

Hacktrix - Industrial Electronics and Environmental Sustainability

Problem Statement 1: Audio Watermarking for Copyright Protection

Overview:

This challenge revolves around the development of a MATLAB tool for audio watermarking, emphasizing copyright protection. The objective is to implement robust watermarking algorithms that can seamlessly embed copyright information into audio signals. The system's performance will be rigorously evaluated under diverse conditions, and an efficient mechanism for detecting and verifying the embedded watermark will be devised.

Objectives:

1. **Robust Copyright Embedding:** Implement advanced audio watermarking algorithms to embed copyright information securely within audio signals.
2. **Performance Evaluation:** Evaluate the system's performance under various conditions such as different audio formats, noise levels, and compression techniques.
3. **Detection and Verification Mechanism:** Develop an effective mechanism for detecting and verifying the embedded watermark, ensuring accuracy and reliability.

Technologies:

- Signal Processing in MATLAB
- Audio Processing Techniques
- Watermarking Algorithms
- Digital Signal Processing (DSP) Principles

Datasets:

Participants are encouraged to use standardized audio datasets or proprietary datasets related to audio signals and copyright information.

Evaluation Criteria:

1. **Robustness of Copyright Embedding:** The ability of the algorithm to embed copyright information securely without compromising audio quality.
2. **Effectiveness in Diverse Conditions:** The system's performance under varied audio scenarios, including different formats, noise levels, and compression types.
3. **Detection Accuracy:** Precision in detecting and verifying the embedded watermark under diverse conditions.
4. **Innovation:** Innovative approaches in implementing watermarking algorithms and addressing challenges associated with real-world audio applications.

This challenge provides an opportunity to contribute to the domain of copyright protection through cutting-edge audio watermarking techniques. Participants are encouraged to explore innovative solutions with potential applications in digital rights management and intellectual property protection.

Problem Statement 2: Human Activity Recognition using Wearable Sensors

Overview:

This challenge centers around building a machine learning model to accurately recognize and classify human activities based on data from wearable sensors. The objective is to create a system that can identify diverse physical activities, such as walking, running, sitting, or climbing stairs, using sensor data. The application could have implications in health monitoring, fitness tracking, and smart assistive technologies.

Objectives:

1. **Multi-Activity Recognition:** Develop a robust model capable of recognizing and classifying a variety of human activities accurately.
2. **Real-time Detection:** Implement a system that can provide real-time recognition of activities using streaming data from wearable sensors.
3. **Cross-Domain Applicability:** Ensure the model's effectiveness across different domains, accommodating variations in sensor types, placements, and user demographics.

Technologies:

1. **Machine Learning Frameworks:** TensorFlow, PyTorch.
2. **Sensor Fusion Techniques:** Combine data from multiple sensors to enhance the model's accuracy and robustness.
3. **Time-Series Analysis:** Apply techniques such as recurrent neural networks (RNNs) or attention mechanisms to analyze sequential sensor data effectively.

Datasets:

Teams must use the WISDM (Wireless Sensor Data Mining) dataset or any other related dataset with labeled sensor data for human activity recognition.

Evaluation Criteria:

1. **Accuracy and Precision:** The model's ability to accurately identify and classify various human activities.
2. **Real-time Performance:** The efficiency of the model in providing timely activity recognition using streaming sensor data.
3. **Adaptability:** The model's capability to adapt and perform well across diverse sensor types, placements, and user profiles.
4. **Innovative Approaches:** Novel methods for sensor fusion, feature extraction, or model architecture that contribute to improved performance.
5. **Practical Applicability:** Assess the potential practical applications, including health monitoring, fitness tracking, and assistive technologies.

Problem Statement 3: Embedded System for Hazardous Gas Detection and Alerting using ARM

Overview:

Develop an embedded system for detecting hazardous gases, such as LPG and propane, using ARM microcontrollers. The system should provide accurate real-time monitoring, display gas levels on an LCD, and trigger alerts to ensure safety in various environments.

Objectives:

1. **Robust Gas Sensors:** Select and integrate robust gas sensors capable of accurately detecting hazardous gases.
2. **ARM Programming:** Develop the firmware for ARM microcontrollers to process sensor data and control the system.
3. **Real-time Gas Level Display:** Implement a real-time display of gas levels on an LCD for continuous monitoring.
4. **Alerting System:** Design an alerting mechanism to notify users in case of elevated gas levels, ensuring a timely response.
5. **Application in Various Environments:** Modify the system to be adaptable for use in industries, homes, and offices.

