HUMAN ACTIVITY DETECTION USING MMWAVE SENSOR

K.RAJA NIRMAL

Guide: Asst Prof **Mrs.S.ASHA**

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
MASTER OF ENGINEERING

SETHU INSTITUTE OF TECHNOLOGY

PULLOOR, KARIAPATTI-626 115.

ABSTRACT

Human Activity Recognition (HAR) is becoming increasingly useful for applications such as well-being monitoring and personalizing smart spaces. Traditional methods for HAR often require wearable devices or cameras. The former is not feasible for every environment and the latter has strong privacy concerns. The mmWave radar has been shown to be a promising alternative as it does not imply the same privacy concerns and does not require the users to have wearable devices. In this paper we have used a low- cost mmWave radar to generate micro-Doppler spectrograms to ultimately classify different activities. For this, multiple classifiers and methods of spectrogram filtering have been examined. Finally, a Time-Distributed Convolutional Neural Network in conjunction with a Bi-Directional Long Short-Term Memory has attained an average accuracy of 99.62% on a dataset of 5 activities, involving 2 participants.

CHAPTER I

1. INTRODUCTION

Knowing the location of people and what they are doing can be of importance to a variety of applications. Elderly requires constant monitoring to allow them to live an independent lifestyle, while also ensuring their well-being. Smart spaces can better respond to personalized demands, such as heating, lighting, security management and sound selection using this information, increasing comfort and energy efficiency. Currently user, identification and activity tracking methods include using visual camera's, WiFi and device-based solutions, where users are identified by their smartphone, watch or ID-card. While camera's achieve great performance in these tasks, they do have the downsides of light-condition reliance, as well as privacy concerns. Cameras are intrusive and often poorly received in both domestic and commercial settings. In addition, cameras in a hospital have been used to spy on female patients. WiFi-based solutions require a separate transmitter and receiver, and only work when the target is located between them, limiting their usability. Moreover, device-based solutions require human effort and assume inseparability of the device and their users, properties that are ultimately undesired for seamless integration. The mmWave radar is a small device, operating as a transceiver using electromagnetic waves. In addition, its waves can penetrate thin layers of some materials, allowing it to be placed inside furniture or walls. These properties can make the mmWave radar a better fit for user, identification and activity tracking purposes than the aforementioned solutions. Thus, in this paper the efficacy of the mmWave radar will be tested. While it can be used for user-tracking, user identification and activity.

CHAPTER-II

2. MACHINE LEARNING:

Machine learning is a transformative field within artificial intelligence (AI) that focuses on developing algorithms and models capable of learning from data, identifying patterns, and making data-driven predictions or decisions. Unlike traditional programming, where explicit instructions are provided, machine learning systems acquire knowledge from experience. This field has gained significant prominence due to its ability to tackle complex problems and provide solutions across various domains.

Machine learning has evolved in response to the growing availability of data, increased computational power, and advances in algorithmic research. Its applications span a wide range of industries, including healthcare, finance, e-commerce, autonomous vehicles, natural language processing, and image recognition. At its core, machine learning seeks toenable computers to generalize from data, adapt to new information, and improve their performance over time.

Types of Machine Learning:

There are three fundamental categories of machine learning: supervised learning, unsupervised learning, and reinforcement learning.

Supervised Learning:

Supervised learning is a prevalent approach in machine learning. In this paradigm, the algorithm is provided with a labeled dataset, consisting of input samples paired with their corresponding output labels. The primary objective is to learn a mapping from inputs to outputs so that the algorithm can make accurate predictions or classifications on new, unseen data. Examples of supervised learning tasks include email spam detection, sentiment analysis, and image classification. Algorithms commonly used in supervised learning include linear regression, decision trees, random forests, and deep neural networks.

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).
In the real-world, supervised learning can be used for Risk Assessment, Image classification, Fraud Detection, spam filtering, etc.

Supervised Learning Works:

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

- If the given shape has four sides, and all the sides are equal, then it will be labelledas a **Square**.
- If the given shape has three sides, then it will be labelled as a **triangle**.
- If the given shape has six equal sides then it will be labelled as **hexagon**.

Now, after training, we test our model using the test set, and the task of the model is toidentify the shape.

The machine is already trained on all types of shapes, and when it finds a new shape, itclassifies the shape on the bases of a number of sides, and predicts the output.

Steps Involved in Supervised Learning:

- First Determine the type of training dataset
- Collect/Gather the labelled training data.
- Split the training dataset into training dataset, test dataset, and validation dataset.
- Determine the input features of the training dataset, which should have enoughknowledge so that the model can accurately predict the output.
- Determine the suitable algorithm for the model, such as support vector machine,decision tree, etc.
- Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.
- Evaluate the accuracy of the model by providing the test set. If the model predictsthe correct output, which means our model is accurate.

Types of supervised Machine learning Algorithms:

Supervised learning can be further divided into two types of problems:

i. Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some

popular Regression algorithms which come under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

ii. Classification

Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

Spam Filtering,

- Random Forest
- Decision Trees
- Logistic Regression
- Support vector Machines

Random Forest is a powerful ensemble learning method used in machine learning for both classification and regression tasks. It's a versatile and robust algorithm known for its high predictive accuracy and resistance to overfitting. Let's dive into the details of Random Forest:

Ensemble Learning: Random Forest is part of the ensemble learning family, which combines the predictions of multiple machine learning algorithms to improve overall performance. Instead of relying on a single model, ensemble methods leverage the wisdom of the crowd to make more accurate predictions.

Decision Trees as Building Blocks: At the heart of a Random Forest are decision trees. Decision trees are simple yet effective models that recursively split the data into subsets based on feature values. Each branch of the tree represents a decision or a rule, and the leaves contain the final predictions. However, individual decision trees can be prone to overfitting and high variance.

Random Forest Construction: A Random Forest is constructed by creating a collection of decision trees. Here's how it works:

1. **Bootstrapped Sampling:** To create diversity among the individual trees, Random Forest employs bootstrapped sampling. It randomly selects subsets of the training data with replacement. This means that some data points may be sampled multiple times, while others may not be included at all in a given tree.
2. **Random Feature Selection:** In addition to sampling data points, Random Forest randomly selects a subset of features (columns) for each tree. This ensures that not all features are considered for every decision, reducing the risk of overfitting and promoting model diversity.

3. **Decision Tree Construction:** Each tree in the forest is constructed independently, using the bootstrapped sample and the randomly selected features. Typically, decision trees are grown until a certain depth or until a stopping criterion is met, such as a minimum number of samples per leaf node.
4. **Voting or Averaging:** For classification tasks, the predictions of individual trees are combined through a majority vote (each tree "votes" for a class), while for regression tasks, the predictions are averaged. This ensemble approach helps improve accuracy and reduce the impact of outliers or noise.

Advantages of Random Forest:

1. **High Accuracy:** Random Forest often delivers high predictive accuracy due to its ensemble nature. It tends to generalize well to unseen data.
2. **Robustness:** It is less susceptible to overfitting compared to single decision trees, making it a robust choice for various types of datasets.
3. **Feature Importance:** Random Forest can provide insights into feature importance. By analyzing how much each feature contributes to the model's predictions, you cangain a better understanding of your data.
4. **Works for Classification and Regression:** Random Forest can be used for both classification and regression tasks, making it versatile.

Considerations:

1. **Computational Complexity:** Training a Random Forest with a large number of trees can be computationally intensive.
2. **Hyperparameter Tuning:** Like many machine learning algorithms, Random Forest has hyperparameters that need to be tuned for optimal performance.
3. **Interpretability:** While Random Forest provides feature importance, interpretingindividual trees in the forest can be challenging due to their complexity.

In summary, Random Forest is a robust and versatile ensemble learning method that is widely used in practice for its ability to deliver accurate predictions while mitigating overfitting. It's a valuable tool in the machine learning toolbox for a range of applications.

Advantages of Supervised learning:

- With the help of supervised learning, the model can predict the output on the basis of prior experiences.
- In supervised learning, we can have an exact idea about the classes of objects.
- Supervised learning model helps us to solve various real-world problems such asfraud detection, spam filtering, etc.

Disadvantages of supervised learning:

- Supervised learning models are not suitable for handling the complex tasks.
- Supervised learning cannot predict the correct output if the test data is differentfrom the training

dataset.

- Training required lots of computation times.
- In supervised learning, we need enough knowledge about the classes of object.

Unsupervised Learning:

Unsupervised learning deals with unlabeled data, where there are no predefined output labels. Instead, the algorithm's goal is to identify patterns, relationships, or structures within the data. Common unsupervised learning tasks include clustering, where similar data points are grouped together, and dimensionality reduction, which aims to reduce the complexity of data while preserving its essential information. K-means clustering, hierarchical clustering, and principal component analysis (PCA) are widely used techniques in unsupervised learning.

In the previous topic, we learned supervised machine learning in which models are trained using labeled data under the supervision of training data. But there may be many cases in which we do not have labeled data and need to find the hidden patterns from the given dataset. So, to solve such types of cases in machine learning, we need unsupervised learning techniques.

What is Unsupervised Learning?

As the name suggests, unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes placein the human brain while learning new things. It can be defined as:

Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision.

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset,group that data according to similarities, and represent that dataset in a compressedformat.

Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

Why use Unsupervised Learning?

Below are some main reasons which describe the importance of Unsupervised Learning:

- Unsupervised learning is helpful for finding useful insights from the data.

- Unsupervised learning is much similar as a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which makeunsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so tosolve such cases, we need unsupervised learning.

Working of Unsupervised Learning

Here, we have taken an unlabeled input data, which means it is not categorized and corresponding outputs are also not given. Now, this unlabeled input data is fed to the machine learning model in order to train it. Firstly, it will interpret the raw data to find the hidden patterns from the data and then will apply suitable algorithms such as k-means clustering, Decision tree, etc.

Once it applies the suitable algorithm, the algorithm divides the data objects into groups according to the similarities and difference between the objects.

Types of Unsupervised Learning Algorithm:

The unsupervised learning algorithm can be further categorized into two types of problems:

- **Clustering**: Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities betweenthe data objects and categorizes them as per the presence and absence of those commonalities.
- **Association**: An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis.

Unsupervised Learning algorithms:

Below is the list of some popular unsupervised learning algorithms:

- K-means clustering
- KNN (k-nearest neighbors)
- Hierarchal clustering
- Anomaly detection
- Neural Networks

- Principle Component Analysis
- Independent Component Analysis
- Apriori algorithm
- Singular value decomposition

K-means clustering is a popular unsupervised machine learning algorithm used for grouping data points into clusters based on their similarity. The algorithm works by iteratively partitioning data into clusters in such a way that the data points within each cluster are more similar to each other than to those in other clusters. It does this by findingcluster centroids, which are representative points that minimize the sum of squared distances from data points to their respective centroid.

K-means clustering is widely applied in various fields, such as customer segmentation, image compression, anomaly detection, and recommendation systems. One of its key advantages is its simplicity and efficiency, making it suitable for large datasets. However, it has some limitations, including sensitivity to the initial placement of centroids and the need to specify the number of clusters (K) in advance, which can sometimes be challenging in practice. Still, it remains a valuable tool for exploratory data analysis and pattern recognition in many data science and machine learning tasks.

K-nearest neighbors (KNN) is a supervised machine learning algorithm used for classification and regression tasks. It's a simple yet effective instance-based or lazy learning algorithm that makes predictions based on the similarity between a data point andits k-nearest neighbors in the training dataset.

In KNN:

- **K Parameter**: K is a user-defined hyperparameter that represents the number ofnearest neighbors to consider when making predictions. A smaller value of K makes the algorithm more sensitive to noise in the data, while a larger value of K can smooth out predictions but may lead to biases.
- **Distance Metric**: KNN relies on a distance metric (such as Euclidean distance, Manhattan distance, or others) to measure the similarity between data points in the feature space. The choice of distance metric can significantly impact the algorithm'sperformance.
- **Classification**: In KNN classification, the algorithm counts the number of data points in each class among the k-nearest neighbors of the data point to be classified.It assigns the class that appears most frequently among those neighbors to the new data point.
- **Regression**: In KNN regression, the algorithm calculates the average (or weighted average) of the target values of the k-nearest neighbors to predict the target valuefor the new data point.
- **Scaling**: Feature scaling is often important in KNN, as it's sensitive to the scale of features. Standardizing or normalizing features to have similar scales can improve the algorithm's performance.
- **Curse of Dimensionality**: KNN's performance can degrade as the dimensionality ofthe feature space increases. This is known as the "curse of dimensionality" because distances between points become less meaningful in high-dimensional spaces. Dimensionality reduction techniques or feature selection can help mitigate thisissue.
- **Decision Boundary**: KNN's decision boundary can be complex and non-linear, depending on the data distribution and the choice of K. It can adapt well to irregularly shaped clusters in the data.

- **Pros and Cons**: KNN's simplicity and ease of implementation make it a valuable tool for quick and intuitive data exploration. However, it can be computationally expensive for large datasets, especially in high-dimensional spaces. Additionally, it doesn't learn explicit patterns from the data during training, making it lessinterpretable than some other algorithms.

K-nearest neighbors is often used in scenarios where data may not have a clear underlyingmathematical model or when interpretability is not the primary concern. It's commonly employed in recommendation systems, image classification, anomaly detection, and imputing missing data. The choice of K and the distance metric should be carefully tuned based on the specific problem and dataset to achieve optimal performance.

