as Pearson correlation, Adjusted mutual information (AMI), and Shapley value, which provide a numerical assessment for analysing the predictive power of a feature.

Check Your Understanding

When analysing your datasets, what are three key attributes you should look for?

Gathered from a variety of unpredictable sources.

Representative of the real world.

Small enough to load onto a local machine.

Features have predictive power for the label.

Predictions vs. actions

There's no value in predicting something if you can't turn the prediction into an action that helps users. That is, your product should take action from the model's output.

For example, a model that predicts whether a user will find a video useful should feed into an app that recommends useful videos. A model that predicts whether it will rain should feed into a weather app.

Check Your Understanding

Based on the following scenario, determine if using ML is the best approach to the problem.

An engineering team at a large organization is responsible for managing incoming phone calls.

The goal: To inform callers how long they'll wait on hold given the current call volume.

They don't have any solution in place, but they think a heuristic would be to divide the current number of customers on hold by the number of employees answering phones, and then multiply by 10 minutes. However, they know that some customers have their issues resolved in two minutes, while others can take up to 45 minutes or longer.

Their heuristic probably won't get them a precise enough number. They can create a dataset with the following columns: number_of_callcenter_phones, user_issue, time_to_resolve, call_time, time_on_hold.

Use ML. The engineering team has a clearly defined goal. Their heuristic won't be good enough for their use case. The dataset appears to have predictive features for the label, time on hold.

Don't use ML. Although they have a clearly defined goal, they should implement and optimize a non-ML solution first. Also, their dataset doesn't appear to contain enough features with predictive power.

9 Real-World Problems that can be

Solved by Machine Learning

Machine Learning has gained a lot of prominence in the recent years because of its ability to be applied across scores of industries to solve complex problems effectively and quickly. Contrary to what one might expect, Machine Learning use cases are not that difficult to come across. The most common examples of problems solved by machine learning are image tagging by Facebook and spam detection by email providers.

How to Solve a Simple Problem With Machine Learning

Welcome back to the second lesson in my series, ML Lessons for Managers and Engineers. Today, by popular demand, I'll walk you through implementing the solution I wrote about in lesson one. This is a more technical lesson than I originally intended for this series, but I believe that most professionals benefit from a better understanding of machine learning technology.

To keep it as relevant as possible, I'll focus mainly on the underlying reasoning because that's where the valuable lessons exist. If you want to study the code in detail, there's a GitHub link at the bottom of the page. Problem solving is a core aspect of artificial intelligence (AI) that mimics human cognitive processes. It involves identifying challenges, analyzing situations, and applying strategies to find effective solutions.

This article explores the various dimensions of problem solving in AI, the types of problem-solving agents, the steps involved, and the components that formulate associated problems.

Table of Content

- Challenges in Problem Solving with AI

Understanding Problem-Solving Agents

In artificial intelligence (AI), agents are entities that perceive their environment and take actions to achieve specific goals. Problem-solving agents stand out due to their focus on identifying and resolving issues systematically. Unlike reflex agents, which react to stimuli based on predefined mappings, problem-solving agents analyze situations and employ various techniques to achieve desired outcomes.

Types of Problems in AI

1. Ignorable Problems

These are problems or errors that have minimal or no impact on the overall performance of the AI system. They are minor and can be safely ignored without significantly affecting the outcome.

Examples:

- Slight inaccuracies in predictions that do not affect the larger goal (e.g., small variance in image pixel values during image classification).
- Minor data preprocessing errors that don't alter the results significantly.

Handling: These problems often don't require intervention and can be overlooked in real-time systems without adverse effects.

2. Recoverable Problems

Recoverable problems are those where the AI system encounters an issue, but it can recover from the error, either through manual intervention or built-in mechanisms, such as error-handling functions.

Examples:

- Missing data that can be imputed or filled in by statistical methods.
- Incorrect or biased training data that can be retrained or corrected during the process.
- System crashes that can be recovered through checkpoints or retraining.

Handling: These problems require some action—either automated or manual recovery. Systems can be designed with fault tolerance or error-correcting mechanisms to handle these.

3. Irrecoverable Problems

Description: These are critical problems that lead to permanent failure or incorrect outcomes in AI systems. Once encountered, the system cannot recover, and these problems can cause significant damage or misperformance.

Examples:

- Complete corruption of the training dataset leading to irreversible bias or poor performance.
- Security vulnerabilities in AI models that allow for adversarial attacks, rendering the system untrustworthy.
- Overfitting to the extent that the model cannot generalize to new data.

Handling: These problems often require a complete overhaul or redesign of the system, including retraining the model, rebuilding the dataset, or addressing fundamental issues in the AI architecture.