Technologies:

1. **ARM Microcontrollers:** Utilize ARM-based microcontrollers for efficient processing and control.
2. **Gas Sensors:** Select appropriate gas sensors for detecting LPG and propane.
3. **LCD Display:** Implement an LCD for real-time display of gas levels.
4. **Alerting Mechanism:** Use sound or visual alerts to notify users of potential hazards.
5. **Safety Standards:** Adhere to safety standards and regulations for hazardous gas detection systems.

Evaluation Criteria:

1. **Sensor Performance:** Accuracy and reliability of the selected gas sensors.
2. **ARM Programming Quality:** Efficiency and effectiveness of the ARM microcontroller programming.
3. **Real-time Monitoring:** Continuous and accurate display of gas levels on the LCD.
4. **Alerting System Responsiveness:** Speed and reliability of the alerting mechanism.
5. **Adaptability:** Ability to modify and use the system in different environments, including industries, homes, and offices.

This problem statement focuses on the development of an embedded system for gas detection and alerting, emphasizing the importance of robust sensors, ARM programming, and real-time monitoring. The project aims to enhance safety measures in various settings, including industries, homes, and offices.

Problem Statement 4: Real-Time Translation App for Speech & Hearing Impaired Individuals

Overview:

Develop an innovative mobile application to facilitate real-time translation from multiple Indian languages and English to Sign language. The app should operate offline, addressing communication challenges faced by the Speech and hearing-impaired community. Integration of a comprehensive database for words and signs is crucial.

Objectives:

1. **Real-Time Translation:** Enable the app to provide instantaneous translation from spoken or written words in multiple languages to Sign language.
2. **Offline Functionality:** Design the app to operate without the need for an internet connection, ensuring accessibility in various settings.
3. **Comprehensive Database:** Implement a database containing a wide range of words and corresponding signs to enhance the app's vocabulary.
4. **User-Friendly Interface:** Create an intuitive and user-friendly interface for easy navigation and interaction.
5. **Customizable Input:** Allow users to input new words and signs into the database for personalized communication.

Technologies:

1. **Machine Learning:** Utilize machine learning algorithms for efficient language translation.
2. **Offline Mode:** Implement techniques for offline functionality, minimizing dependency on internet connectivity.
3. **Database Management:** Develop a robust database system to store words and corresponding signs.
4. **User Interface (UI/UX):** Design an intuitive and visually appealing interface for a positive user experience.

Evaluation Criteria:

1. **Translation Accuracy:** Precision and accuracy in translating spoken or written words into Sign language.
2. **Offline Performance:** Effectiveness of the app in offline mode without compromising translation quality.
3. **Database Coverage:** Extensiveness and relevance of the words and signs available in the database.
4. **User Interface Design:** User-friendliness and accessibility of the app interface.
5. **Customization Feature:** User's ability to input new words and signs into the database for personalized communication.

This problem statement focuses on addressing the communication challenges faced by the Speech and hearing-impaired community by developing a real-time translation app. The app aims to bridge the communication gap and enhance accessibility through offline functionality and a comprehensive database.

Problem Statement 5: Geo Tracking of Waste and Waste Quantity Index Mapping

Overview:

Cities need help in waste management, with difficulty in identifying areas requiring urgent attention for garbage pickup. The current methods, such as the Swachh Bharat App, have low adoption rates. This problem statement proposes a solution using geo-tracking, Drones, and intelligent algorithms to monitor waste areas, capture images or videos, and map high-waste areas on a city map with a Waste Quantity Index.

Objectives:

1. **Automated Waste Monitoring:** Develop a system that uses drones and vehicle-mounted cameras to automatically monitor and capture images/videos of waste areas in cities.
2. **Geo-tracking:** Implement geo-tracking for images/videos to pinpoint the exact location of waste areas, aiding in efficient waste management.
3. **Pattern Matching Algorithms:** Utilize intelligent algorithms for pattern matching to identify and map areas with high waste accumulation.
4. **Waste Quantity Index:** Create a Waste Quantity Index to approximate the level of waste in specific areas, facilitating prioritized waste management.
5. **Scalability and Lateral Scenarios:** Design the solution to be scalable and adaptable for additional use cases, such as flood monitoring, traffic monitoring, and toll booth queue monitoring.

Technologies:

1. **Drones:** Incorporate drone technology for aerial monitoring of waste areas.
2. **Image and Video Processing:** Implement algorithms for image and video processing to identify and categorize waste.
3. **Geo Tracking Systems:** Utilise geo-tracking systems to determine the precise location of waste areas.
4. **Mapping Technology:** Employ mapping technology to visualize waste areas on a city map with a Waste Quantity Index.
5. **Scalable Architecture:** Design a scalable architecture to accommodate additional scenarios and functionalities.