Reinforcement Learning:

Reinforcement learning is a different paradigm where agents interact with an environment and learn through trial and error. Agents take actions and receive feedback in the form of rewards or penalties, which guide their decision-making. The objective is to learn a policy that maximizes the cumulative reward over time. Reinforcement learning finds applications in robotics, game playing, autonomous systems, and recommendation systems. Key components include the agent, environment, actions, rewards, and exploration strategies. Algorithms like Q-learning and deep reinforcement learning methods (e.g., Deep Q-Networks) are commonly employed in this domain.

Reinforcement learning is an area of Machine Learning. It is about taking suitable actionto maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervisedlearning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

Reinforcement Learning (RL) is the science of decision making. It is about learning the optimal behavior in an environment to obtain maximum reward. In RL, the data is accumulated from machine learning systems that use a trial-and-error method. Data is not part of the input that we would find in supervised or unsupervised machine learning.

Reinforcement learning uses algorithms that learn from outcomes and decide which action to take next. After each action, the algorithm receives feedback that helps it determine whether the choice it made was correct, neutral or incorrect. It is a good technique to use for automated systems that have to make a lot of small decisions without human guidance.

Reinforcement learning is an autonomous, self-teaching system that essentially learns by trial and error. It performs actions with the aim of maximizing rewards, or in other words, it is learning by doing in order to achieve the best outcomes.

Example:

The problem is as follows: We have an agent and a reward, with many hurdles in between.The agent is supposed to find the best possible path to reach the reward. The following problem explains the problem more easily.

The goal of the robot is to get the reward that is the diamond and avoid the hurdles that arefired. The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles. Each right step will give the robot a reward and each wrong step will subtract the reward of the robot. The total reward will be calculated when it reaches the final reward that is the diamond.

Main points in Reinforcement learning –

- Input: The input should be an initial state from which the model will start
- Output: There are many possible outputs as there are a variety of solutions to a particular problem
- Training: The training is based upon the input, The model will return a state and theuser will decide to reward or punish the model based on its output.
- The model keeps continues to learn.
- The best solution is decided based on the maximum reward.

Types of Reinforcement:

There are two types of Reinforcement:

- **Positive**: Positive Reinforcement is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behavior. In other words, it has a positive effect on behavior.
Advantages of reinforcement learning are:
  - Maximizes Performance
  - Sustain Change for a long period of time
  - Too much Reinforcement can lead to an overload of states which can diminish the results
- **Negative**: Negative Reinforcement is defined as strengthening of behavior because a negative condition is stopped or avoided.
Advantages of reinforcement learning:
  - Increases Behavior
  - Provide defiance to a minimum standard of performance
  - It Only provides enough to meet up the minimum behavior

Elements of Reinforcement Learning

Reinforcement learning elements are as follows:

i. Policy

ii. Reward function

iii. Value function

iv. Model of the environment

Policy: Policy defines the learning agent behavior for given time period. It is a mapping from perceived states of the environment to actions to be taken when in those states.

Reward function: Reward function is used to define a goal in a reinforcement learning problem.A reward function is a function that provides a numerical score based on the stateof the environment.

Value function: Value functions specify what is good in the long run. The value of a stateis the total amount of reward an agent can expect to accumulate over the future, starting from that state.

Model of the environment: Models are used for planning.

Credit assignment problem: Reinforcement learning algorithms learn to generate an internal value for the intermediate states as to how good they are in leading to the goal.The learning decision maker is called the agent. The agent interacts with the environment that includes everything outside the agent.

The agent has sensors to decide on its state in the environment and takes action that modifies its state.

The reinforcement learning problem model is an agent continuously interacting with an environment. The agent and the environment interact in a sequence of time steps. At each time step t, the agent receives the state of the environment and a scalar numerical reward for the previous action, and then the agent then selects an action.

Reinforcement learning is a technique for solving Markov decision problems. Reinforcement learning uses a formal framework defining the interaction between a learning agent and its environment in terms of states, actions, and rewards. This framework is intended to be a simple way of representing essential features of the artificialintelligence problem.

Various Practical Applications of Reinforcement Learning –

- RL can be used in robotics for industrial automation.
- RL can be used in machine learning and data processing
- RL can be used to create training systems that provide custom instruction and materials according to the requirement of students.

Application of Reinforcement Learnings:

i. Robotics: Robots with pre-programmed behavior are useful in structured environments, such as the assembly line of an automobile manufacturing plant, where the task is repetitive in nature.

ii. A master chess player makes a move. The choice is informed both by planning, anticipating possible

replies and counter replies.

iii. An adaptive controller adjusts parameters of a petroleum refinery's operation in realtime.

RL can be used in large environments in the following situations:

- A model of the environment is known, but an analytic solution is not available;
- Only a simulation model of the environment is given (the subject of simulation- based optimization)
- The only way to collect information about the environment is to interact with it.

Advantages and Disadvantages of Reinforcement Learning:Advantages of Reinforcement learning:

i. Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.

ii. The model can correct the errors that occurred during the training process.

iii. In RL, training data is obtained via the direct interaction of the agent with the environment

iv. Reinforcement learning can handle environments that are non-deterministic, meaning that the outcomes of actions are not always predictable. This is useful in real-world applications where the environment may change over time or is uncertain.

v. Reinforcement learning can be used to solve a wide range of problems, including those that involve decision making, control, and optimization.

vi. Reinforcement learning is a flexible approach that can be combined with other machine learning techniques, such as deep learning, to improve performance.

Disadvantages of Reinforcement learning:

i. Reinforcement learning is not preferable to use for solving simple problems.

ii. Reinforcement learning needs a lot of data and a lot of computation

iii. Reinforcement learning is highly dependent on the quality of the reward function. If the reward function is poorly designed, the agent may not learn the desired behavior.

iv. Reinforcement learning can be difficult to debug and interpret. It is not always clear why the agent is behaving in a certain way, which can make it difficult to diagnose and fix problems.

Machine learning involves several key concepts and challenges:

Data Preprocessing:

Data quality is paramount in machine learning. Raw data often requires preprocessing, including cleaning, normalization, and feature engineering, to make it suitable for modeling. Insufficient or noisy data can significantly impact the performance of machine learning models.

Model Selection:

Choosing the appropriate machine learning algorithm or model for a specific task is critical. The selection depends on the nature of the data, the problem's complexity, and theavailable resources. Hyperparameter tuning and cross-validation are techniques used to optimize model performance.

Overfitting and Underfitting:

One of the fundamental challenges in machine learning is finding the right balance between overfitting and underfitting. Overfitting occurs when a model learns to memorize the training data, resulting in poor generalization to unseen data. Underfitting, on the otherhand, indicates that the model is too simple to capture the underlying patterns in the data. Techniques such as regularization and validation sets help mitigate these issues.

Evaluation Metrics:

To assess the performance of machine learning models, various evaluation metrics are employed. These metrics differ depending on the type of problem (classification, regression, clustering, etc.) and include accuracy, precision, recall, F1-score, mean squared error, and others.

Ethical Considerations:

As machine learning systems increasingly influence decision-making processes, ethical considerations have become paramount. Bias in data and algorithms, privacy concerns,and fairness issues are some of the ethical challenges that need to be addressed in machinelearning applications.

In conclusion, machine learning is a multifaceted field with diverse applications and challenges. Understanding its core principles, types, and key concepts is essential for harnessing its power and responsibly deploying machine learning solutions across various domains.

Decision trees are a popular machine learning algorithm used for both classification and regression tasks. They are a type of supervised learning algorithm that can be used for predictive modeling. Decision trees are particularly attractive because they are easy to understand and interpret, making them a valuable tool for both beginners and experts inthe field of machine learning and data science.

Here are some key characteristics and concepts related to decision trees:

- **Tree Structure:** A decision tree is a hierarchical structure that resembles an upside-down tree. It consists of nodes and branches. The top node is called the "root," and it represents the initial decision or question. The nodes further down the tree are called "internal nodes," and the end nodes are called "leaves" or "terminal nodes." Internal nodes represent decisions or tests on specific features, while leaves represent the predicted outcomes or classes.
- **Splitting:** At each internal node of the tree, a decision is made to split the data into two or more

subsets based on a specific feature and a threshold value. This process continues recursively until a stopping criterion is met, such as a maximum treedepth or a minimum number of data points in a leaf node.

- **Decision Making:** To make a prediction for a new data point, you start at the root node and traverse down the tree by following the decision rules at each internalnode until you reach a leaf node. The class or value associated with the leaf node is the prediction.
- **Criterion:** Decision trees use various criteria to determine the best feature and threshold for splitting the data at each node. Common criteria include Gini impurity for classification tasks and mean squared error for regression tasks. The goal is to reduce impurity or error as much as possible with each split.
- **Pruning:** Decision trees can be prone to overfitting, where they become too complex and perform well on training data but poorly on unseen data. Pruning is a technique used to simplify a tree by removing nodes that do not contribute significantly to improving predictive performance. This helps reduce overfitting andimproves the model's generalization.
- **Ensemble Methods:** Decision trees can be combined into ensemble methods like Random Forests and Gradient Boosting to enhance predictive accuracy and reduce overfitting. These methods involve training multiple decision trees and aggregating their predictions.
- **Categorical and Numerical Features:** Decision trees can handle both categorical and numerical features. For categorical features, the tree performs a binary split for each category. For numerical features, it selects a threshold value and splits the dataaccordingly.
- **Interpretability:** One of the main advantages of decision trees is their interpretability. You can easily visualize and understand the decision-making process of a tree, which is useful for explaining model predictions to stakeholders.
- **Disadvantages:** Decision trees can be sensitive to small variations in the data, leading to different trees with similar performance. They can also be biased if a class is dominant in the dataset. Addressing these issues often requires techniques like randomization or balancing the dataset.

In summary, decision trees are a versatile and interpretable machine learning algorithm used for both classification and regression tasks. They are a fundamental building block for more advanced ensemble methods and can be a valuable tool in a data scientist's toolkit.

Logistic regression is a statistical and machine learning technique used for binary classification and sometimes for multiclass classification problems. It's a type of regression analysis that models the probability of a binary outcome (0 or 1) based on one or more predictor variables. Despite its name, logistic regression is used for classification, not regression.

Here are some key characteristics and concepts related to logistic regression:

- **Logit Function**: Logistic regression models the relationship between the binary dependent variable (target) and one or more independent variables (features) using the logistic function, also known as the sigmoid function. The logistic functionmaps any real-valued number to a value between 0 and 1, which can be interpreted as a probability.
- **Binary Classification**: Logistic regression is primarily used for binary classification problems, where the target variable has two classes, typically labeled as 0 and 1. For example, it can be used to predict whether an email is spam (1) ornot spam (0) based on features like the presence of certain keywords, sender information, etc.
- **Multiclass Classification**: Logistic regression can be extended to handle multiclass classification problems using techniques such as one-vs-all (OvA) or softmax regression. In OvA, you train multiple binary

logistic regression models, one for each class, and then combine their outputs to make a multiclass prediction. Softmax regression, on the other hand, directly models the probabilities of multiple classes using the softmax function.

- **Logistic Function**: The logistic function (sigmoid function) is defined as:

i. where z is a linear combination of the predictor variables, and σ(z) represents theprobability of the positive class (1). The logistic function "s"-shaped curveallows logistic regression to model the probability of the positive class as a function of the predictor variables.

ii. **Coefficients and Odds Ratio:** In logistic regression, you estimate coefficients (weights) for each predictor variable. These coefficients represent the change in the log-odds of the event happening (positive class) for a unit change in the predictor variable. The odds ratio, which is the exponent of the coefficient, represents how the odds of the event change for a unit change in the predictor.

iii. **Maximum Likelihood Estimation (MLE):** The logistic regression model is typically trained using MLE to find the coefficients that maximize the likelihood of the observed data given the model. It's a common optimization technique usedto fit logistic regression models to the data.

iv. **Threshold**: To make binary predictions, a threshold (usually 0.5) is applied to the predicted probabilities. If the probability is greater than or equal to the threshold, the data point is classified as 1; otherwise, it's classified as 0. You can adjust the threshold to control the trade-off between precision and recall in classification.

v. **Evaluation Metrics:** Common evaluation metrics for logistic regression models include accuracy, precision, recall, F1-score, ROC curve, and AUC-ROC. These metrics help assess the model's performance on the classification task.

Logistic regression is widely used in various fields, including healthcare (e.g., disease prediction), marketing (e.g., customer churn prediction), and finance (e.g., credit risk assessment), due to its simplicity, interpretability, and effectiveness for binary and multiclass classification problems. It serves as a foundational algorithm in the field of machine learning and statistics.

Support Vector Machines (SVMs) are a powerful and versatile machine learning algorithm used for both classification and regression tasks. They are particularly well- suited for binary classification problems but can be extended to handle multi-class classification as well.

Here are some key characteristics and concepts related to Support Vector Machines:

- **Binary Classification**: SVMs are primarily used for binary classification, where thegoal is to separate data points into two classes. The algorithm finds a hyperplane (a decision boundary) that maximizes the margin between the two classes. Data points that are closest to the decision boundary and influence its position are called supportvectors.