Steps in Problem Solving in Artificial Intelligence (AI)

The process of problem solving in AI consists of several finite steps that parallel human cognitive processes. These steps include:

1. **Problem Definition:** This initial step involves clearly specifying the inputs and acceptable solutions for the system. A well-defined problem lays the groundwork for effective analysis and resolution.
2. **Problem Analysis:** In this step, the problem is thoroughly examined to understand its components, constraints, and implications. This analysis is crucial for identifying viable solutions.
3. **Knowledge Representation:** This involves gathering detailed information about the problem and defining all potential techniques that can be applied. Knowledge representation is essential for understanding the problem's context and available resources.
4. **Problem Solving:** The selection of the best techniques to address the problem is made in this step. It often involves comparing various algorithms and approaches to determine the most effective method.

Components of Problem Formulation in AI

Effective problem-solving in AI is dependent on several critical components:

- **Initial State:** This represents the starting point for the AI agent, establishing the context in which the problem is addressed. The initial state may also involve initializing methods for problem-solving.
- **Action:** This stage involves selecting functions associated with the initial state and identifying all possible actions. Each action influences the progression toward the desired goal.
- **Transition:** This component integrates the actions from the previous stage, leading to the next state in the problem-solving process. Transition modeling helps visualize how actions affect outcomes.
- **Goal Test:** This stage verifies whether the specified goal has been achieved through the integrated transition model. If the goal is met, the action ceases, and the focus shifts to evaluating the cost of achieving that goal.
- **Path Costing:** This component assigns a numerical value representing the cost of achieving the goal. It considers all associated hardware, software, and human resource expenses, helping to optimize the problem-solving strategy.

Techniques for Problem Solving in AI

Several techniques are prevalent in AI for effective problem-solving:

1. Search Algorithms

Search algorithms are foundational in AI, used to explore possible solutions in a structured manner. Common types include:

- **Uninformed Search**: Such as breadth-first and depth-first search, which do not use problem-specific information.
- **Informed Search**: Algorithms like A* that use heuristics to find solutions more efficiently.

2. Constraint Satisfaction Problems (CSP)

CSPs involve finding solutions that satisfy specific constraints. AI uses techniques like backtracking, constraint propagation, and local search to solve these problems effectively.

3. Optimization Techniques

AI often tackles optimization problems, where the goal is to find the best solution from a set of feasible solutions. Techniques such as linear programming, dynamic programming, and evolutionary algorithms are commonly employed.

4. Machine Learning

Machine learning techniques allow AI systems to learn from data and improve their problem-solving abilities over time. Supervised, unsupervised, and reinforcement learning paradigms offer various approaches to adapt and enhance performance.

5. Natural Language Processing (NLP)

NLP enables AI to understand and process human language, making it invaluable for solving problems related to text analysis, sentiment analysis, and language translation. Techniques like tokenization, sentiment analysis, and named entity recognition play crucial roles in this domain.

Conclusion and Future Scope:

Conclusion & future References: It is completed with lot of corrections and Supply Chain management, and return with lot of support for the future reference. It is closed with reference to ML and Computing Learning techniques with proper updating of Supervised Learning.

Challenges in Problem Solving with AI

Despite its advancements, AI problem-solving faces several challenges:

1. **Complexity**: Some problems are inherently complex and require significant computational resources and time to solve.
2. **Data Quality**: AI systems are only as good as the data they are trained on. Poor quality data can lead to inaccurate solutions.
3. **Interpretability**: Many AI models, especially deep learning, act as black boxes, making it challenging to understand their decision-making processes.
4. **Ethics and Bias**: AI systems can inadvertently reinforce biases present in the training data, leading to unfair or unethical outcomes.

Bibliography :

1.Eletrôn. Potên., Florianópolis, v. 27, n. 3, p. 236-243, jul./set. 2022236Eletrôn. Potên., Florianópolis, v. 27, n. 3, p. 236-243, jul./set. 2022236Fault Diagnosis of Rotary Machines Using Machine 1. LearningIago M. Brandao, Cesar da CostaFAULT DIAGNOSIS OF ROTARY MACHINES USING MACHINE LEARNINGIago M. Brandao, Cesar da CostaDepartment of Electrical Engineering, IFSP Federal Institute of São Paulo, São Paulo, Braz.

Bibliography:

1. LearningIago M. Brandao, Cesar da CostaFAULT DIAGNOSIS OF ROTARY MACHINES USING MACHINE LEARNINGIago M. Brandao, Cesar da Costa Department of Electrical Engineering, IFSP Federal Institute of São Paulo, São Paulo, Braz.

2. “Machine learning and learning from machines”, Naeini, Ehsan Zabihi, and Kenton Prindle. "Machine learning and learning from machines." *Leading Edge* 37, no. 12

3. “Credit Risk Rating Using State Machines and Machine Learning”,Sabeti, Behnam, Hossein Abedi Firouzjaee, Reza Fahmi, Saeid Safavi, Wenwu Wang, and Mark D. Plumbley. "Credit Risk Rating Using State Machines and Machine Learning." *International Journal of Trade, Economics and Finance* 11, no. 6

4. “Deceiving Machines: Sabotaging Machine Learning”Trott, David. " Deceiving Machines: Sabotaging Machine Learning." *CHANCE* 33, no. 2.