Beneficiaries:

1. **Government Agencies:** Enable efficient waste management and allocation of resources based on real-time data.
2. **Citizens:** Contribute to cleaner and healthier living environments.

National Impact:

The implementation of this solution benefits the nation by improving waste management, ensuring cleaner cities, and optimizing resource allocation for waste collection and disposal.

Challenges:

1. **Device Manufacturing:** Ensuring devices work seamlessly across regions and are economically viable.

2. **Security:** Addressing concerns related to the security of devices, drones, and permissions for execution in specific areas.
3. **Costs:** Managing costs associated with drones and ensuring economic viability for widespread adoption.

This problem statement addresses the critical issue of waste management by leveraging geo-tracking, drones, and intelligent algorithms for automated monitoring, mapping, and waste quantity indexing. The proposed solution aims to enhance efficiency and contribute to a cleaner and healthier living environment for citizens.

Problem statement 6: Green Blockchain for Community Carbon Credits

Overview:

As the world grapples with problems such as climate change, people seek innovative solutions to actively participate in carbon reduction efforts, traditional carbon credit systems often lack transparency and inclusivity.

This problem statement proposed the implementation of a Matlab-based green blockchain system for managing community carbon credits for managing community carbon credits, providing transparency and a community-driven platform.

Objectives:

1. **Secure and transparent transaction records:** Develop a decentralized blockchain infrastructure using advanced cryptography techniques for secure and transparent transaction records.
2. **Smart contracts:** create smart contracts in Matlab for automated issuance of carbon credit based on validated carbon reduction activities, reducing administrative overhead.
3. **Consensus algorithm:** Create smart contracts in Matlab for automated issuance of carbon credit based on validated carbon reduction activities, reducing administrative overhead
4. **Decentralized carbon credit exchange:** design a decentralized exchange within the blockchain, allowing community members to securely trade carbon credits, fostering collaboration, etc.
5. **Community engagement:** Develop a user-friendly Matlab-based dashboard for community members to track carbon credits, engage in transactions, and monitor the progress community-wise.

Technologies:

1. **Blockchain technology:** used for decentralized and tamper-proof transaction records.
2. **Cryptography:** utilize advanced cryptography techniques for secure and private transactions.
3. **Smart contracts:** implement smart contracts in Matlab for automated and transparent execution of carbon credit issuance and trading.
4. **Matlab dashboard:**
A Matlab-based dashboard for user-friendly interaction, providing real-time insights into carbon credit transactions and community progress.

Evaluation criteria:

1. **Transparency and security:** clarity in the transaction record, accessibility of blockchain data, and robustness in cryptography techniques.
2. **User-friendliness:** ease the use of Matlab software and accessibility to information to community members.
3. **Scalability:** Ability to accommodate additional scenarios and smooth integration of new functionalities.

Beneficiaries:

Residents: Empower individuals with a platform to actively contribute to carbon reduction, trade carbon credits, and benefit from sustainable practices.

Government Authorities: Enable efficient allocation of resources and monitoring of community-driven carbon reduction efforts based on transparent, real-time data

National Impact:

The implementation of this Green Blockchain for Community Carbon Credits system contributes to national efforts in climate change mitigation by fostering community-driven, transparent, and inclusive carbon reduction initiatives.

Challenges:

1. **Device Manufacturing:** Ensure devices work seamlessly across regions and are economically viable for widespread adoption.
2. **Security:** Address concerns related to the security of devices, drones, and permissions for execution in specific areas.
3. **Cost Management:** Manage costs associated with blockchain implementation, ensuring economic viability and scalability.

Problem statement 7: Advanced Hydraulic Managing for Watershed Management

Overview:

Watershed management is critical to sustainable water resource planning, flood control, and ecological preservation. Previous hydrological models often struggle to account for varied terrain, changing land patterns, and climate variability.

This problem statement proposes the development of an advanced Matlab-based hydrological model that integrates optimization techniques to address these challenges, providing effective strategies for watershed management.

Objectives:

1. **Terrain modeling:** Develop a Matlab-based terrain modeling system that accurately represents the diverse topography of the watershed.
2. **Land use change simulation:** use algorithms to simulate the change in land use over time, considering factors such as urbanization, deforestation, and agricultural practices.
3. **Climate variability integration:** it should incorporate climate variability data into the model, accounting for changing patterns.
4. **Hydrological simulation:** develop a comprehensive hydrological simulation module that integrates terrain, land use, and climate data to model the flow of water in watersheds accurately.
5. **Optimization techniques:** implement optimization algorithms within the model to identify optimal strategies for water resource management, flood control, and ecological preservation.
6. **Visualization and reporting:** Create visualization tools within Matlab to represent simulation outputs, allowing people to analyze the result effectively.