- **Margin**: The margin is the distance between the decision boundary (hyperplane)and the nearest data points from each class. SVM aims to maximize this margin while maintaining correct classification. A larger margin indicates better generalization to unseen data.

- **Hyperplane**: In a two-dimensional feature space, the decision boundary is a straight line (a hyperplane). In higher-dimensional spaces, it becomes a hyperplane. The goal is to find the hyperplane that best separates the classes.

- **Kernel Trick**: SVMs can handle non-linearly separable data by mapping the original features into a higher-dimensional space using a kernel function. The most common kernel functions include the linear kernel, polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel. The choice of kernel depends on the problem's characteristics.

- **C Parameter**: SVMs have a hyperparameter called "C" that controls the trade-off between maximizing the margin and minimizing classification errors. A smaller C value leads to a larger margin but may allow some misclassifications, while a larger C value results in a smaller margin but fewer misclassifications.

- **Soft Margin SVM**: In cases where data is not perfectly separable, SVMs can be used with a "soft margin." Soft margin SVM allows for some misclassification of data points to achieve a balance between maximizing the margin and minimizing classification errors.

- **Support Vectors**: Support vectors are the data points that lie closest to the decision boundary and play a crucial role in defining the margin. These data points have nonzero Lagrange multipliers (alpha values) in the SVM optimization problem.

- **Multi-Class Classification**: While SVM is originally designed for binary classification, it can be extended to multi-class classification using techniques such as one-vs-all (OvA) or one-vs-one (OvO) classification. In OvA, multiple binary SVM classifiers are trained, one for each class, and the class with the highest confidence is selected as the final prediction. In OvO, a binary classifier is trained for each pair of classes, and a majority voting scheme is used to make predictions.

- **Regression (SVR)**: SVMs can also be used for regression tasks, where the goal is to predict a continuous target variable. Support Vector Regression (SVR) uses the same principles as SVM for classification but applies them to regression problems.

- **Regularization**: SVMs inherently perform regularization because they aim to maximize the margin. This helps in reducing overfitting, making SVMs effective even with small datasets.

Support Vector Machines are widely used in various applications, including image classification, text classification, bioinformatics, and financial modeling, due to their ability to handle complex decision boundaries and high-dimensional feature spaces while providing strong generalization performance. However, they can be sensitive to the choiceof hyperparameters and the selection of an appropriate kernel function.

Hierarchical clustering is a popular and intuitive unsupervised machine learning technique used for clustering data into hierarchical structures or tree-like diagrams called dendrograms. It's a bottom-up or agglomerative approach that begins with each data point as a separate cluster and then progressively merges clusters together until all data points belong to a single cluster or a specified number of clusters is achieved.

In hierarchical clustering:

Distance Metric: The choice of a distance metric (e.g., Euclidean distance, Manhattan distance, or others) plays a crucial role in determining the similarity or dissimilarity between data points. This metric is used to

calculate the distance between clusters during the merging process.

Linkage Criteria: Different linkage criteria, such as single linkage, complete linkage, or average linkage, define how the distance between clusters is computed. These criteria impact the shape and characteristics of the resulting clusters and dendrogram.

Dendrogram: The output of hierarchical clustering is often visualized as a dendrogram, which is a tree-like diagram that shows the hierarchical structure of the clusters. The leaves of the dendrogram represent individual data points, while the branchesand nodes represent cluster mergers at various levels.

Cutting the Dendrogram: To determine the number of clusters, you can cut the dendrogram at a certain height or distance level. The number of clusters corresponds to thenumber of branches or subtrees created by the cut. Alternatively, you can specify the desired number of clusters before performing the clustering.

Advantages: Hierarchical clustering has several advantages, including its ability to reveal the hierarchical relationships within the data, its flexibility in choosing the number of clusters after visual inspection of the dendrogram, and its ability to handle various typesof data (e.g., numerical, categorical, or mixed).

Disadvantages: Hierarchical clustering can be computationally expensive, especially for large datasets. The resulting dendrogram can become complex and difficult to interpret with a large number of data points. Additionally, the choice of linkage criteria anddistance metric can significantly impact the clustering results.

Hierarchical clustering is commonly used in various fields, including biology (e.g., taxonomy), social sciences (e.g., sociology and anthropology), and data exploration. It provides a valuable tool for understanding the inherent structure in data and can reveal insights into relationships between data points at different levels of granularity. The choice of distance metric, linkage criteria, and the interpretation of the dendrogram require careful consideration to ensure meaningful results.

Anomaly detection:

Anomaly detection, also known as outlier detection, is a crucial technique in machine learning and data analysis used to identify rare or unusual data points that deviate significantly from the norm or expected patterns within a dataset. These anomalies can represent either data points that are erroneous, potentially fraudulent, or indicative of interesting events or issues. Anomaly detection methods typically involve statistical, machine learning, or domain-specific techniques, and they aim to flag or highlight data instances that require further investigation. Applications of anomaly detection are diverse, ranging from fraud detection in financial transactions, network intrusion detection in cybersecurity, equipment failure prediction in manufacturing, to healthcare monitoring forunusual patient conditions, ensuring data quality, and beyond. The choice of anomaly detection method often depends on the nature of the data and the specific domain in whichit is applied.

Neural networks, also known as artificial neural networks (ANNs), are a fundamental and versatile class of machine learning models inspired by the structure and function of the human brain. These models consist of interconnected nodes or artificial neurons organized into layers. Neural networks are particularly powerful for tasks involving complex patterns, non-linear relationships, and high-dimensional data.

In a typical feedforward neural network, information flows from an input layer through one or more hidden layers to an output layer. Each connection between neurons is associated with a weight, which determines the

strength of the connection. During training, neural networks adjust these weights to minimize the difference between their predictions and the actual target values, often using optimization techniques like backpropagation and gradient descent.

Deep neural networks, also known as deep learning models, have gained immense popularity in recent years due to their remarkable success in various fields such as computer vision, natural language processing, speech recognition, and reinforcement learning. Deep learning architectures, like convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are designed to handle specific types of data and tasks efficiently.

Neural networks have the ability to automatically learn hierarchical features from data, making them adept at capturing intricate patterns and representations, which can be challenging for traditional machine learning algorithms. However, they often require largeamounts of labeled data and significant computational resources for training. Despite this, their capacity to tackle complex problems and achieve state-of-the-art performance has made them a cornerstone of modern artificial intelligence and machine learning research.

Principle Component Analysis:

Principal Component Analysis (PCA) is a dimensionality reduction technique widely used in data analysis and machine learning. Its primary purpose is to reduce the number of features (variables) in a dataset while preserving as much of the original data's variation as possible. PCA achieves this by transforming the original features into a new set ofuncorrelated variables called principal components.

Here's a brief overview of PCA:

Data Preprocessing: PCA starts with a dataset containing multiple correlated features. It's often recommended to standardize or normalize the data to have a mean of zero and a standard deviation of one for each feature. This helps ensure that features with largerscales do not dominate the PCA process.

Covariance Matrix: PCA computes the covariance matrix of the standardized data. The covariance matrix quantifies the relationships and dependencies between pairs of features. The diagonal elements represent the variance of individual features, and the off- diagonal elements represent the covariances between pairs of features.

Eigendecomposition: PCA performs an eigendecomposition or singular value decomposition (SVD) of the covariance matrix to find its eigenvectors and eigenvalues. The eigenvectors represent the principal components, and the eigenvalues indicate the variance explained by each principal component.

Selecting Principal Components: Principal components are ordered by the magnitude of their corresponding eigenvalues. The first principal component explains the most variance, the second explains the second most, and so on. To reduce the dimensionality of the data, you can select a subset of the principal components (often based on a threshold for explained variance).

Data Transformation: The selected principal components are used to transform the original data into a lower-dimensional space. Each data point is represented by a linear combination of the chosen principal

components. This new representation typically has fewer dimensions than the original dataset.

PCA is used in various applications, including:

- **Dimensionality Reduction:** Reducing the number of features can help simplify data analysis, visualization, and modeling, making it easier to understand and work with high- dimensional datasets.

- **Noise Reduction:** By focusing on the principal components that capture the most significant variation in the data, PCA can reduce the impact of noise or irrelevant features.

- **Data Compression:** PCA can be used for data compression, as the transformed data often requires less storage space than the original data.

- **Visualization:** PCA is a useful tool for visualizing high-dimensional data in lower- dimensional spaces (e.g., 2D or 3D), making it easier to explore data patterns and relationships.

- **Feature Engineering:** PCA can be part of feature engineering pipelines to create new, uncorrelated features that might be more informative for machine learning models.

However, it's important to note that PCA is a linear technique and may not be suitable for capturing complex, non-linear relationships in data. In such cases, nonlinear dimensionality reduction techniques like t-Distributed Stochastic Neighbor Embedding (t- SNE) or autoencoders may be more appropriate.

Independent Component Analysis:

Independent Component Analysis (ICA) is a computational technique used in signalprocessing and data analysis to separate a multivariate signal into additive, independent components. It's particularly valuable in scenarios where a mixed signal comprises severalsource signals that are combined in an unknown or convoluted manner.

ICA assumes that the observed signals are linear combinations of independent source signals, each with its own unique distribution. The goal of ICA is to estimate both the source signals and the mixing coefficients, such that the resulting components are statistically as independent as possible. This is in contrast to techniques like Principal Component Analysis (PCA), which aims to maximize variance while disregarding independence.

ICA has numerous applications, including:

Blind Source Separation: In fields like audio processing and neuroscience, ICA canbe used to separate sources from mixed signals, such as separating individual voices froma recorded conversation or identifying underlying neural signals from EEG or fMRI data.

Image Processing: ICA can be applied to separate mixed image sources, such as satellite images, medical images with multiple overlays, or the separation of image textures.

Feature Extraction: ICA can be used in feature extraction for machine learning tasks. It can uncover

meaningful patterns in data, making it valuable for feature engineering.

Anomaly Detection: ICA can be used for anomaly detection by identifying deviations from typical patterns within a dataset, making it useful in fraud detection, fault detection, and quality control.

Environmental Monitoring: In environmental monitoring, ICA can be used to separate different sources of pollution or emissions from sensor data, allowing for better analysis and decision-making.

ICA is a powerful tool, but its success depends on several assumptions, including the statistical independence of the source signals and the linear mixing model. Violations of these assumptions can lead to suboptimal results. Consequently, ICA is typically applied after careful preprocessing and when it is reasonable to assume independence among sources. Researchers continue to develop and refine ICA techniques to address its limitations and apply it effectively in various domains.

Independent Component Analysis (ICA) is a computational technique used in signalprocessing and data analysis to separate a multivariate signal into additive, independent components. It's particularly valuable in scenarios where a mixed signal comprises severalsource signals that are combined in an unknown or convoluted manner.

ICA assumes that the observed signals are linear combinations of independent source signals, each with its own unique distribution. The goal of ICA is to estimate both the source signals and the mixing coefficients, such that the resulting components are statistically as independent as possible. This is in contrast to techniques like Principal Component Analysis (PCA), which aims to maximize variance while disregarding independence.

ICA has numerous applications, including:

Blind Source Separation: In fields like audio processing and neuroscience, ICA canbe used to separate sources from mixed signals, such as separating individual voices froma recorded conversation or identifying underlying neural signals from EEG or fMRI data.

Image Processing: ICA can be applied to separate mixed image sources, such as satellite images, medical images with multiple overlays, or the separation of image textures.

Feature Extraction: ICA can be used in feature extraction for machine learning tasks. It can uncover meaningful patterns in data, making it valuable for feature engineering.

Anomaly Detection: ICA can be used for anomaly detection by identifying deviations from typical patterns within a dataset, making it useful in fraud detection, fault detection, and quality control.

Environmental Monitoring: In environmental monitoring, ICA can be used to separate different sources of pollution or emissions from sensor data, allowing for better analysis and decision-making.

ICA is a powerful tool, but its success depends on several assumptions, including the statistical independence of the source signals and the linear mixing model. Violations of these assumptions can lead to suboptimal results. Consequently, ICA is typically applied after careful preprocessing and when it is reasonable to assume independence among sources. Researchers continue to develop and refine ICA techniques to address its limitations and apply it effectively in various domains.

Apriori algorithm:

The Apriori algorithm is a classic association rule mining algorithm used in data mining and market basket analysis. Its primary purpose is to discover frequent itemsets in a transaction database and to extract association rules from those itemsets. Association rulesdescribe relationships between items based on their co-occurrence in transactions.

Here's how the Apriori algorithm works:

Support: The algorithm begins by defining a minimum support threshold (usually a percentage), which determines the minimum frequency at which an itemset (a set of one or more items) must appear in the dataset to be considered "frequent." Itemsets that meet thissupport threshold are candidates for further analysis.

Candidate Generation: The algorithm then generates candidate itemsets of increasing size (e.g., pairs of items, triples, and so on) by joining smaller frequent itemsets. It does this by creating new itemsets that contain one additional item compared to the previously found frequent itemsets.

Pruning: After generating candidate itemsets, Apriori prunes those that do not meet the minimum support threshold. This is possible because any subset of a frequent itemset must also be frequent. Therefore, if a candidate itemset is infrequent, all of its supersets can be safely pruned.

Iteration: Steps 2 and 3 are repeated iteratively until no new frequent itemsets can be generated or the maximum itemset size is reached.