5. “Machine Learning and Cryptography”, Siddique, Shumaila. "Machine Learning and Cryptography." Journal of Advanced Research in Dynamical and Control Systems *12, SP7.*

6. “Economie et Statistique / Economics and Statistics n° 505-506 – 2018”,Arthur, Emmanuel Flachaire, and Antoine Ly. "Econometrics and Machine Learning." *Economie et Statistique / Economics and Statistics*, no. 505.

7. “Introduction to Machine Learning”, Mor, Laksanya. "Introduction to Machine Learning." *International Journal of Science and Research (IJSR)* 11, no. 3.

8.”Learning Machine Learning”, Lewis, Ted G., and Peter J. Denning. "Learning machine learning." *Communications of the ACM* 61, no. 12.

9. “Topic Detection using Machine Learning”, Rasi, Mr Ajmal, Dr Rajasimha A. Makram, and Ms Shilpa Das. "Topic Detection using Machine Learning." *International Journal of Trend in Scientific Research and Development* Volume-2, Issue-4.

10. “Detecting Phishing using Machine Learning”, Mudiraj, Nakkala Srinivas. "Detecting Phishing using Machine Learning." *International Journal of Trend in Scientific Research and Development* Volume-3, Issue-4.

11. “Create a spot-on reference in APA, MLA, Chicago, Harvard”, Virley, David, Sarah J. Hadingham, Jenny C. Roberts, Belinda Farnfield, Heather Elliott, Greg Whelan, Jackie Golder, Chris David, Andrew A. Parsons, and A. Jackie Hunter. "A New Primate Model of Focal Stroke: Endothelin-1—Induced Middle Cerebral Artery Occlusion and Reperfusion in the Common Marmoset." *Journal of Cerebral Blood Flow & Metabolism* 24, no. 1 (January 2004): 24–41.

12. “Social implications of gummivory in marmosets”, Harrison, Mary L., and Suzette D. Tardif. "Social implications of gummivory in marmosets." *American Journal of Physical Anthropology* 95, no. 4.

13. “The Monstar and Calmoncoi Polygone , behaviour anatomy conversion and convertation ”, Sussman, Robert W. "The Marmosets and Callimico: Phylogeny, Behavior, Anatomy and Conservation." *Journal of Mammalian Evolution* 18.

14. “Systematics and evolution of the jacchus group of marmosets (Platyrrhini)”,, Marroig, Gabriel, Susan Cropp, and James M. Cheverud. "Systematics and evolution of the jacchus group of marmosets (Platyrrhini)." *American Journal of Physical Anthropology* 123, no. 1.

15. ” Anatomical description and morphometry of the skeleton of the common marmoset (*Callithrix jacchus*)”, Casteleyn, C., J. Bakker, S. Breugelmans, I. Kondova, J. Saunders, J. A. M. Langermans, P. Cornillie, et al. "Anatomical description and morphometry of the skeleton of the common marmoset (Callithrix jacchus).

16. “Kin preference in marmosets and tamarins: *Saguinus oedipus* and *Callithrix jacchus* (callitrichidae, primates), Harrison, Mary L., and Suzette D. Tardif. "Kin preference in marmosets and tamarins:Saguinus oedipus andCallithrix jacchus (callitrichidae, primates)." *American Journal of Physical Anthropology* 77, no. 3.

17. “Comment on the evolution of claw-like nails in callitrichids (marmosets/tamarins)”, Ford, Susan M. "Comment on the evolution of claw-like nails in callitrichids (marmosets/tamarins)." *American Journal of Physical Anthropology* 70, no. 1.

18. “Substrate determines asymmetrical gait dynamics in marmosets (*Callithrix jacchus*) and squirrel monkeys (*Saimiri boliviensis*)”, Young, Jesse W. "Substrate determines asymmetrical gait dynamics in marmosets (Callithrix jacchus) and squirrel monkeys (Saimiri boliviensis)." *American Journal of Physical Anthropology* 138, no. 4 (April 2009): 403–20.

19. “Feeding habits of marmosets: A case study of bark anatomy and chemical composition of *Anadenanthera peregrina* gum”, Francisco, Talitha Mayumi, Karina Lucas Barbosa Lopes-Mattos, Edgard Augusto de Toledo Picoli, Dayvid Rodrigues Couto, Juraci Alves Oliveira, José Cola Zanuncio, José Eduardo Serrão, Ita de Oliveira Silva, and Vanner Boere. "Feeding habits of marmosets: A case study of bark anatomy and chemical composition ofAnadenanthera peregrinagum." *American Journal of Primatology* 79, no. 3.

20. “Nasal Morphometry in Marmosets: Loss and Redistribution of Olfactory Surface Area”, Smith, Timothy D., Thomas P. Eiting, Christopher J. Bonar, and Brent A. Craven. "Nasal Morphometry in Marmosets: Loss and Redistribution of Olfactory Surface Area." *Anatomical Record* 297, no. 11.