Technologies:

1. **Matlab programming:** Leverage Matlab's capabilities for scientific computing, data analysis, and visualization in the implementation of the model.
2. **Optimization toolboxes:** Utilize Matlab toolboxes to implement optimization techniques for water resource management.
3. **Geospatial data integration:** Integrate geospatial data, including terrain maps, land use data, and climate change in the Matlab model.

Beneficiaries:

1. **Water Resource Planners:** Provide planners with a powerful tool for optimizing water resource allocation and management strategies.
2. **Flood Control Authorities:** Offer insights and strategies for effective flood control measures, enhancing preparedness and response capabilities.
3. **Ecologists and Conservationists:** Aid in ecological preservation efforts by providing a tool to model and understand the impact of watershed management decisions on ecosystems.

National Impact:

The implementation of this advanced hydrological model contributes to improved watershed management practices, fostering sustainable water resource utilization, reducing flood risks, and preserving ecosystems.

Challenges:

1. **Data Integration Complexity:** Overcoming challenges associated with integrating

diverse datasets, including terrain, land use, and climate data.

2. **Algorithm Calibration:** Ensuring accurate calibration of algorithms to represent real-world hydrological processes effectively.
3. **People Engagement:** Engaging people effectively to ensure the model aligns with their needs and incorporates valuable insights.

Problem Statement 8: Smart Grid Optimization for Renewable Energy Integration

Overview:

Smart grids play an important role in optimizing energy distribution and managing the integration

of renewable energy sources. However, efficient optimization strategies are crucial to fully harness the benefits of renewable energy.

This problem statement proposes the development of a MATLAB-based smart grid optimization model, emphasizing the seamless integration of renewable energy sources.

Objectives:

1. **Renewable Energy Forecasting:** Utilize MATLAB to develop algorithms for accurate forecasting of renewable energy generation, considering variables such as solar irradiance and wind speed.
2. **Energy Storage Optimization:** Implement optimization algorithms to manage energy storage systems efficiently.
3. **Demand Response Strategies:** Develop MATLAB-based models to optimize demand response strategies, considering real-time pricing, load forecasting, and user preferences.
4. **Grid Resilience Enhancement:** Integrate optimization techniques to enhance grid resilience against disturbances, including faults and fluctuations in renewable energy generation.
5. **Integration of Distributed Energy Resources (DERs):** Use MATLAB to model and optimize the integration of various DERs, including solar panels, wind turbines, and electric vehicles, into the smart grid.
6. **Cybersecurity Measures:** Incorporate MATLAB-based cybersecurity measures to ensure the security and integrity of smart grid communications and data.

Technologies:

1. **MATLAB Programming:** Use MATLAB's extensive capabilities for mathematical modeling, optimization, and data analysis.
2. **Machine Learning for Forecasting:** Use MATLAB's machine learning toolbox for training models that improve the accuracy of renewable energy generation forecasts.
3. **Optimization Toolboxes:** Use MATLAB's optimization toolboxes to implement algorithms for optimizing energy storage, demand response, and grid resilience.
4. **Cybersecurity Measures:** Implement cybersecurity measures using MATLAB for secure communication, encryption, and threat detection.

Beneficiaries:

1. **Power Utilities:** Improve grid efficiency, reliability, and resilience, leading to better energy management and reduced operational costs.
2. **Renewable Energy Providers:** Optimize the integration of renewable energy sources into the grid, ensuring maximum utilization and contributing to grid stability.
3. **Consumers:** Benefit from optimized energy pricing, enhanced demand response programs, and increased access to clean energy.

National Impact:

The implementation of smart grid optimization contributes to a more sustainable and resilient energy infrastructure, fostering increased reliance on renewable energy sources and reducing carbon emissions.

Challenges:

1. **Data Security and Privacy:** Concerns related to the security and privacy of data in a smart grid environment.
2. **User Adoption:** Overcome challenges related to user acceptance and adoption of demand response programs and smart grid technologies.
3. **Interoperability:** Ensure seamless integration and interoperability between various components of the smart grid infrastructure.

Datasource: Use publicly available datasets on renewable energy generation, smart grid communication networks, historical pricing data, and grid topology from sources such as national energy agencies and research institutions.