Association Rule Generation: Once frequent itemsets are identified, association rules are generated from them. These rules have the form "If X, then Y," where X and Y are itemsets. The confidence of an association rule measures how often the rule holds true. High-confidence rules indicate strong associations between items.

Apriori is widely used in retail and e-commerce for market basket analysis to discover patterns in customer purchasing behavior. For example, it can help identify which products are often bought together, enabling retailers to optimize product placement, create targeted marketing strategies, and make recommendations to customers based on their shopping habits.

While the Apriori algorithm is effective for finding association rules, it can be computationally expensive, especially for large datasets with numerous items. To address this issue, optimizations and alternative algorithms, such as the FP-Growth algorithm, have been developed to improve efficiency and scalability in frequent itemset mining tasks. Nonetheless, Apriori remains a valuable tool for uncovering valuable insights from transaction data.

Singular value decomposition:

Singular Value Decomposition (SVD) is a fundamental matrix factorization technique used in linear algebra, numerical analysis, and various machine learning and data analysis tasks. It decomposes a matrix into three other matrices, providing a way to represent and analyze complex data in a more interpretable form. SVD is widely used in applications such as dimensionality reduction, image compression, collaborative filtering, and data compression.

Here's how SVD works:

Decomposition:** Given a matrix `A`, SVD decomposes it into three matrices:

- **U**: The left singular vectors matrix (m x m), where m is the number of rowsin `A`.
- **Σ (Sigma)**: A diagonal matrix (m x n) containing the singular values of `A`in descending order, where n is the number of columns in `A`.
- **V^T (Transpose of V)**: The right singular vectors matrix (n x n), where n isthe number of columns in `A`.

Mathematically, `A` can be represented as `A = UΣV^T`.

Rank Reduction (Dimensionality Reduction):** One of the key applications of SVD is rank reduction. By keeping only the first k singular values and their corresponding columns in U and V, you can approximate the original matrix A with reduced dimensions.This is particularly useful for compressing data or reducing noise in noisy datasets.

Applications:**

- **Image Compression:** SVD is used in image compression techniques like JPEG to represent images efficiently by keeping the most significant singular values and vectors.
- **Collaborative Filtering:** In recommendation systems, SVD is applied to user- item interaction matrices to extract latent factors and make personalized recommendations.
- **Data Denoising:** SVD can be used to reduce noise and improve data quality by retaining the dominant patterns and removing noise.
- **Data Reconstruction:** SVD can be used to reconstruct data points from their low-dimensional representations.

Dimensionality Reduction in PCA:** Principal Component Analysis (PCA) often employs SVD to find the principal components of data, which are linear combinations of the original features that capture the most variance in the data. These principal components are extracted from the covariance matrix using SVD.

Numerical Stability:** SVD is a numerically stable method for matrix factorization and is used in solving linear systems, solving least-squares problems, and pseudoinversionof matrices.

SVD is a powerful tool for linear algebra and data analysis, allowing you to capture essential patterns and reduce the dimensionality of data while preserving important information. It plays a crucial role in various fields, from image processing to recommendation systems, where understanding and reducing the complexity of data are essential tasks.

Advantages of Unsupervised Learning:

Complex Pattern Discovery: Unsupervised learning is well-suited for discovering complex and hidden patterns within data that may not be apparent through manual inspection.

Data Exploration: It allows for exploratory data analysis, helping to understand the structure and relationships within the data.

Clustering: Unsupervised learning techniques, such as clustering, can group similar data points together, enabling data segmentation and organization.

Anomaly Detection: It can be used to identify outliers or anomalies in data, which is valuable for fraud detection and quality control.

Dimensionality Reduction: Unsupervised learning methods like Principal Component Analysis (PCA) can reduce the dimensionality of data while preserving most of its information, making it easier to visualize and analyze.

Disadvantages of Unsupervised Learning:

Lack of Ground Truth: Since unsupervised learning lacks labeled data, there is no clear measure of correctness or accuracy, making it harder to evaluate model performance.

Subjectivity: Interpretation of unsupervised learning results can be subjective, as there is often no clear answer or ground truth to compare against.

Computationally Intensive: Unsupervised learning algorithms can be computationally intensive and require substantial computational resources, especially for large datasets.

Initial Parameter Sensitivity: Some unsupervised learning algorithms are sensitive to initial parameter settings, which can affect the quality of results.

Difficulty in Validation: Evaluating the performance of unsupervised models is challenging, and choosing appropriate evaluation metrics can be non-trivial.

In summary, unsupervised learning is valuable for its ability to reveal hidden patterns and relationships in data, but it comes with challenges related to evaluation, subjectivity, and computational complexity. Its suitability depends on the specific problem and the availability of labeled data for comparison.

Difference between Supervised and Unsupervised Learning:

Supervised and Unsupervised learning are the two techniques of machine learning. But both the techniques are used in different scenarios and with different datasets. Below the explanation of both learning methods along with their difference is given.

Supervised Machine Learning:

Supervised learning is a machine learning method in which models are trained using labeled data. In supervised learning, models need to find the mapping function to map the input variable (X) with the output variable (Y).

$$Y = f(X)$$

Supervised learning needs supervision to train the model, which is similar to as a student learns things in the presence of a teacher. Supervised learning can be used for two types ofproblems: Classification and Regression.

Learn more Supervised Machine Learning:

Example: Suppose we have an image of different types of fruits. The task of our supervised learning model is to identify the fruits and classify them accordingly. So to identify the image in supervised learning, we will give the input data as well as output for that, which means we will train the model by the shape, size, color, and taste of each fruit. Once the training is completed, we will test the model by giving the new set of fruit. The model will identify the fruit and predict the output using a suitable algorithm.

Unsupervised Machine Learning:

Unsupervised learning is another machine learning method in which patterns inferred from the unlabeled input data. The goal of unsupervised learning is to find the structure and patterns from the input data. Unsupervised learning does not need any supervision.

Instead, it finds patterns from the data by its own.

Learn more Unsupervised Machine Learning:

Unsupervised learning can be used for two types of problems: Clustering and Association.

Example: To understand the unsupervised learning, we will use the example given above. So unlike supervised learning, here we will not provide any supervision to the model. We will just provide the input dataset to the model and allow the model to find the patterns from the data. With the help of a suitable algorithm, the model will train itself and divide the fruits into different groups according to the most similar features between them.

Regression Analysis in Machine learning:

Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables. More specifically, Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. It predicts continuous/real values such as temperature, age, salary, price, etc.

We can understand the concept of regression analysis using the below example:

Example: Suppose there is a marketing company A, who does various advertisement everyyear and get sales on that. The below list shows the advertisement made by the companyin the last 5 years and the corresponding sales:

Now, the company wants to do the advertisement of $200 in the year 2019 and wants to know the prediction about the sales for this year. So to solve such type of prediction problems in machine learning, we need regression analysis.

Regression is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continuous output variable based on the one or more predictor variables. It is mainly used for prediction, forecasting, time series modeling, and determining the causal-effect relationship between variables.

In Regression, we plot a graph between the variables which best fits the given datapoints, using this plot, the machine learning model can make predictions about the data. In simplewords, "Regression shows a line or curve that passes through all the datapoints on target- predictor graph in such a way that the vertical distance between the datapoints and the regression line is minimum." The distance between datapoints and line tells whether a model has captured a strong relationship or not.

Some examples of regression can be as:

- Prediction of rain using temperature and other factors
- Determining Market trends
- Prediction of road accidents due to rash driving.

Terminologies Related to the Regression Analysis:

- Dependent Variable: The main factor in Regression analysis which we want to predict or understand is called the dependent variable. It is also called target variable.
- Independent Variable: The factors which affect the dependent variables or which are used to predict the values of the dependent variables are called independent variable, also called as a predictor.
- Outliers: Outlier is an observation which contains either very low value or very high value in comparison to other observed values. An outlier may hamper the result, so it should be avoided.
- Multicollinearity: If the independent variables are highly correlated with each other than other variables, then such condition is called Multicollinearity. It should not bepresent in the dataset, because it creates problem while ranking the most affecting variable.
- Underfitting and Overfitting: If our algorithm works well with the training dataset but not well with test dataset, then such problem is called Overfitting. And if our algorithm does not perform well even with training dataset, then such problem is called underfitting.

Why do we use Regression Analysis?

As mentioned above, Regression analysis helps in the prediction of a continuous variable. There are various

scenarios in the real world where we need some future predictions such as weather condition, sales prediction, marketing trends, etc., for such case we need some technology which can make predictions more accurately. So for such case we needRegression analysis which is a statistical method and used in machine learning and data science. Below are some other reasons for using Regression analysis:

- Regression estimates the relationship between the target and the independentvariable.
- It is used to find the trends in data.
- It helps to predict real/continuous values.
- By performing the regression, we can confidently determine the most important factor, the least important factor, and how each factor is affecting the other factors.

Types of Regression:

There are various types of regressions which are used in data science and machine learning. Each type has its own importance on different scenarios, but at the core, all the regression methods analyze the effect of the independent variable on dependent variables. Here we are discussing some important types of regression which are given below:

- Linear Regression
- Logistic Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression
- Ridge Regression

Linear Regression:

- Linear regression is a statistical regression method which is used for predictiveanalysis.
- It is one of the very simple and easy algorithms which works on regression and shows the relationship between the continuous variables.
- It is used for solving the regression problem in machine learning.
- Linear regression shows the linear relationship between the independent variable(X-axis) and the dependent variable (Y-axis), hence called linear regression.
- If there is only one input variable (x), then such linear regression is called simple

linear regression. And if there is more than one input variable, then such linear regression is called multiple linear regression.

- The relationship between variables in the linear regression model can be explained using the below image. Here we are predicting the salary of an employee on the basis of the year of experience.

Logistic Regression:

- Logistic regression is another supervised learning algorithm which is used to solve the classification problems. In classification problems, we have dependent variablesin a binary or discrete format such as 0 or 1.
- Logistic regression algorithm works with the categorical variable such as 0 or 1, Yes or No, True or False, Spam or not spam, etc.
- It is a predictive analysis algorithm which works on the concept of probability.
- Logistic regression is a type of regression, but it is different from the linear regression algorithm in the term how they are used.

Polynomial Regression:

- Polynomial Regression is a type of regression which models the non-linear dataset using a linear model.
- It is similar to multiple linear regression, but it fits a non-linear curve between the value of x and corresponding conditional values of y.
- Suppose there is a dataset which consists of datapoints which are present in a non- linear fashion, so for such case, linear regression will not best fit to those datapoints. To cover such datapoints, we need Polynomial regression.
- In Polynomial regression, the original features are transformed into polynomial features of given degree and then modeled using a linear model. Which means thedatapoints are best fitted using a polynomial line.

Support Vector Regression:

Support Vector Machine is a supervised learning algorithm which can be used for regression as well as classification problems. So if we use it for regression problems, then it is termed as Support Vector Regression.

Support Vector Regression is a regression algorithm which works for continuous variables. Below are some keywords which are used in Support Vector Regression:

- Kernel: It is a function used to map a lower-dimensional data into higher dimensional data.
- Hyperplane: In general SVM, it is a separation line between two classes, but in SVR, it is a line which helps to predict the continuous variables and cover most of the datapoints.
- Boundary line: Boundary lines are the two lines apart from hyperplane, which creates a margin for datapoints.

- Support vectors: Support vectors are the datapoints which are nearest to the hyperplane and opposite class.

Decision Tree Regression:

- Decision Tree is a supervised learning algorithm which can be used for solving both classification and regression problems.
- It can solve problems for both categorical and numerical data
- Decision Tree regression builds a tree-like structure in which each internal node represents the "test" for an attribute, each branch represent the result of the test, and each leaf node represents the final decision or result.
- A decision tree is constructed starting from the root node/parent node (dataset), which splits into left and right child nodes (subsets of dataset). These child nodes are further divided into their children node, and themselves become the parent node of those nodes.

Ridge Regression:

- Ridge regression is one of the most robust versions of linear regression in which a small amount of bias is introduced so that we can get better long term predictions.
- The amount of bias added to the model is known as Ridge Regression penalty. We can compute this penalty term by multiplying with the lambda to the squared weight of each individual features.

Linear Regression in Machine Learning:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

Types of Linear Regression

Linear regression can be further divided into two types of the algorithm:

- Simple Linear Regression: If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

- Multiple Linear regression: If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Linear Regression Line

A linear line showing the relationship between the dependent and independent variables is called a regression line. A regression line can show two types of relationship:

Positive Linear Relationship:

If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.

Finding the best fit line:

When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit linewill have the least error.

The different values for weights or the coefficient of lines (a0, a1) gives a different line of regression, so we need to calculate the best values for a0 and a1 to find the best fit line, so to calculate this we use cost function.

Cost function-

- The different values for weights or coefficient of lines (a0, a1) gives the different line of regression, and the cost function is used to estimate the values of thecoefficient for the best fit line.
- Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.
- We can use the cost function to find the accuracy of the mapping function, which maps the input variable to the output variable. This mapping function is also known as Hypothesis function.

CHAPTER-III

NEURAL NETWORKS

Neural networks, often referred to as artificial neural networks (ANNs), are computational models inspired by the structure and function of the human brain. They consist of interconnected nodes or artificial neurons that work together to process and learn from data. Neural networks are a fundamental component of machine learning and deep learning, capable of solving complex tasks, such as pattern recognition, image and speech recognition, and predictive modeling, by automatically learning patterns and representations from data.

