

```
graph LR; User[User wears Myo] --> Myo[Myo]; Myo --> Bluetooth[Bluetooth]; Bluetooth --> Phone[Phone]; Phone -- "EMG signals sent to cloud" --> Cloud((Cloud)); Cloud -- "EMG signals downloaded and cut in PC" --> PC[PC]; PC -- "Cut EMG signals saved into .csv files" --> Files[Files];
```

The diagram illustrates the workflow for processing EMG signals. It begins with a user wearing a Myo device, which transmits signals via Bluetooth to a smartphone. The smartphone then sends the EMG signals to a cloud storage service. From the cloud, the signals are downloaded to a PC, where they are processed and saved into CSV files.

The flowchart illustrates the data partitioning process for the proposed model. It starts with 'N-Recordings' at the top. A diagonal arrow points from 'N-Recordings' to a box labeled '-20% Evaluation Set'. A vertical arrow points from 'N-Recordings' down to a box labeled 'M-Recordings'. From 'M-Recordings', five diagonal arrows, each labeled '20%', point down to five boxes labeled '1', '2', '3', '4', and '5' in the 'Test folds' section. Below these, in the 'Train folds' section, are five more boxes labeled '1', '2', '3', '4', and '5'. Every box in the 'Test folds' section is connected to every box in the 'Train folds' section by a straight line, representing a fully connected cross-validation setup.

Diagram illustrating the data structure for the 'Time' variable. The data is organized into a table with **N rows** and **9 cols**. The columns are labeled: **Time**, **ch1**, **ch2**, **ch3**, **ch4**, **ch5**, **ch6**, **ch7**, and **ch8**. The **Time** column is highlighted in light blue. A bracket on the right indicates that the **Time** column is concatenated across all rows into a single array.

The diagram illustrates the process of feature extraction and model training for EEG data. It starts with a "Discarded time vector" which is a 128x8 array of channels (ch1 to ch8). This vector is processed "For each window" to produce a 128x8 feature matrix. This matrix is then processed by 11 different feature extraction methods: rms, smooth, STFT, haar, db8, sym4, sym8, bior1.3, bior2.2, coif3, and coif4. The resulting features are used to train three different models: Train folds, Test folds, and Evaluation set. Each model is trained on N Rows x M cols of data. A note states: "N is the number of recordings and M depends on the processing we apply".

```

graph TD
    Input["Discarded time vector  
ch1 ch2 ch3 ch4 ch5 ch6 ch7 ch8  
128  
128  
128  
128"] -- "For each window  
run the features" --> FeatureMatrix["128x8"]
    FeatureMatrix --> rms
    FeatureMatrix --> smooth
    FeatureMatrix --> STFT
    FeatureMatrix --> haar
    FeatureMatrix --> db8
    FeatureMatrix --> sym4
    FeatureMatrix --> sym8
    FeatureMatrix --> bior13["bior1.3"]
    FeatureMatrix --> bior22["bior2.2"]
    FeatureMatrix --> coif3
    FeatureMatrix --> coif4
    rms --> Train["Train folds  
N Rows x M cols"]
    smooth --> Train
    STFT --> Train
    haar --> Train
    db8 --> Train
    sym4 --> Train
    sym8 --> Train
    bior13 --> Train
    bior22 --> Train
    coif3 --> Train
    coif4 --> Train
    rms --> Test["Test folds  
N Rows x M cols"]
    smooth --> Test
    STFT --> Test
    haar --> Test
    db8 --> Test
    sym4 --> Test
    sym8 --> Test
    bior13 --> Test
    bior22 --> Test
    coif3 --> Test
    coif4 --> Test
    rms --> Eval["Evaluation set  
N Rows x M cols"]
    smooth --> Eval
    STFT --> Eval
    haar --> Eval
    db8 --> Eval
    sym4 --> Eval
    sym8 --> Eval
    bior13 --> Eval
    bior22 --> Eval
    coif3 --> Eval
    coif4 --> Eval

```

The flowchart illustrates the classification process for two tasks: Extension and Flexion. It is organized into two main sections, one for training and one for testing.

Training Section (Left):

- Inputs: **Extension** and **Flexion** data, both labeled **TrainData_Feature_rms**.
- Each input is associated with a **Fold Number** box containing the value **1**.
- The data from both folds is combined in a **Concatenate** step.
- The concatenated data is used to **Train classifier**.

Testing Section (Right):

- Inputs: **Extension** and **Flexion** data, both labeled **TestData_Feature_rms**.
- Each input is associated with a **Fold Number** box containing the value **1**.
- The data from both folds is combined in a **Concatenate** step.

Classification and Output:

- The output from the **Train classifier** and the concatenated test data are fed into the **Classify** step.
- The **Classify** step produces two outputs: **Classification Errors** and **Confusion Matrix**.