|  |
| --- |
|  |
| Driver Drowsiness Detection |

|  |
| --- |
| October 23, 2024 |

Table of Contents

[**Data Source** 4](#_Toc180386156)

[Dataset Description 4](#_Toc180386157)

[**Figure:** Experimental design. 4](#_Toc180386158)

[Electrode Layout in EEG Cap 5](#_Toc180386159)

[**Figure:** EEG Cap Electrode Layout. Blue electrodes represent the 10–20 system, and green indicates additional electrodes. Contact impedance was maintained below 5 kΩ for all electrodes. 6](#_Toc180386160)

[Data Recording and Storage 6](#_Toc180386161)

[Data extraction 7](#_Toc180386162)

[**System Workflow** 8](#_Toc180386163)

[**Figure:** System Workflow Diagram 8](#_Toc180386164)

[**Figure:** System Block Diagram 9](#_Toc180386165)

[**Data Manipulation** 10](#_Toc180386166)

[Data Manipulation Methodology 10](#_Toc180386167)

[**Figure:** Flowchart of the Data Manipulation Methodology. 10](#_Toc180386168)

[Data Manipulation Result 11](#_Toc180386169)

[References 11](#_Toc180386170)

[**Noise Cancellation: Savitzky-Golay** 12](#_Toc180386171)

[**Table:** Altering window size and polynomial order with KNN accuracy to find the optimal values. 12](#_Toc180386172)

[**Figure:** KNN model accuracy across different window lengths and polynomial orders 12](#_Toc180386173)

[Signal Visualization 13](#_Toc180386174)

[**Figure:** Alert Data Visualization of unfiltered EEG signals (Oz) vs. filtered Oz signals. 13](#_Toc180386175)

[**Figure:** Drowsy Data visualization of unfiltered EEG signals (Oz) vs. filtered Oz signals. 14](#_Toc180386176)

[Machine Learning Evaluation with Savitzky-Golay Filter 15](#_Toc180386177)

[**Table:** Machine Learning Evaluation with Filtered Dataset 15](#_Toc180386178)

[**Figure:** ROC Curve for Machine Learning Results with Filtered Dataset. 15](#_Toc180386179)

[**Feature Selection - Shapley Approximation (shap)** 16](#_Toc180386180)

[**Figure:** Feature Importance SHAP Values 16](#_Toc180386181)

[**Figure:** Summary Plot for Feature Impact Distribution. 17](#_Toc180386182)

[**Figure:** Electrode Layout on EEG Cap Sorted by Feature Importance Values 18](#_Toc180386183)

[**Table:** Feature Importance with Corresponding Denotations 18](#_Toc180386184)

[References 19](#_Toc180386185)

[**Machine Learning Results** 20](#_Toc180386186)

[Mean Accuracy 20](#_Toc180386187)

[**Table:** Mean accuracy comparison across various feature sets. 20](#_Toc180386188)

[**Figure:** Mean accuracy comparison across various feature sets. 20](#_Toc180386189)

[Mean Precision 21](#_Toc180386190)

[**Table:** Mean precision comparison across various feature sets. 21](#_Toc180386191)

[**Figure:** Mean precision comparison across various feature sets. 21](#_Toc180386192)

[Mean Recall 22](#_Toc180386193)

[**Table:** Mean recall comparison across various feature sets. 22](#_Toc180386194)

[**Figure:** Mean recall comparison across various feature sets. 22](#_Toc180386195)

[Mean F1 Score 23](#_Toc180386196)

[**Table:** Mean F1 Score comparison across various feature sets. 23](#_Toc180386197)

[**Figure:** Mean F1 Score comparison across various feature sets. 23](#_Toc180386198)

[Training Time of Machine Learning Model 24](#_Toc180386199)

[**Table:** Machine learning model training time across various feature sets 24](#_Toc180386200)

[**Figure:** Machine learning model training time across various feature sets 24](#_Toc180386201)