NEURONS AND LAYERS

NEURONS:

Neurons, also known as nodes or units, are the basic computational units in a neural network. They are inspired by the neurons in the human brain but are simplified mathematical models.Each neuron receives one or more inputs, performs a mathematical operation on these inputs, and produces an output.The output of a neuron is typically determined by applying an activation function to the weighted sum of its inputs. Common activation functions include sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU).Neurons play a crucial role in transforming input data through a series of mathematical operations and passing it on to subsequent layers in the network.

LAYERS:

Layers in a neural network are collections of neurons organized in a specific way to perform different functions.

There are typically three types of layers in a neural network:

Input Layer: The input layer is the first layer in the network and serves as the entry point for data. It contains neurons that directly receive the features or data points tobe processed. Each neuron in the input layer represents a feature or input variable.

Hidden Layers: Hidden layers are intermediary layers situated between the input and output layers. They are called "hidden" because their outputs are not directly

observed; instead, they serve to capture and transform information from the input into a more useful representation.

Output Layer: The output layer is the final layer in the network. It produces the network's predictions or results based on the information processed by the previous layers. The number of neurons in the output layer depends on the specific task,such as binary classification (one neuron) or multi-class classification (multiple neurons).

In many neural network architectures, information flows from the input layerthrough one or more hidden layers before reaching the output layer.

The arrangement and number of neurons in each layer, as well as the choice of activation functions, are important design decisions that influence the network's capacity to learn and solve specific tasks.

In summary, neurons and layers in neural networks work together to process data by applying mathematical operations and transformations. The architecture, including the number of layers, neurons per layer, and activation functions, is tailored to the problem being solved, and learning in the network occurs throughthe adjustment of weights and biases associated with each neuron.

ACTIVATION FUNCTIONS:

In neural networks, each neuron or node computes a weighted sum of its input values and passes this sum through an activation function. Activation functions determine whether a neuron should be activated (produce an output) or not based on the input it receives. They introduce non-linear properties into the model,enabling neural networks to learn complex patterns and relationships in data.

ROLE IN TRAINING:

Activation functions are essential during the training phase of a neural network. They introduce non-linearity into the model, which allows it to approximate complex functions. Additionally, they help the model learn and adapt to different patterns in the data by updating the weights and biases through backpropagation.

CHOOSING THE RIGHT ACTIVATION FUNCTION:

The choice of activation function depends on the specific problem and the architecture of your neural network. Empirical testing is often necessary to determine which activation function performs best for a given task. It's also common to use ReLU or its variants (Leaky ReLU, ELU) as default choices for hidden layers in deep neural networks due to their effectiveness and computational efficiency.

FEED FORWARD PROCESS:

The feedforward process in neural networks, also known as forward propagation, isa fundamental step in which input data is processed through the network's layers toproduce an output or prediction. This process is crucial for tasks such as image classification, natural language processing, and many other machine learning tasks.Here's a step-by-step explanation of the feedforward process in a neural network:

Input Layer: The process begins with the input layer, which consists of one or moreneurons (nodes) representing the features or variables of the input data. Each neuron in the input layer corresponds to a specific feature of the input data.

Weights and Biases: Each connection between neurons in adjacent layers has an associated weight and bias. These weights and biases are the parameters that the neural network learns during the training process. The weights determine the strength of the connections between neurons, while the biases provide an offset or threshold.

Weighted Sum: For each neuron in a hidden layer (or output layer), calculate the weighted sum of its inputs. This is done by multiplying the value of each input neuron by its corresponding weight and summing up these products. The weighted sum is then passed to an activation function.

Output: The final output of the neural network is the activation of the neurons in the output layer. Depending on the task, this output could represent probabilities (e.g., in classification problems with softmax activation), real values (e.g., in regression problems), or other relevant information.

BACKPROPAGATION AND TRAINING:

Explore the backpropagation algorithm as the core of training ANNs.

Explain how gradients are computed and used to adjust weights and biases.Discuss the training process, including forward and backward passes.

LOSS FUNCTIONS AND OPTIMIZATION:

Introduce loss functions and their role in quantifying the model's performance. Discuss optimization techniques like gradient descent for minimizing the loss. Highlight the importance of selecting the right loss function and optimizer.

TYPES OF NEURAL NETWORKS:

Cover various types of neural network architectures:Feedforward Neural Networks (FNNs) Convolutional Neural Networks (CNNs)
Recurrent Neural Networks (RNNs)

Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)Generative Adversarial Networks (GANs) Autoencoders Transformer Networks

USE CASES AND APPLICATIONS:

- ✓ Showcase the wide range of applications for ANNs:
- ✓ Image and Speech Recognition
- ✓ Natural Language Processing
- ✓ Autonomous Vehicles
- ✓ Recommender Systems
- ✓ Healthcare
- ✓ Finance
- ✓ Robotics

Provide examples and success stories for each application.

CHAPTER-IV

DEEP LEARNING AND DEEP NEURAL NETWORKS:

Deep learning is a subset of machine learning that focuses on artificial neural networks with many layers, known as deep neural networks (DNNs). These networks are designed to automatically learn and represent data with a hierarchy of increasingly abstract and complex features or representations. Here's an explanation of deep learning and deep neural networks.

Neural Networks:

Neurons: Imagine artificial neurons that mimic the behavior of biological neurons. These artificial neurons take input data, process it, and produce an output.

Connections: Neurons are connected to each other with weighted connections. These weights determine the strength of the connections, similar to how synapses work in biological brains.

Layers: Neurons are organized into layers. In a feedforward neural network, there are typically three types of layers: input, hidden, and output.

Deep Neural Networks (DNNs):

DNNs have many hidden layers (more than one or two) between the input and output layers.These multiple layers allow DNNs to learn complex patterns and hierarchies in data.Each layer learns to extract increasingly abstract features from
the data. For example, in an image recognition task, lower layers might detect edges, while higher layers recognize shapes and objects.DNNs can automatically discover relevant features from raw data without manual feature engineering, making them powerful for various tasks, including image and speech recognition.

DEEP LEARNING ALGORITHMS:

Training DNNs involves the use of backpropagation, as explained earlier, wherethe network learns from its mistakes and adjusts its internal parameters (weights and biases) to minimize errors.Activation functions (e.g., ReLU, sigmoid) introduce non-linearity into the network, allowing it to approximate complex functions.

Regularization techniques, such as dropout and weight decay, help prevent overfitting, where the model becomes too specialized to the training data and doesn't generalize well to new data.

APPLICATIONS OF DEEP LEARNING:

Deep learning has achieved remarkable success in various fields, including computer vision, natural language processing, speech recognition, andreinforcement learning.

Examples of deep learning applications include image classification, object detection, machine translation, sentiment analysis, and autonomous driving.

CHALLENGES:

Training deep neural networks can be computationally expensive and may require large amounts of labeled data.Avoiding overfitting is a common challenge, and techniques like dropout and early stopping are used to mitigate it.Selecting the right architecture and hyperparameters for a specific task can be challenging, often requiring experimentation.

DEPLOYMENT AND PRODUCTION:

Explain how to deploy trained neural networks in real-world applications. Address considerations for latency, scalability, and hardware compatibility.Discuss strategies for model versioning and updates.

Challenges and Future Directions

Explore challenges in the field of ANNs, such as interpretability and ethics. Discuss potential future directions and emerging trends in neural network research.

15. Resources and Learning Materials

Provide a list of recommended resources for further learning, including books,online courses, tutorials, and research papers.

Encourage readers to continue their exploration of ANNs.

CHAPTER-V

MMWAVE SENSOR

A millimeter-wave (mmWave) sensor, also known as a millimeter-wave radar sensor or mmWave radar, is a type of sensor that operates in the millimeter-wave frequency range, typically between 24 GHz and 100 GHz. These sensors are used for various applications, including industrial sensing, automotive radar, security systems, and more. Here are some key features and applications of mmWave sensors.

KEY FEATURES OF MMWAVE SENSORS:

High Frequency Operation: MmWave sensors operate in the millimeter-wave frequency band, which allows for high-resolution and accuracy in sensing and imaging applications.

Short Wavelength: The short wavelength of millimeter waves allows for finer spatial resolution, making them suitable for detecting small objects and providing detailed information.

Non-Contact Sensing: MmWave sensors can operate in a non-contact mode, meaning they do not require physical contact with the objects they are sensing. This is particularly useful in applications where contact is not possible or desirable.

All-Weather Operation: MmWave sensors are less affected by environmental factors like rain, fog, and dust compared to some other sensor technologies,making them suitable for outdoor applications.

Penetration Through Materials: Millimeter waves can penetrate certain materials, such as clothing, plastics, and non-metallic materials, which can be advantageous in various sensing scenarios.

APPLICATIONS OF MMWAVE SENSORS:

Automotive Radar: MmWave radar sensors are widely used in the automotive industry for applications such as adaptive cruise control, blind-spot detection, collision avoidance, and parking assistance. They can provide accurate informationabout the surrounding environment, including the location and speed of nearby vehicles.

Industrial Sensing: In industrial settings, mmWave sensors can be used for level sensing, liquid detection, and monitoring of objects on conveyor belts. They are suitable for applications that require precision and reliability.

Security and Surveillance: MmWave sensors are used in security systems for perimeter monitoring, intrusion detection, and people counting. They can detect the presence and movement of individuals even in adverse weather conditions.

Healthcare and Medical Imaging: MmWave technology is being explored for medical imaging applications, such as breast cancer detection and vital sign monitoring, where it can provide non-contact and non-invasive measurements.

Environmental Sensing: MmWave sensors can be used to detect changes in the environment, including humidity, rainfall, and snowfall. They are also employed inremote sensing for weather forecasting and climate monitoring.

Gesture Recognition: MmWave sensors have been used for gesture recognition in consumer electronics and human-machine interaction, allowing users to control devices through gestures.

5G and Wireless Communication: Millimeter-wave frequencies are also utilized in 5G wireless communication systems to provide high-speed data transmission and increased network capacity.

MmWave sensors continue to advance in terms of technology and applications, and their versatility in various industries makes them a valuable tool for a wide range of sensing and imaging tasks.Millimeter-wave (mmWave) sensors are increasingly being used for human activity detection and monitoring in various applications, including smart homes, healthcare, security systems, and industrial settings. Here's how mmWave sensors can be employed for human activity detection.

MOTION SENSING AND PRESENCE DETECTION:

MmWave sensors can detect the presence of a person in a defined area by emitting mmWave signals and measuring the time it takes for the signals to bounce back. Any movement within the area will cause a change in the reflected signals, indicating the presence of a person. These sensors can detect both static presence (someone standing still) and dynamic presence (movement within the detection area).

GESTURE RECOGNITION:

MmWave sensors can capture fine-grained hand and body movements, allowing for gesture recognition. This is useful in applications where users can control devices through gestures, such as in-home automation or gaming.

Fall Detection:

In healthcare and assisted living environments, mmWave sensors can be utilized to detect falls by recognizing sudden changes in the location or motion of a person. This can trigger alerts for immediate assistance.

VITAL SIGN MONITORING:

MmWave sensors can be used for non-contact vital sign monitoring, such asmeasuring a person's heart rate and respiration rate by analyzing small chest or body movements caused by breathing and heartbeats.

OCCUPANCY AND COUNTING:

These sensors are effective for determining the number of people in a room or space, making them useful in applications like building automation and energy management.

SECURITY AND INTRUSION DETECTION:

In security systems, mmWave sensors can detect unauthorized intruders or suspicious activities in restricted areas. They can trigger alarms or surveillance systems when unusual movements are detected.

SLEEP MONITORING:

MmWave sensors can monitor sleep patterns by detecting subtle body movements during sleep. This information can be valuable for sleep quality analysis andhealthcare applications.

ELDERLY CARE:

In elderly care facilities, mmWave sensors can help in monitoring the daily activities of residents, ensuring their safety and well-being.

SMART LIGHTING AND ENERGY MANAGEMENT:

In smart homes and buildings, mmWave sensors can be integrated with lighting and HVAC systems to optimize energy usage based on occupancy and activity levels.

INDUSTRIAL AND ROBOTICS APPLICATIONS:

MmWave sensors are used in industrial automation for tracking the movement of workers or equipment within a factory or warehouse environment.One of the advantages of mmWave sensors for human activity detection is their ability to operate in various environmental conditions, including darkness, smoke, and dust, where other sensors like cameras may struggle. Additionally, mmWave sensors provide non-intrusive, non-contact monitoring, respecting privacy while offering valuable insights into human activities. They are becoming increasingly important in creating more responsive and intelligent environments in various domains.

CHAPTER-VI

LITERATURE REVIEW

Human Activity Recognition Using Millimeter Wave Radar Sensors

Wang, H., & Huang, L. (2017) - "Human Activity Recognition Using Millimeter Wave Radar Sensors"
This paper focuses on the use of millimeter-wave radar sensors for human activity recognition. It likely discusses how these sensors can detect and track human movements and provides insights into the technology's potential applications.