Hacktrix - AI & Deep Learning

Submission Format:

- **Code Repository:** Well-documented source code of the entire tool, including the backend (model).
- **Working Prototype:** A live demonstration, preferably a comparative display where users can test the tool with their own images.
- **Technical Report:** A detailed report describing the approach, model architecture, challenges faced, and how they were overcome. The report should also include before and after examples of given challenges.

-> The above-mentioned Submission format is the same for all challenges!

Problem Statement 1: Generative AI for Image Colorization

Overview:

The goal of this challenge is to create a tool that automatically adds color to grayscale (*black and white*) images, simulating how they might have looked in real life. This tool should use computer vision and deep learning techniques to understand the content of the images and apply realistic colors.

Objectives:

- Develop a Generative model that can intelligently colorize black and white images.
- Ensure the colorization is as realistic as possible, with attention to natural color tones and lighting conditions.
- The tool should be able to handle a variety of images, from portraits and landscapes to urban scenes.

Datasets for Training:

Candidates may use CIFAR - 10, CIFAR - 100 or The Diverse 2k Resolution (DIV2K) dataset or any other datasets and are not restricted to datasets mentioned above.

-> For demonstration purposes teams may choose to use any grayscale/Binary image(s).

Technologies Recommended:

- **Deep Learning Frameworks:** TensorFlow or PyTorch for model development.
- **Image Processing:** OpenCV for initial image processing and augmentations.
- **No Front-End required!**

Evaluation Criteria:

- **Accuracy of Colorization:** How realistic and accurate the colors are in the colorized images.

- **Performance:** The efficiency of the tool in processing time and resource utilization.
- **Innovation:** Any novel approaches or techniques used in the model or overall design.

Problem Statement 2: **Predictive Modeling for Patient Health Outcomes Using Medical Imagery**

Overview:

This challenge involves developing a predictive model that leverages medical imaging data, such as CT scans, MRIs, and X-rays, to forecast patient health outcomes. The focus is on creating a system that can predict the progression of diseases, likelihood of recurrence, recovery rates, or responses to specific treatments.

Objectives:

- **Disease Progression and Outcome Prediction:** Utilize medical imagery to predict how diseases will progress and what the health outcomes for patients might be.
- **Integration of Clinical Data:** Combine imaging data with electronic health records (EHRs) to enhance prediction accuracy.
- **Interpretable Models:** Ensure that the model's predictions are interpretable by healthcare professionals, providing insights into the factors influencing the predictions.

Technologies:

- **Deep Learning Frameworks:** TensorFlow, PyTorch, for developing complex models.
- **Advanced Image Processing Techniques:** Techniques like segmentation, feature extraction, and augmentation specific to medical images.
- **Explainable AI (XAI) Tools:** Methods like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) or any method of your choice for model interpretability.
- **Federated Learning:** For privacy-preserving model training across multiple healthcare institutions.

Datasets:

Teams are free to use any open-source or licensed medical datasets of CT scans, MRI, X-ray, etc., for this problem statement.

Evaluation Criteria:

- **Prediction Accuracy:** How accurately the model predicts patient outcomes.
- **Model Robustness and Generalizability:** The model's performance across different diseases, imaging modalities, and patient demographics.
- **Interpretability:** The effectiveness of the explanation component in making the model's decisions understandable to clinicians.
- **Innovation:** Novel approaches in handling medical imagery, model development, and integration of clinical data.

Problem Statement 3: **Generative Image Stylization**

Overview:

This challenge focuses on developing a generative model for artistic image stylization. The goal is to create a system that can take input images from the webcam and transform it stylistically to resemble the work of a specific artist or art movement, going beyond basic filters to achieve a more profound, generative transformation.

Objectives:

- **High-Quality Stylization:** Develop a model that can convincingly apply complex artistic styles to input images, ensuring the output maintains both the style's essence and the input image's recognizability.
- **Diverse Style Adaptation:** The system should be capable of adapting to a range of artistic styles, potentially even allowing users to choose from various artists or movements.
- **Real-Time Processing:** Optimize the model for reasonable processing times, making it suitable for real-time applications.

Technologies:

- **Deep Learning Frameworks:** TensorFlow, PyTorch.
- **Generative Models:** GANs (Generative Adversarial Networks), style transfer algorithms, variational autoencoders.
- **Image Processing Tools:** OpenCV for pre- and post-processing of images.

Recommended Datasets:

WikiArt Dataset, a comprehensive dataset of artwork that can be used to train the model on various artistic styles. Teams may choose to use any other styling choices.