[Cross Validation 25](#_Toc180386202)

[**Figure:** Cross Validation Workflow 25](#_Toc180386203)

[**Deep Learning** 26](#_Toc180386204)

[**Artificial Neural Network Architecture** 26](#_Toc180386205)

[The input and hidden layers 26](#_Toc180386206)

[Compilation and Training 26](#_Toc180386207)

[**Figure:** Artificial Neural Network Architecture 27](#_Toc180386208)

[**Long short-term memory Architecture** 28](#_Toc180386209)

[The input and hidden layers 28](#_Toc180386210)

[Compilation and Training 28](#_Toc180386211)

[**Figure:** Long short-term memory Architecture 29](#_Toc180386212)

[**Gated Recurrent Unit Networks Architecture** 30](#_Toc180386213)

[The input and hidden layers 30](#_Toc180386214)

[Compilation and Training 30](#_Toc180386215)

[**Figure:** Gated Recurrent Unit Networks Architecture 31](#_Toc180386216)

[**Ensemble Learning** 32](#_Toc180386217)

[Ensemble Learning Architecture 32](#_Toc180386218)

[Meta-Model Training 32](#_Toc180386219)

[**Figure:** Ensemble Learning Architecture 33](#_Toc180386220)

[Deep Learning Results 34](#_Toc180386221)

[Mean Accuracy 34](#_Toc180386222)

[**Figure:** Box Plot of Mean Accuracy Across Feature Sets in Deep Learning Models 34](#_Toc180386223)

[Mean Precision 35](#_Toc180386224)

[**Figure:** Box Plot of Mean Precision Across Feature Sets in Deep Learning Models 35](#_Toc180386225)

[Mean Recall 36](#_Toc180386226)

[**Figure:** Box Plot of Mean Recall Across Feature Sets in Deep Learning Models 36](#_Toc180386227)

[Mean F1-Score 37](#_Toc180386228)

[**Figure:** Box Plot of Mean F1 Score Across Feature Sets in Deep Learning Models 37](#_Toc180386229)

[Training Time (in seconds) 38](#_Toc180386230)

[**Figure:** Comparison of Deep Learning Model Training Times Across Feature Sets. 38](#_Toc180386231)

[**Overall Performance Comparison** 39](#_Toc180386232)

[Mean Accuracy 39](#_Toc180386233)

[**Figure:** Comparison of Mean Accuracy Across Models with Varying Feature Sets 39](#_Toc180386234)

[Mean F1-Score 40](#_Toc180386235)

[**Figure:** Comparison of Mean F1 Score Across Models with Varying Feature Sets 40](#_Toc180386236)

[Performance Comparison Using 16 Features 41](#_Toc180386237)

[**Figure:** Bar Plots Comparing Accuracy of Machine Learning and Deep Learning Models 41](#_Toc180386238)

[**Figure:** Bar Plots Comparing F1 Score of Machine Learning and Deep Learning Models 42](#_Toc180386239)

[Training Time Comparison Using 16 Features 43](#_Toc180386240)

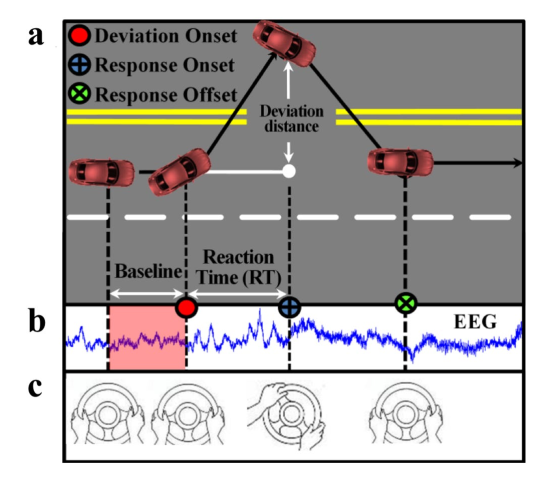
[**Figure:** Training Time Comparison of All Models Using 16 Features 43](#_Toc180386241)