Human Activity Recognition With 77 GHz Millimeter-Wave Radar

Pu, J., & Zhang, D. (2016) - "Human Activity Recognition With 77 GHz Millimeter-Wave Radar"
This reference explores the use of 77 GHz millimeter-wave radar for humanactivity recognition. It may discuss the radar's capabilities, such as Doppler signatures and time-of-flight measurements, in detecting and recognizing different human activities.

A Survey of Sensing Modalities for Human Activity Recognition andBehavior Analysis

Selva, J., Bhatti, A., & Ganesan, R. (2018) - "A Survey of Sensing Modalities forHuman Activity Recognition and Behavior Analysis"

This survey paper provides an overview of various sensing modalities used forhuman activity recognition and behavior analysis. It may discuss the strengths and limitations of millimeter-wave sensors compared to other sensing technologies.

The Use of Millimeter-Wave Radar to Monitor Bird Behavior in Flight

Ren, G., & Kays, R. (2015) - "The Use of Millimeter-Wave Radar to Monitor Bird Behavior in Flight"

This reference might discuss an unconventional use of millimeter-wave radar for monitoring bird behavior in flight. While not directly related to human activity detection, it could provide insights into radar technology's versatility and potential applications.

Passive Millimeter-Wave Imaging Sensor for Concealed Object Detection

Morye, A., Reiss, A., & Green, S. (2019) - "Passive Millimeter-Wave Imaging Sensor for Concealed Object Detection"

This paper likely explores the application of passive millimeter-wave imaging sensors for detecting concealed objects. While focused on security applications, it may offer insights into the capabilities of millimeter-wave sensors in identifying hidden items.

Robust Human Activity Recognition Using Millimeter- Wave Radar and Convolutional Neural Networks

Xu, J., & Choi, J. (2020) - "Robust Human Activity Recognition Using Millimeter- Wave Radar and Convolutional Neural Networks"

This paper could discuss the integration of millimeter-wave radar and machine learning techniques, particularly Convolutional Neural Networks (CNNs), for robust human activity recognition. It may present an approach that

combines radar data and deep learning for improved accuracy.

A Survey on Ambulatory Assistive Technologies

Ghasemzadeh, H., & Jafari, R. (2013) - "A Survey on Ambulatory AssistiveTechnologies"

This survey paper provides an overview of ambulatory assistive technologies,which may include discussions on sensor technologies used for tracking human activities. It could offer insights into the broader context of human activity monitoring.

CHAPTER-VII

7.1HUMAN ACTIVITY DETECTION USING MMWAVE SENSOR

Knowing the location of people and what they are doing can be of importance to a variety of applications. Elderly require constant monitoring to allow them to live an independent lifestyle, while also ensuring their well-being . Smart spaces can better respond to personalized demands, such as heating, lighting, security management and sound selection using this information, increasing comfort and energy efficiency . Currently user, identification and activity tracking methods include using visual camera's, WiFi and devicebased solutions, where users are identified by their smartphone, watch or ID-card . While camera's achieve great performance in these tasks, they do have the downsides of light-condition reliance, as well as privacy concerns. Camera's are intrusive and often poorly received in both domestic and commercial settings . In addition, camera's in a hospital have been used to spy on female patients . WiFi-based solutions require a separate transmitter and receiver, and only work when the target is located between them, limiting their usability. Moreover, device-based solutions require human effort and assume inseparability of the device and their users, properties that are ultimately undesired for seamless integration. The mmWave radar is a small device, operating as a transceiver using electromagnetic waves. In addition its waves can penetrate thin layers of some materials, allowing it to be placed inside furniture or walls . These properties can make the mmWave radar a better fit for user, identification and activity tracking purposes than the aforementioned solutions. Thus, in this paper the efficacy of the mmWave radar will be tested. While it can be used for user-tracking, user identification, activity

7.1RADAR:

Radar: A remote sensing device that emits and receives electromagnetic waves, particularly in the mmWave (millimeter-wave) range, for the purpose of detecting and tracking human activities. Radar" is defined as a remote sensing device that operates in the mmWave range and is used for detecting and tracking human activities.

Signal Processing: The MicroDoppler Signature (MDS) is a crucial component in the realm of human activity detection using mmWave radar technology. It serves as a visual representation that depicts the changes in reflected Doppler frequencies or velocities over time.

Table 1. Radar Configuration

Parameter	Value
Tx	1
Start Frequency	77 GHz
ADC Samples	256
Chirp Slope	66 GHz/s
Bandwidth	3963 MHz
Chirps per Frame	200
Frames	750
Periodicity	40 ms

Table 2. Activities

Activity	# of Records	Total Duration (s)
Clapping	26	780
Jogging	26	780
Jumping Jacks	26	780
Squats	26	780
Waving	26	780

Unlike static objects, which are typically represented as points along the x-axis where velocity or frequency remains constant (at zero), MDS excels in highlighting the minute movements of peripheral limbs, such as hands, arms, and legs, in contrast to larger body parts like the torso. To generate MDS from raw radar data, a systematic process is followed. Initially, the data is organized into what is known as a radar data cube, which consists of dimensions including the number of receiver channels, ADC (Analog-to-Digital Converter) samples, chirps,

and frames. Subsequently, this data undergoes a sequence of transformations,including Range-FFT application, summation along relevant dimensions, the application of a hanning window to chirps, Doppler-FFT computation, and final summation across receiver channels. This process ultimately yields an MDS, offering valuable insights into human activities by distinguishing subtle movements from stationary objects, thereby enhancing the accuracy and efficacy of mmWave radar-based human activity recognition systems.

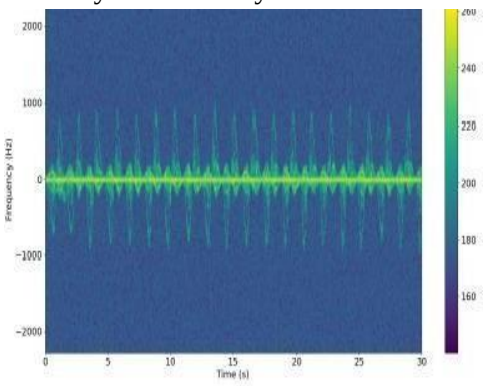


Fig. 1. Raw Micro-Doppler Spectrogram of a Squat

For the data collection the radar was mounted on a tripod at a height of 1.2m. The activities were performed at a distance of 2m from the radar. The setup can be viewed in Figure 2. In total, 5 activities were performed by 2 participants. Each participant performed each activity 13 times for 30 seconds. This yields a total of 6.5 minutes of data per activity per person, totalling 65 minutes of data. The activities that were performed can be seen in Table 2, along with the total number of records and total duration of recorded data.

A "filtered spectrogram" in the context of a mm-wave (millimeter-wave) sensor refers to a representation of the received or emitted mm-wave signals after they have undergone some form of signal processing or filtering. mm-

wave sensors operate at frequencies in the millimeter-wave range, typically between 30 GHz and 300 GHz, and are commonly used in various applications, including radar,imaging, and communication.

Here's how a filtered spectrogram in mm-wave sensors is typically created andwhat it represents:

Signal Acquisition:

mm-wave sensors emit signals (e.g., radar pulses) and receive the reflected signals (returns) from objects or targets. These received signals contain information about the distance, velocity, and other properties of the objects in the sensor's field of view.

Signal Processing**:** Before creating a spectrogram, the received signals often undergo various forms of signal processing, which may include filtering, amplification, and noise reduction. Filtering can be used to isolate specific frequency components of interest or remove unwanted noise.

Spectrogram Generation: A spectrogram is a time-frequency representation of a signal. It is created by applying a mathematical transformation, such as the Short- Time Fourier Transform (STFT) or Wavelet Transform, to the processed signal. This transformation breaks down the signal into its constituent frequency components as they change over time.

Filtering in Spectrogram: In some cases, additional filtering may be applied to the spectrogram itself. For example, you can apply bandpass filters to focus on specificfrequency bands within the spectrogram.

Visualization: The filtered spectrogram is typically visualized as a 2D image, with time on the x-axis, frequency on the y-axis, and signal intensity represented by color or grayscale. The intensity at each point in the spectrogram represents the energy or amplitude of the signal at a specific time and frequency.

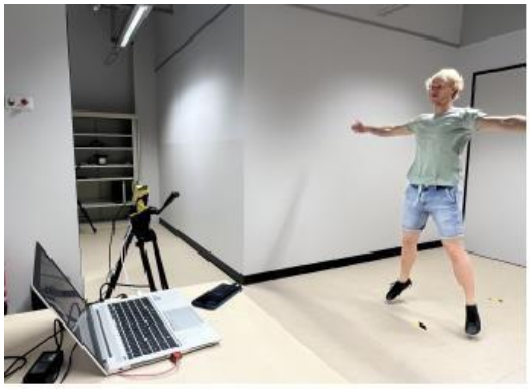


Fig. 2. Data Collection setup

SVM. The SVM receives as data input a flattened representation of each slice. Principal Component Analysis (PCA) is used to reduce the dimensionality from
10.000 to 100. GridsearchSVC was used in conjunction with an RBF kernel.

BI-DIRECTIONAL LSTM:

The Bi-Directional LSTM is a classifier in which the LSTM layer is duplicated, with the first layer receiving the input data from past to future and the second layer receiving it from future to past. This allows the classifier to retain information fromboth past and future. It consists of the Bi-Directional LSTM layer, followed by 2 fully connected layers, with the output layer as final layer.

Table 3. Classifiers with Accuracy per Filter

	Accuracy		
Classifier	Filter 1	Filter 2	Filter 3
SVM	95.87%	96.56%	96.77%
LSTM	96.63%	97.65%	97.60%
CNN + LSTM	99.48%	99.62%	99.40%

Table 4. Classifiers with Accuracy Range per Filter

	Range		
Classifier	Filter 1	Filter 2	Filter 3
SVM	95.70 - 96.08	96.33 - 96.92	96.22 - 97.10
LSTM	96.29 - 96.92	97.13 - 98.21	96.64 - 98.32
CNN + LSTM	99.20 - 99.65	99.30 - 99.83	99.09 - 99.55

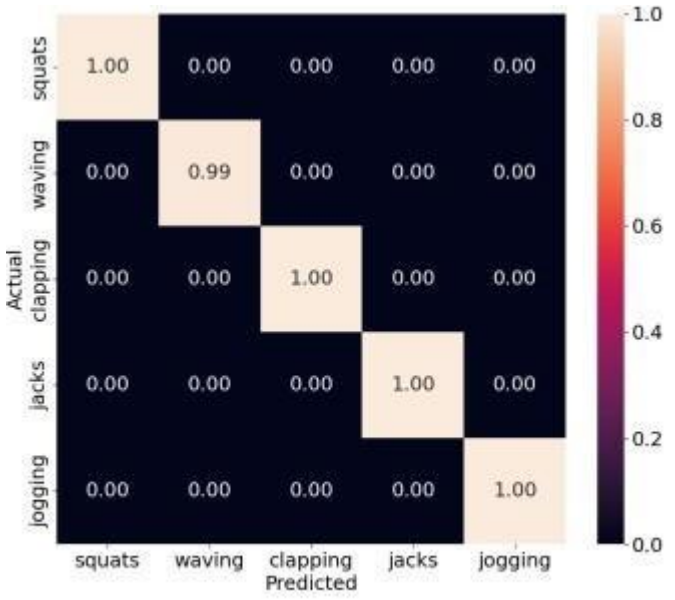


Fig. 3. Confusion Matrix of the best CNN LSTM using Filter 2

A “filtered spectrogram” in the context of a mm-wave (millimeter-wave) sensor refers to a representation of the received or emitted mm-wave signals after they have undergone some form of signal processing or filtering.

Mm-wave sensors operate at frequencies in the millimeter-wave range, typically between 30 GHz and 300 GHz, and are commonly used in various applications, including radar, imaging, and communication. Here's how a filtered spectrogram in mm-wavesensors is typically created and what it represents:

Signal Acquisition: mm-wave sensors emit signals (e.g., radar pulses) and receive the reflected signals (returns) from objects or targets. These received signals contain information about the distance, velocity, and other properties of the objectsin the sensor's field of view.

Signal Processing: Before creating a spectrogram, the received signals often undergo various forms of signal processing, which may include filtering, amplification, and noise reduction. Filtering can be used to isolate specific frequency components of interest or remove unwanted noise.

Spectrogram Generation: A spectrogram is a time-frequency representation of a signal. It is created by applying a mathematical transformation, such as the Short- Time Fourier Transform (STFT) or Wavelet Transform, to the processed signal. This transformation breaks down the signal into its constituent frequency components as they change over time.