Evaluation Criteria:

- **Quality of Stylization:** The aesthetic and artistic quality of the stylized images.
- **Diversity of Styles:** The range and accuracy of different artistic styles the model can emulate.
- **Efficiency:** The model's processing time and resource utilization.

- **Creativity and Innovation:** Novel approaches in achieving deep and convincing style transformations.

Problem Statement 4: **Image Recognition through Clustering**

Overview:

This challenge involves developing a clustering algorithm that enables AI systems to identify and understand various scenes encountered. The goal is to enhance the model's decision-making systems by providing accurate scene context through clustering similar scenes.

Objectives:

- **Accurate Scene Classification:** Create a model that can cluster and classify different scenes accurately.
- **Real-Time Processing:** The algorithm must operate in real-time, given the fast-paced nature environments.

Technologies:

- **Deep Learning Frameworks:** TensorFlow, PyTorch.
- **Computer Vision Techniques:** Any Convolutional Neural Networks (CNNs) Architecture for feature extraction, unsupervised or semi-supervised clustering algorithms.
- **Real-Time Processing Tools:** CUDA, TensorRT for optimizing deep learning models for real-time performance

Datasets:

Team can use any of the mentioned dataset only. KITTI Vision Benchmark Suite, Cityscapes Dataset, Berkeley DeepDrive Dataset (BDD100K).

Evaluation Criteria:

- **Accuracy of Scene Recognition:** How effectively the model identifies and classifies different scenes/Objects.
- **Processing Speed:** The model's ability to operate in real-time, crucial for quick decision applications.
- **Robustness:** Performance across different environmental conditions and scenarios.
- **Innovativeness:** Novel approaches in clustering, feature extraction

Problem Statement 5: **Animal Classification and Detection of Species**

Overview:

This challenge focuses on developing a deep learning model for accurate classification and detection of animal species from images or videos. The goal is to create a system that can identify a wide range of species, from common to endangered, in various environments and settings, such as wildlife reserves, forests, oceans, and urban areas.

Objectives:

- **Broad Species Recognition:** Develop a model capable of recognizing and classifying a wide range of animal species with high accuracy.
- **Robust Detection in Varied Environments:** Ensure the system can detect and classify animals in different settings, including under challenging conditions like low light, motion blur, or partial obstruction.
- **Conservation and Biodiversity Tracking:** Utilize the model for applications in wildlife conservation, biodiversity monitoring, and ecological studies.

Technologies:

- **Deep Learning Frameworks:** TensorFlow, PyTorch.
- **Computer Vision Techniques:** CNNs for image classification and object detection, transfer learning to leverage pre-trained models.
- **Image Augmentation:** Techniques to increase dataset diversity and robustness, like rotation, scaling, and color adjustments.

Datasets:

Teams are required to use iNaturalist Dataset or any other related dataset

Evaluation Criteria:

- **Accuracy and Precision:** The model's ability to correctly identify and classify a wide range of animal species.
- **Performance in Different Conditions:** Robustness of the model in various environmental and image capture conditions.
- **Innovativeness:** Novel approaches in handling challenges related to dataset diversity, fine-grained classification, and real-world applicability.
- **Conservation Impact:** Potential contribution of the model to wildlife conservation and biodiversity studies.

Problem Statement 6: YOLO v8-based Traffic Sign Detection

Overview:

This challenge focuses on leveraging the latest advancements in the YOLO (You Only Look Once) architecture, specifically YOLO v8, to create a highly efficient and accurate traffic sign detection system. The goal is to develop a model that can quickly and accurately identify various traffic signs in real-time from video feeds, which is crucial for autonomous driving systems and advanced driver-assistance systems (ADAS).

Objectives:

- **Accurate Traffic Sign Recognition:** Develop a YOLO v8-based model capable of detecting and classifying a wide range of traffic signs under various conditions.
- **Real-Time Processing:** Optimize the model for high-speed, real-time detection suitable for in-vehicle systems.
- **Robustness in Diverse Conditions:** Ensure the system performs reliably in different lighting, weather, and environmental conditions.

Technologies:

Deep Learning Frameworks: Utilization of the latest YOLO architecture (YOLO v8) for object detection tasks.

Datasets:

Teams may use German Traffic Sign Recognition Benchmark (GTSRB), LISA Traffic Sign Dataset or any other custom dataset of their choice.

Evaluation Criteria:

- **Detection Accuracy:** The precision and recall of the traffic sign detection system.
- **Processing Speed:** The ability of the model to operate in real-time, a critical factor for in-vehicle systems.
- **Robustness:** The model's performance consistency across various environmental and situational conditions.
- **Innovation:** Creative approaches in adapting and optimizing YOLO v8 for the specific task of traffic sign detection.