# **Data Source**

## Dataset Description

An open EEG dataset released in 2019 by Cao et al. was used in our study. The study involved 27 voluntary participants aged 22–28 years, consisting of National Chiao Tung University students and staff. They participated in a 90-minute sustained-attention driving task multiple times on the same or different days, resulting in 62 EEG datasets. Participants were required to have normal or corrected-to-normal vision and no history of sleep deprivation, drug abuse, or irregular sleep patterns prior to the experiments. They avoided alcohol, caffeine, and strenuous exercise a day before the experiments. A VR driving environment was created using a dynamic driving simulator mounted on a six-degree-of-freedom Stewart motion platform. Six interactive highway driving scenes synchronized over local area networks were projected onto the screens at viewing angles of 0°, 42°, 84°, 180°, 276°, and 318° to provide a nearly complete 360° visual field.  An event-related lane-departure paradigm3 was implemented in the VR-based driving simulator using WorldToolKit (WTK) R9 Direct and Visual C++.  The driving task involved a visually monotonous night on a straight four-lane highway that mirrors real-life driving conditions. The lane-departure paradigm measured participants' reaction times to perturbations during a continuous driving task. Participants were instructed to keep the car in the center of the lane and respond to random lane-departure events by steering the car back to the original lane. The task lasted 90 minutes without breaks, and participants' activities were monitored via a surveillance video camera.

**Reference Link:** <https://www.nature.com/articles/s41597-019-0027-4>



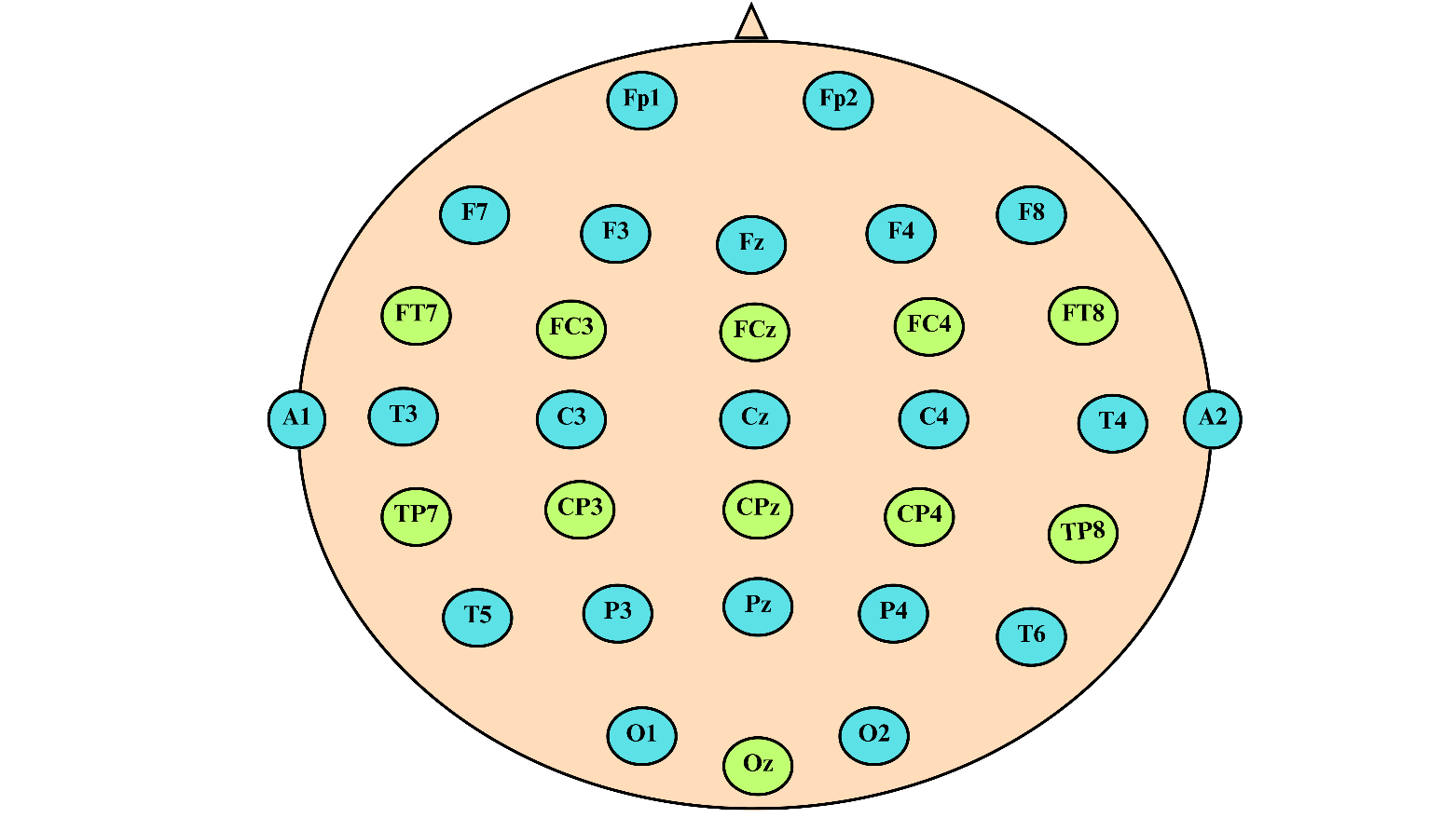
### **Figure:** Experimental design.