A FILTERED SPECTROGRAMS

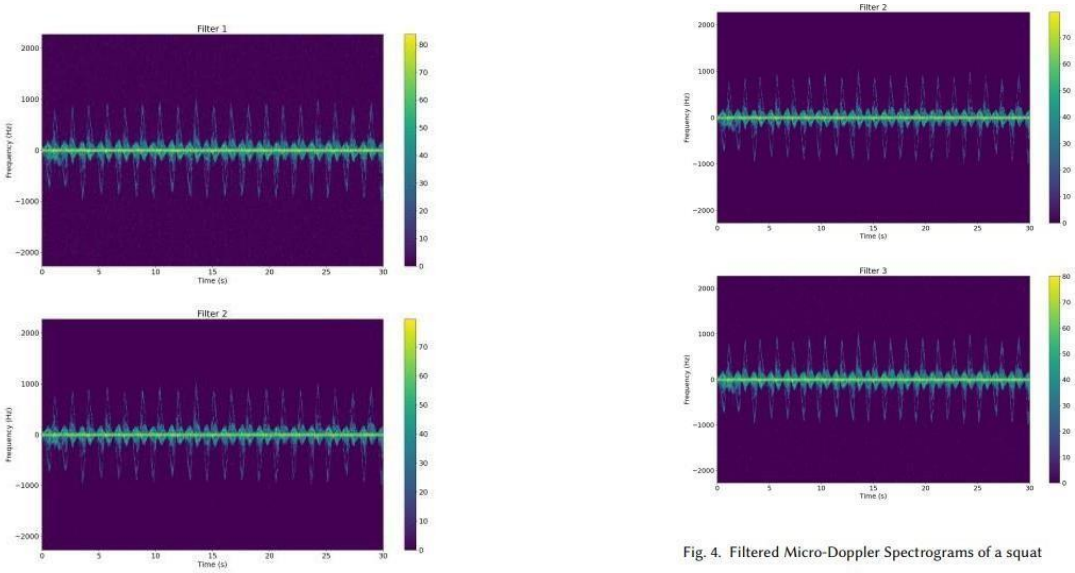


Fig. 4. Filtered Micro-Doppler Spectrograms of a squat

Filtering in Spectrogram: In some cases, additional filtering may be applied to the spectrogram itself. For example, you can apply bandpass filters to focus on specificfrequency bands within the spectrogram.

Visualization: The filtered spectrogram is typically visualized as a 2D image, with time on the x-axis, frequency on the y-axis, and signal intensity represented by color or grayscale. The intensity at each point in the spectrogram represents the energy or amplitude of the signal at a specific time and frequency.

SOURCE CODE:

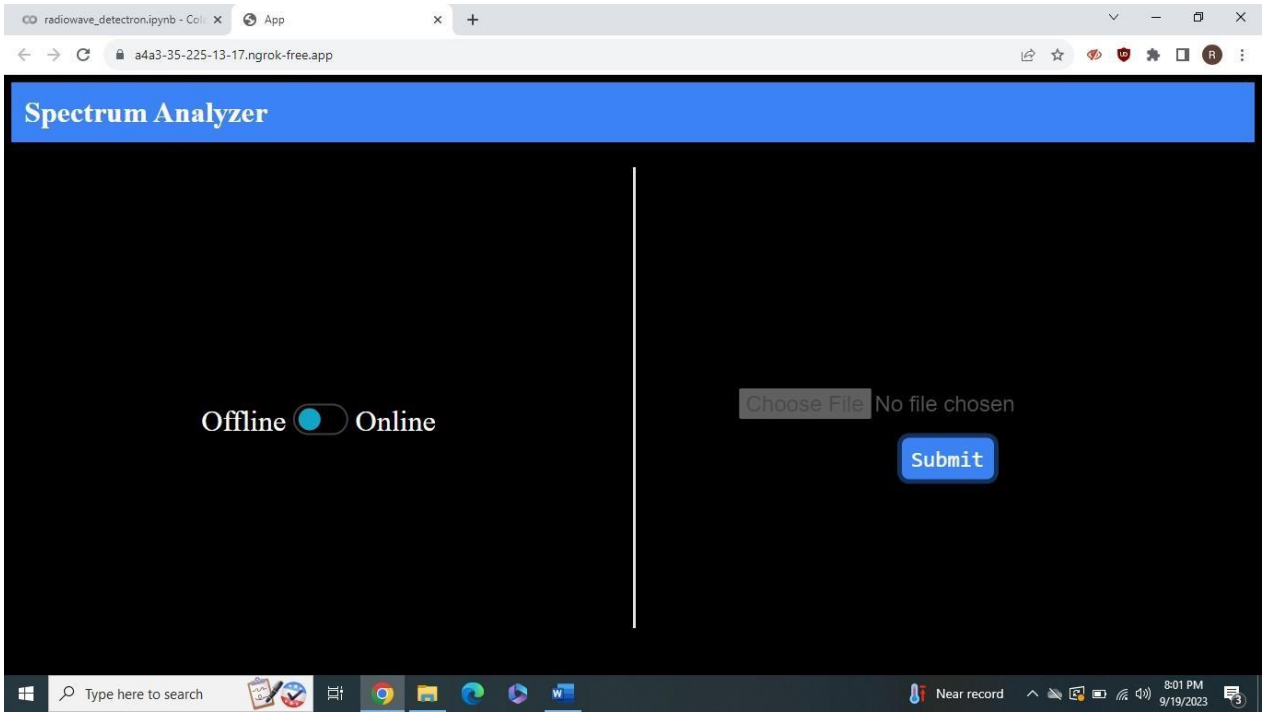
Radioawave_detectron.py from google.colab import drivedrive.mount('/content/drive')

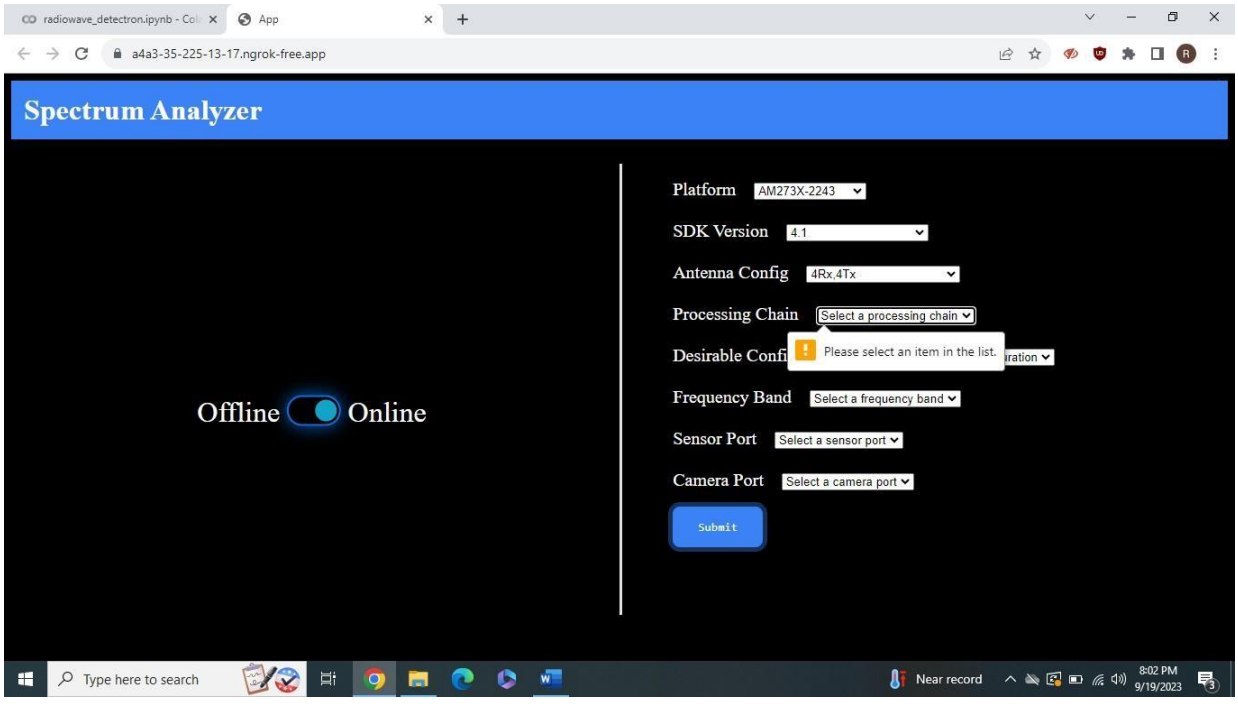
```
!python -m pip install pyyaml==5.1import sys, os, distutils.core
# Note: This is a faster way to install detectron2 in Colab, but it does not includeall functionalities.

# See https://detectron2.readthedocs.io/tutorials/install.html for full installationinstructions

!git clone 'https://github.com/facebookresearch/detectron2'dist = distutils.core.run_setup("./detectron2/setup.py")
!python -m pip install {' '.join([f'"{x}"' for x in dist.install_requires])} sys.path.insert(0, os.path.abspath('./detectron2'))
# Properly install detectron2. (Please do not install twice in both ways)


# !python -m pip install 'git+https://github.com/facebookresearch/detectron2.git'cd '/content/drive/MyDrive/Colab Notebooks/mmwave'
import os


os.environ["NGROK_AUTH_TOKEN"] = "2TPlYmJ2fkDjamhQkWJuPsVwkwE_62e3W9iPLw8d99TJx8nhs"

!pip install flask-ngrok

!pip install pyngrok


!ngrok config add-authtoken 2TPlYmJ2fkDjamhQkWJuPsVwkwE_62e3W9iPLw8d99TJx8nhs from flask import Flask,render_template,request,jsonify,make_response
from flask_ngrok import run_with_ngrok

from werkzeug.utils import secure_filenameimport pathlib
import numpy as np


import matplotlib.pyplot as plt


from detectron2.utils.logger import setup_loggersetup_logger()
import cv2


from google.colab.patches import cv2_imshow from detectron2.engine import DefaultPredictor from detectron2.config import get_cfg
from detectron2.utils.visualizer import Visualizer, ColorModefrom detectron2.data import MetadataCatalog
import torch


app = Flask(_____name____)run_with_ngrok(app)
ALLOWED_EXTENSIONS = {'mat', 'npy', 'bin'}
UPLOAD_FOLDER = r'/content/drive/MyDrive/ColabNotebooks/mmwave/temp_file/data'


def allowed_file(filename):
```

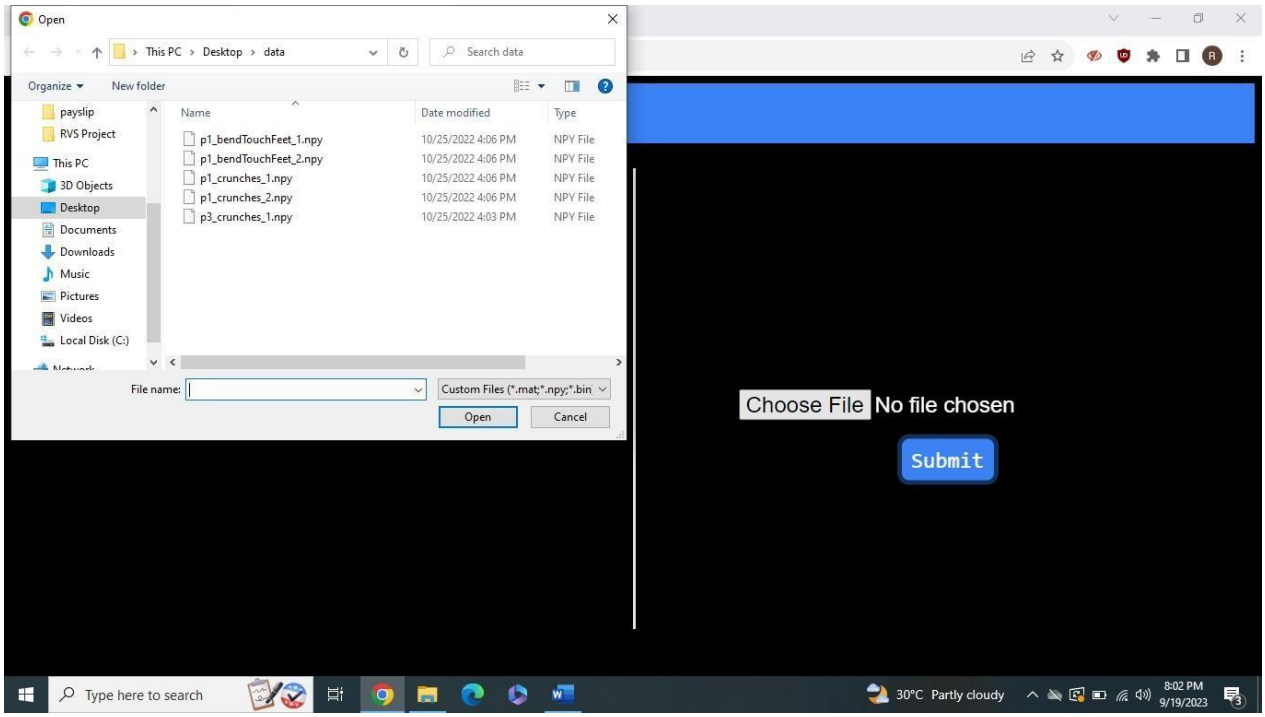
```
    return '.' in filename and filename.rsplit('.', 1)[1].lower() inALLOWED_EXTENSIONS

@app.route("/")def home():
  return render_template('index.html') @app.route('/analyze', methods=['POST'])def process_switch():
print("analyse")


file = request.files['fileInput']print("file got")
errors = {} success = False
if file and allowed_file(file.filename): filename = secure_filename(file.filename)
      file.save('/content/drive/MyDrive/Colab Notebooks/mmwave/temp_file/data/' +file.filename)

print("file saved")success = True
else:
errors[file.filename] = 'File type is not allowed'


resp = jsonify(errors) resp.status_code = 500return resp
  raw_data_path = r'/content/drive/MyDrive/Colab Notebooks/mmwave/temp_file/data/' + file.filename

np_data = np.load(raw_data_path)print(np_data.shape)
# Now you can use the 'data' array as needed#print(data[0])
  # Create x and y values for the line plot using the first two columns of the 'data'array

  x = np.arange(np_data.shape[0]) # Assuming the x-axis represents the index ofthe data points

y = np_data[:, 0] # Assuming you want to plot the first column# Create a line plot
plt.plot(x, y) plt.xlabel('time') plt.ylabel('frequency')
# Turn off the grid lines, axis ticks, and labelsplt.grid(False)
plt.axis('off')


# Save the plot as an image (e.g., PNG format) to your local system
  output_file_path = r"/content/drive/MyDrive/ColabNotebooks/mmwave/temp_file/line_plot.png"

plt.savefig(output_file_path, bbox_inches='tight', pad_inches=0)plt.close()
# Test with the given datacfg = get_cfg()
  cfg.merge_from_file("/content/drive/MyDrive/ColabNotebooks/mmwave/config.yaml")

  cfg.MODEL.WEIGHTS ="/content/drive/MyDrive/ColabNotebooks/mmwave/model_final.pth"

cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.2


  image_path ="/content/drive/MyDrive/ColabNotebooks/mmwave/temp_file/line_plot.png"

im = cv2.imread(image_path) predictor = DefaultPredictor(cfg)outputs = predictor(im)
metadata = MetadataCatalog.get(cfg.DATASETS.TEST[0])v = Visualizer(im[:, :, ::-1],
metadata=metadata,scale=0.8,
```