Problem Statement 7: **Bias Detection and Mitigation in AI Models**

Overview:

This project aims to develop a comprehensive system for detecting and mitigating biases in Artificial Intelligence (AI) models, especially those used in critical decision-making areas like recruitment, credit scoring, and law enforcement. The focus is to ensure AI systems are fair, equitable, and do not perpetuate existing societal biases.

Objectives:

- **Identify and Assess Biases:** Develop tools to detect biases in AI models and datasets, whether they are related to race, gender, age, or other sociodemographic factors.
- **Mitigation Strategies:** Implement strategies to mitigate detected biases, ensuring the AI models make fair and unbiased decisions.
- **Transparency and Interpretability:** Enhance the transparency and interpretability of AI models to understand how decisions are made.

Technologies:

- **Machine Learning Tools:** TensorFlow, PyTorch, scikit-learn for model development and analysis.
- **Fairness Assessment Libraries:** AI Fairness 360 (AIF360), Fairlearn, or similar tools for bias detection and mitigation.
- **Data Visualization Tools:** To illustrate biases in data and model decisions.

Framework Components:

- **Bias Detection Mechanisms:** Algorithms and methodologies to detect various types of biases in datasets and model predictions.
- **Bias Mitigation Techniques:** Approaches like re-sampling, re-weighting, or model adjustment to mitigate identified biases.
- **Model Evaluation Protocols:** Standardized procedures to evaluate AI models for fairness alongside traditional performance metrics.

Evaluation Criteria:

- **Effectiveness in Bias Detection:** The toolset's ability to accurately identify biases in various types of AI models and datasets.
- **Mitigation Success:** How effectively the project's methodologies can mitigate detected biases without significantly compromising the model's performance.
- **Usability and Accessibility:** The ease with which AI practitioners can use these tools and integrate them into their workflows.
- **Innovation and Scalability:** Novel approaches in bias detection and mitigation, and the scalability of solutions to different types of AI models and applications.

Problem Statements 8: **Energy Consumption Forecasting for Smart Grids**

Overview:

This project entails the development of a sophisticated machine learning model to accurately forecast energy consumption for smart grid management. The model should predict energy usage at various scales, such as individual households, commercial buildings, and entire city grids, aiding in efficient energy distribution and management.

Objectives:

- **Accurate and Scalable Predictions:** Develop a model that can accurately forecast energy consumption at different scales and adapt to various consumption patterns.
- **Integration of Diverse Data Sources:** Incorporate data from multiple sources, including weather conditions, user behavior patterns, historical consumption data, and real-time grid performance.
- **Demand Response Optimization:** Use the forecasts to optimize demand response strategies, helping to balance the load on the grid during peak periods.

Technologies:

- Machine Learning Frameworks: TensorFlow, PyTorch for building and training regression models.
- Time-Series Analysis: Specialized techniques and models for time-series forecasting, such as ARIMA, LSTM networks, Prophet etc.,
- Big Data Platforms: Technologies like Apache Hadoop or Spark for processing large-scale datasets.

Datasets:

Smart Meter Data or any other of your choice

Evaluation Criteria:

- Forecast Accuracy: The precision of the energy consumption forecasts at different time scales and conditions.
- Scalability and Robustness: The model's ability to handle different scales of data and maintain performance under varying conditions.
- Integration of Diverse Data Sources: Effectiveness in integrating and utilizing data from multiple sources.
- Real-Time Processing Capability: The efficiency and responsiveness of the model in a real-time scenario.

Problem Statement 9: **Housing Cost Estimation in Chennai**

Overview:

This project involves developing a regression model to estimate housing costs in Chennai, India, based on various factors such as location, property size, amenities, neighborhood characteristics, and market trends. The objective is to provide accurate price predictions that can assist buyers, sellers, and real estate professionals in making informed decisions.

Objectives:

- Accurate Price Prediction: Develop a model that can predict housing prices in Chennai with high accuracy, taking into account a wide range of influencing factors.
- Localization and Contextualization: Ensure the model is tailored to the specific real estate market dynamics and cultural context of Chennai.
- Feature Integration: Incorporate diverse data points like infrastructure developments, proximity to key locations, local economic indicators, and historical price trends.

Technologies:

- Machine Learning Frameworks: Tools like scikit-learn, TensorFlow, or PyTorch for building regression models.
- Data Processing and Analysis Tools: Pandas, NumPy, and SQL for data manipulation and analysis.
- Geospatial Analysis Tools: GIS software or APIs for incorporating location-based features.