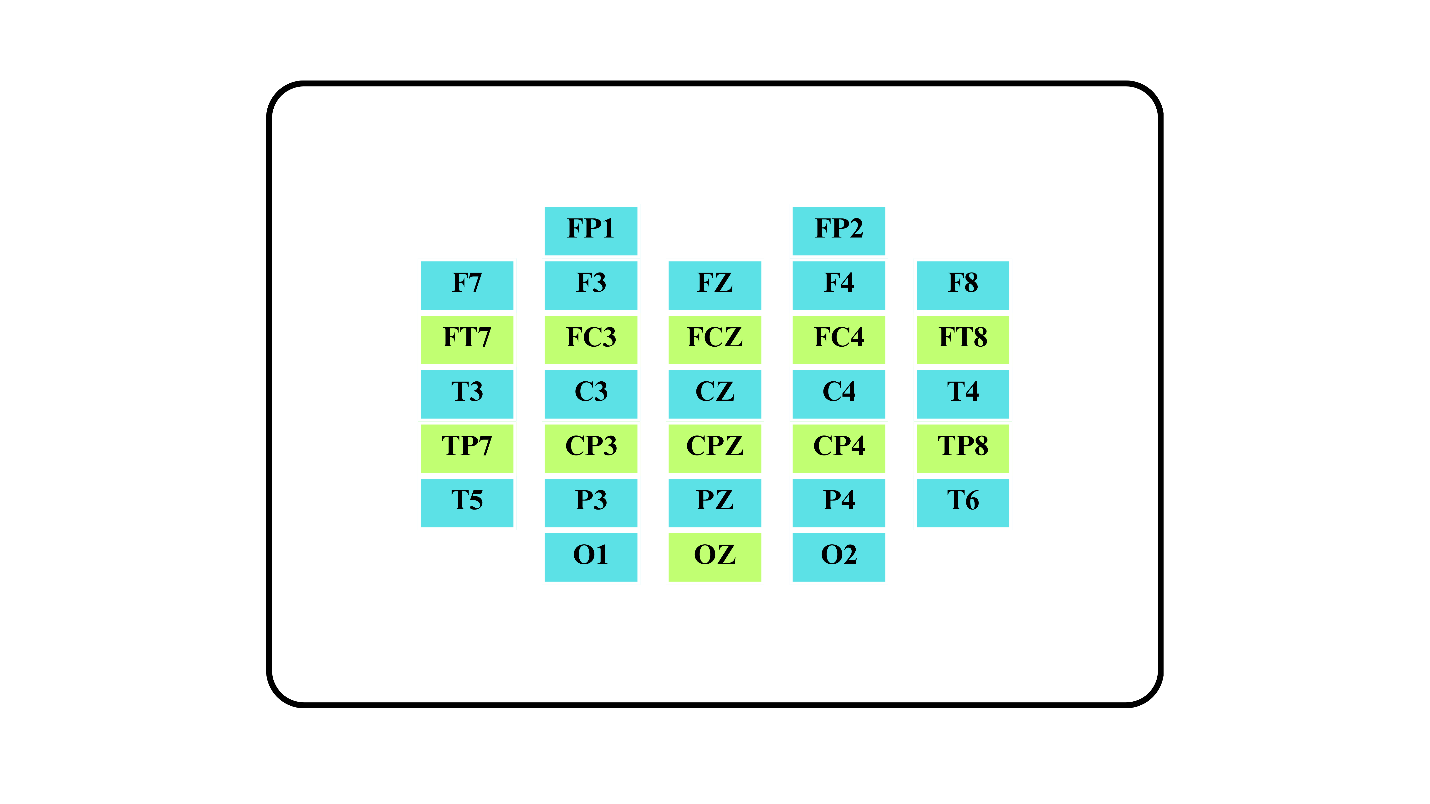
1. Event-related lane-deviation task. (b, c) Simultaneous EEG and behavioral recording.

