ECG DataSet

Dataset: Sudden Cardiac Death Holter Database

- **Name**: Sudden Cardiac Death Holter Database
- **Source**: Typically sourced from research institutions or databases like PhysioNet.
- **Contents**:
  - ECG recordings collected using Holter monitors from individuals who may be at risk of sudden cardiac death.
  - The recordings are typically longer in duration compared to standard ECG datasets and capture the heart's electrical activity over an extended period, often spanning 24 hours or more.
  - Annotations indicating events related to sudden cardiac death, such as arrhythmias, conduction abnormalities, and other cardiac events.
- **Format**: The database may be provided in various formats, but commonly in CSV format, where each row represents a data sample (typically a segment of the ECG recording), and columns represent features extracted from the ECG signals, such as RR intervals, peak values, intervals, morphological characteristics, etc.
- **Usage**: Researchers and healthcare professionals utilise this database to develop algorithms and models for predicting the risk of sudden cardiac death, identifying potential triggers or predictors of arrhythmias, and studying the underlying mechanisms of cardiac events leading to sudden death.

Dataset: MIT-BIH Arrhythmia Database

- **Name**: MIT-BIH Arrhythmia Database
- **Source**: PhysioNet
- **Contents**:
  - ECG recordings from 48 different subjects.
  - Each recording consists of multiple leads and contains information about the electrical activity of the heart over time.
  - Annotations indicating the presence of different types of arrhythmias (e.g., ventricular ectopic beats, supra-ventricular ectopic beats, fusion beats, etc.).
- **Format**: Typically provided in CSV format, where each row represents a data sample, and columns represent various features extracted from the ECG signals, such as RR intervals, peak values, intervals, morphological characteristics, etc.
- **Usage**: This dataset is commonly used for developing and evaluating machine learning models for ECG signal processing tasks, such as arrhythmia detection, classification, and anomaly detection.

Deep Learning Models

1. ***Artificial Neural Network (ANN) Model:***

- **Purpose:** ANNs are versatile models capable of learning complex patterns from data. They consist of interconnected layers of neurons that process input data and learn to map it to output labels.

- **Architecture:** ANNs typically consist of an input layer, one or more hidden layers, and an output layer. Each layer contains multiple neurons, and connections between neurons have associated weights that are adjusted during training.

- **Usage:** In the provided code, the ANN model is used for tasks such as ECG signal classification or reconstruction. It learns to map input features to output labels or reconstruct input data based on learned patterns.

- **Implementation:** The ANN model is implemented using TensorFlow's Keras API. It consists of densely connected layers with activation functions like ReLU to introduce non-linearity. The final layer's activation depends on the task, such as softmax for classification or linear for regression.

2. ***Gated Recurrent Unit (GRU) Model:***

- **Purpose:** The GRU model is a type of recurrent neural network (RNN) designed for sequential data processing tasks, such as time series analysis, natural language processing (NLP), and speech recognition.
- **Architecture:** GRU is similar to the more widely known Long Short-Term Memory (LSTM) model but with fewer parameters, making it computationally less expensive. It consists of a gating mechanism that regulates the flow of information within the network, allowing it to retain relevant information over long sequences while avoiding the vanishing gradient problem.
- **Usage:** In the provided code, the GRU model is used for ECG signal classification, specifically for detecting arrhythmias based on features extracted from the ECG signal.
- **Implementation:** The GRU model is implemented using TensorFlow's Keras API. It consists of one or more GRU layers followed by a dense layer with a sigmoid activation function for binary classification.

3. ***Convolutional Neural Network (CNN) Model:***

- **Purpose:** CNNs are primarily designed for image recognition tasks, but they are also effective for sequential data analysis, such as time series data like ECG signals. They can automatically learn hierarchical patterns and spatial dependencies within the data.

- **Architecture:** CNNs consist of convolutional layers followed by pooling layers to extract features and reduce dimensionality. These layers are typically interleaved with activation functions like ReLU to introduce non-linearity. The final layers are fully connected layers for classification or regression tasks.

- **Usage:** In the provided code, the CNN model is used for ECG signal classification, where it learns to identify different types of arrhythmias based on patterns in the ECG signal.

- **Implementation:** The CNN model is implemented using TensorFlow's Keras API. It typically comprises convolutional layers with filters of varying sizes, followed by pooling layers to downsample the feature maps. The final layers are dense layers with softmax activation for multi-class classification.

4. ***Long Short-Term Memory (LSTM) Model:***

- **Purpose:** LSTMs are a type of recurrent neural network (RNN) designed to handle sequence data with long-term dependencies. They are well-suited for tasks involving time series data, natural language processing, and speech recognition.

- **Architecture:** LSTMs have a more complex architecture compared to traditional RNNs, with additional gating mechanisms to control the flow of information. They have a memory cell that can maintain information over long sequences, helping overcome the vanishing gradient problem.

- **Usage:** In the provided code, the LSTM model is used for tasks such as ECG signal classification or prediction. It learns to capture temporal dependencies and patterns in the sequential data.

- **Implementation:** The LSTM model is implemented using TensorFlow's Keras API. It consists of LSTM layers followed by dense layers for classification or regression tasks. The LSTM layers are responsible for capturing temporal dependencies, while the dense layers perform the final mapping to output labels or values.

5. ***Autoencoder Model:***

- **Purpose:** Autoencoder is a type of neural network used for unsupervised learning and dimensionality reduction tasks. It learns to encode input data into a lower-dimensional representation and then decode it back to its original form, aiming to minimize reconstruction error.

- **Architecture:** The autoencoder consists of an encoder network, which compresses the input data into a latent space representation, and a decoder network, which reconstructs the original input from the latent space representation. The encoder and decoder typically mirror each other in terms of architecture.

- **Usage:** In the provided code, the autoencoder model is used for ECG signal reconstruction. It learns to reconstruct ECG signals from their compressed representations, potentially capturing important features or anomalies in the process.

- **Implementation:** The autoencoder model is implemented using a fully connected neural network architecture in TensorFlow's Keras API. It comprises multiple dense layers for both the encoder and decoder, with the number of neurons decreasing and then increasing, respectively.