Dataset:

A custom dataset will be provided from R&D Domain.

Evaluation Criteria:

- Prediction Accuracy: How closely the model's predictions align with actual housing prices.
- Feature Relevance and Interpretation: The model's ability to utilize locally relevant features effectively and provide interpretable insights.
- Robustness and Generalizability: The model's performance across different property types and market conditions.
- Innovation and Usability: Creativity in model design and potential ease of use for end-users.

Hacktrix - Web and App Development

Problem Statement 1: Enhancing Customer Relationship Management (CRM)

Overview:

This project aims to refine a CRM application to improve customer engagement and satisfaction for businesses in diverse sectors. The focus will be on personalizing customer interactions, predicting customer needs, and automating service processes to boost retention and sales.

Objectives:

- Personalized Customer Interaction: Tailor the CRM to predict and meet individual customer preferences and behavior.
- Efficiency in Service: Utilize AI to streamline customer service operations, reducing response time and improving resolution rates.
- Data-Driven Insights: Leverage data analytics to provide actionable insights into customer behavior and trends.

Technologies:

- CRM Software: Custom or established CRM platforms enhanced with AI capabilities.
- Analytics and AI Tools: Integration of tools like R, Python, and machine learning libraries for predictive analytics.
- Automation Technologies: Incorporating chatbots and process automation tools to improve customer service workflows.

Dataset:

Datasets shall include customer interaction logs, sales data, and service records from business operations.

Evaluation Criteria:

- Customer Satisfaction: Improvement in customer service ratings and feedback.
- Service Efficiency: Metrics related to service response times and issue resolution effectiveness.
- Business Impact: Evaluation of sales growth and customer retention attributable to CRM enhancements.
- User Adoption: Rate of adoption and ease of use for customer service representatives and customers.
- This format outlines a clear plan for enhancing a CRM application, with an emphasis on practical objectives, technological tools, and evaluation metrics to measure success.

Problem Statement 2: **Streamlined Health & Wellness App Development**

Overview:

The aim is to create a comprehensive health and wellness application that not only tracks various health metrics but also provides personalized wellness plans. The app should offer integration with wearable devices, expert advice, and community support to encourage a healthier lifestyle.

Objectives:

- **User-Centric Health Tracking:** Develop features that allow users to monitor vital health metrics and daily activities.
- **Customized Wellness Programs:** Integrate AI to design personalized fitness and nutrition plans based on user data.
- **Community Engagement:** Foster a supportive environment through social features that motivate and engage users.

Technologies:

- **Mobile Development Platforms:** Use cross-platform tools like React Native or Flutter for app development.
- **Health Data Analytics:** Employ analytics engines to process health data and generate insights.
- **Integration APIs:** APIs to integrate with various health wearables and external databases for comprehensive tracking.

Dataset:

The app will utilize user-generated data, input from wearable devices, and expert health databases.

Evaluation Criteria:

- **User Engagement:** Track active users and engagement levels with the app's features.
- **Health Outcome Improvements:** Assess the tangible health benefits users gain from the app.
- **Technical Performance:** Ensure the app's responsiveness, data accuracy, and uptime.
- **Market Viability:** Consider user feedback and adoption rates post-launch to gauge market fit.

Problem Statement 3: **Eco-Friendly Transportation Routing App**

Overview:

This initiative will focus on developing an eco-friendly routing application aimed at reducing carbon footprints by optimizing travel routes. It will provide users with the most sustainable travel options, considering factors like fuel efficiency, traffic congestion, and availability of public transportation.

Objectives:

- Sustainable Route Optimization: Implement algorithms that find the most eco-friendly routes for various modes of transport.
- Carbon Footprint Tracking: Enable users to track and minimize their environmental impact over time.
- Public Transit Integration: Seamlessly incorporate public transportation schedules to encourage the use of more sustainable travel options.

Technologies:

Mapping APIs: Use platforms like Google Maps or OpenStreetMap for accurate routing capabilities.

Data Analytics: Utilize tools for analyzing traffic patterns and environmental data to optimize routes.

Cloud Services: Leverage cloud computing for scalable and reliable app performance.

Dataset:

Data should include traffic patterns, public transit schedules, vehicle emissions data, and user travel preferences.

Evaluation Criteria:

- Environmental Impact: Measure the reduction in carbon emissions achieved through the app.
- User Adoption: Evaluate the growth in user base and active users.
- App Performance: Assess the accuracy of routing and real-time updates.
- User Feedback: Collect user feedback for continuous improvement of the app's features.