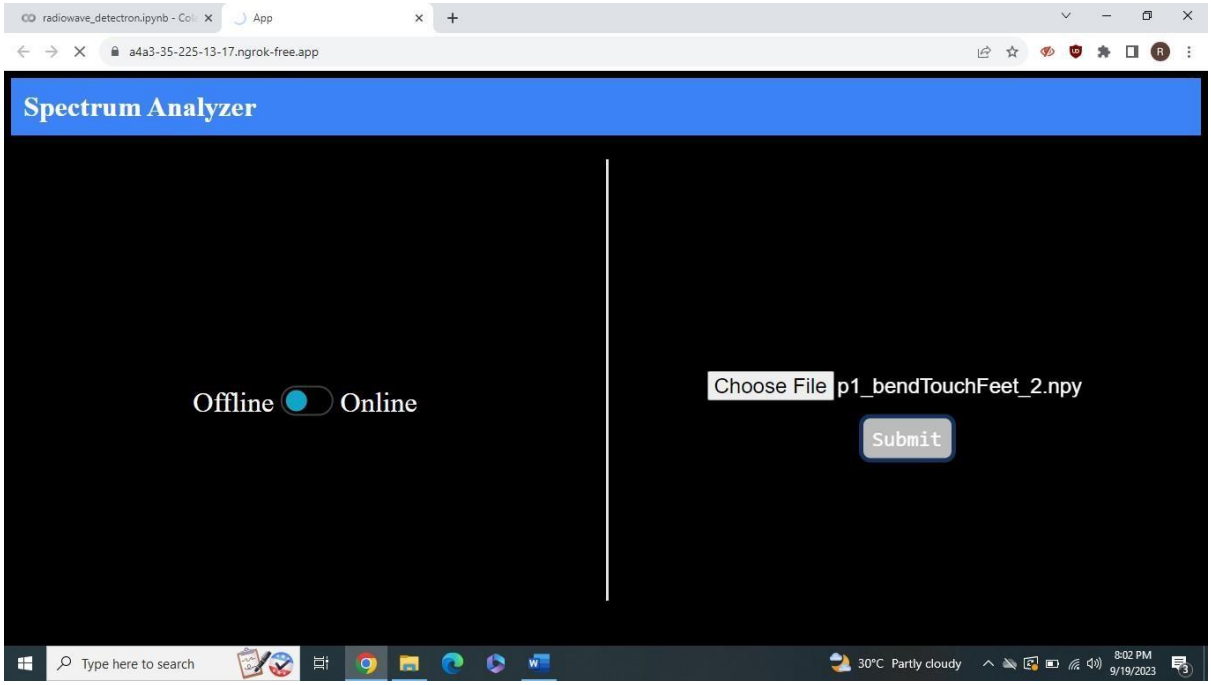
```
instance_mode=ColorMode.IMAGE_BW)

v = v.draw_instance_predictions(outputs["instances"].to("cpu"))cv2_imshow(v.get_image()[:, :, ::-1])
cv2.waitKey(0)


  class_names = ["cross_toe_touch","crunches","hand_rotation","jogging","lunge","lat_squats", "squats"]

  pred_classes = [class_names[i] for i in outputs["instances"].pred_classes.to("cpu")]

# print(pred_classes)


# print(outputs["instances"].pred_boxes)# print(outputs["instances"].scores) tensor = outputs["instances"].scores
# Find the maximum value and its corresponding indexmax_value, max_index = torch.max(tensor, dim=0)
# Convert the maximum value to a Python float (if needed)max_value = max_value.item()
print(pred_classes[0]) print("Maximum Score:", max_value)
# top_indices = torch.topk(tensor, k=2)[1]


# # Get the second maximum value using the top_indices


# second_max_value = tensor[top_indices[1]]


# # Convert the second maximum value to a Python float (if needed) # second_max_value = second_max_value.item()
# print("Second Maximum Score:", second_max_value)# print(pred_classes[1])
# Return the response


response = make_response("result")


  return render_template('app.html' ,inputtext=pred_classes[0],inputValue=max_value*100)

app.run()
```

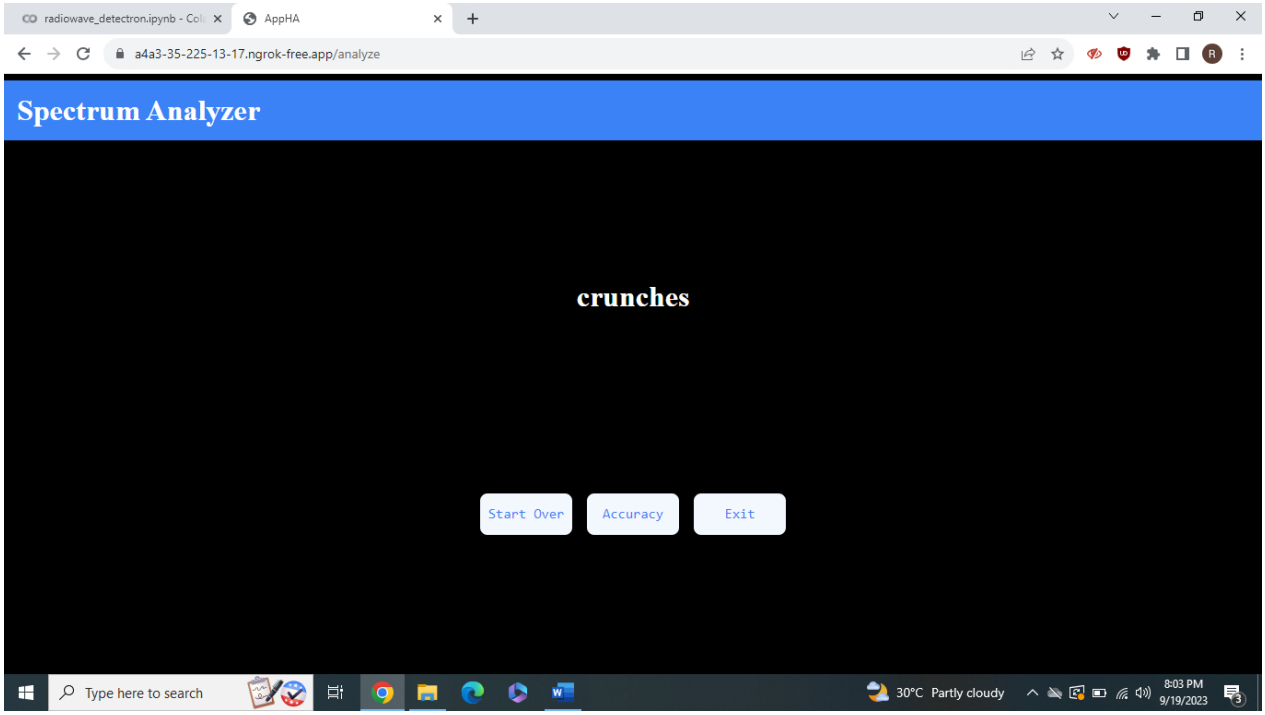
SCREEN SHOTS:

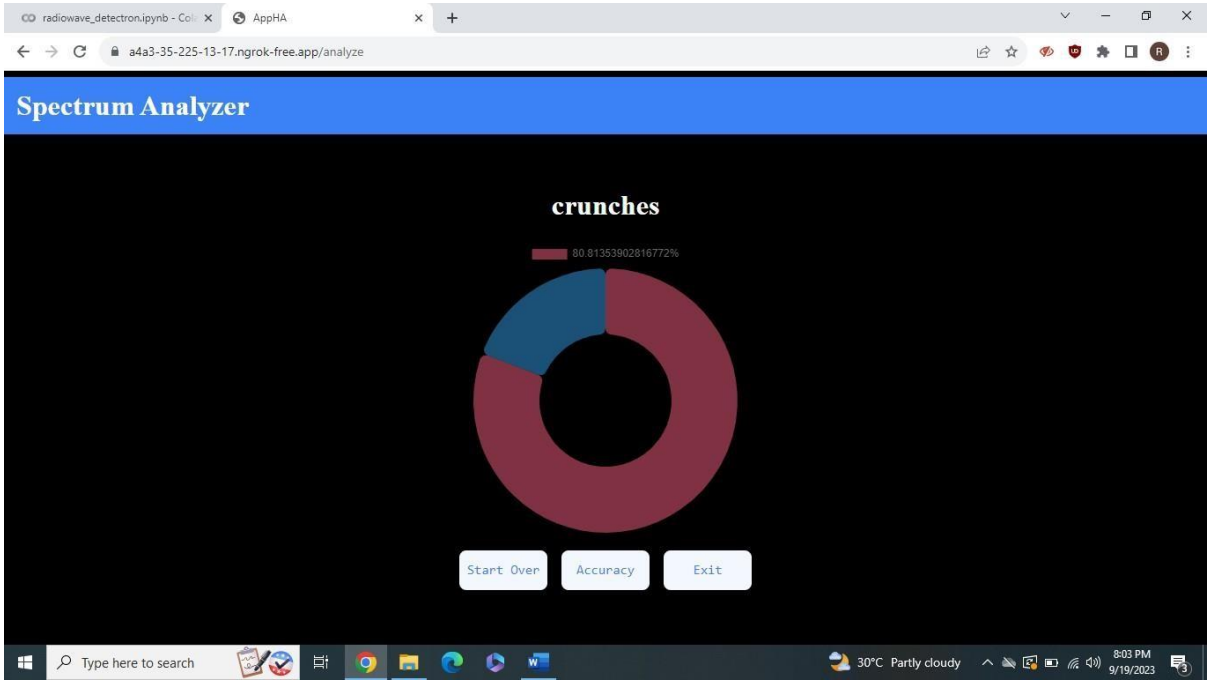





Open
This PC › Desktop › data
Search data
Organize
New folder
payslip
RVS Project
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Name
Date modified
Type
p1_bendTouchFeet_1.npy 10/25/2022 4:06 PM NPY File
p1_bendTouchFeet_2.npy 10/25/2022 4:06 PM NPY File
p1_crunches_1.npy 10/25/2022 4:06 PM NPY File
p1_crunches_2.npy 10/25/2022 4:06 PM NPY File
p3_crunches_1.npy 10/25/2022 4:03 PM NPY File
File name:
Custom Files (*.mat;*.npy;*.bin
Open
Cancel
Choose File No file chosen
Submit
Type here to search
30°C Partly cloudy
8:02 PM
9/19/2023


radiowave_detectron.ipynb - Col
App
a4a3-35-225-13-17.ngrok-free.app
Spectrum Analyzer
Offline
Online
Choose File p1_bendTouchFeet_2.npy
Submit
Type here to search
30°C Partly cloudy
8:02 PM
9/19/2023


radiowave_detectron.ipynb - Col
AppHA
a4a3-35-225-13-17.ngrok-free.app/analyze
Spectrum Analyzer
crunches
Start Over
Accuracy
Exit
Type here to search
30°C Partly cloudy
8:03 PM
9/19/2023


radiowave_detectron.ipynb - Col
AppHA
a4a3-35-225-13-17.ngrok-free.app/analyze
Spectrum Analyzer
crunches
80.81353902816772%
Start Over
Accuracy
Exit
Type here to search
30°C Partly cloudy
8:03 PM
9/19/2023

CONCLUSION:

In this paper, we generated our own human activity dataset using a low-costmmWave radar for the purpose of Human Activity Recognition. Using the radar's raw data, micro-Doppler spectrograms have been created and subsequently used to train different machine and deep learning classifiers. Using the classifiers of the RadHAR paper [9] on a similar dataset we have been able to achieve superior results with the best combination of classifier and filtering method achieving an average accuracy of 99.62%. This could imply that micro-Doppler spectrograms are a superior signal processing option compared to the sparse point clouds, both inperformance a and training data size.