As shown in (Fig. a), lane-departure events were randomly induced to make the car drift from the original cruising lane towards the left or right sides (deviation onset). Each participant was instructed to quickly compensate for this perturbation by steering the wheel (response onset) to cause the car to move back to the original cruising lane (response offset). To avoid the impacts of other factors during the task, participants only reacted to the lane-perturbation event by turning the steering wheel. They did not have to control the accelerator or brake pedals in this experiment. Each lane-departure event was defined as a “trial,” including a baseline period, deviation onset, response onset, and response offset. EEG signals were recorded simultaneously (Fig. b).

The corresponding directions for turning the steering wheel are shown in (Fig. c.) Of note, the next trial occurred within a 5–10 second interval after finishing the current trial, during which the subject had to maneuver the car back to the center line of the third car lane. If the participant fell asleep during the experiment, no feedback was provided to alert him/her.

## Electrode Layout in EEG Cap





### **Figure:** EEG Cap Electrode Layout. Blue electrodes represent the 10–20 system, and green indicates additional electrodes. Contact impedance was maintained below 5 kΩ for all electrodes.

## Data Recording and Storage

EEG data were recorded using a Scan SynAmps2 Express system with 32 Ag/AgCl electrodes and were digitized at 500 Hz. The stimulus computer recorded Vehicle trajectories and events in a log file, which synchronized triggers with the EEG acquisition system. The data were integrated into a new file and imported into EEGLAB in MATLAB.

The EEG signals included 32 channels from electrodes placed according to a modified international 10–20 system. The recordings were analyzed using the EEGLAB toolbox in MATLAB, and events were classified as deviation onset, response onset, or response offset.

The first 32 signals were from the Fp1, Fp2, F7, F3, Fz, F4, F8, FT7, FC3, FCZ, FC4, FT8, T3, C3, Cz, C4, T4, TP7, CP3, CPz, CP4, TP8, A1, T5, P3, PZ, P4, T6, A2, O1, Oz and O2 electrodes. Two electrodes (A1 and A2) were references placed on the mastoid bones. The 33rd signal was used to describe the position of the simulated vehicle. A wired EEG cap with 30 EEG electrodes and two reference electrodes, placed according to a modified international 10–20 system, sampled the EEG signals at 500 Hz throughout the experiment.

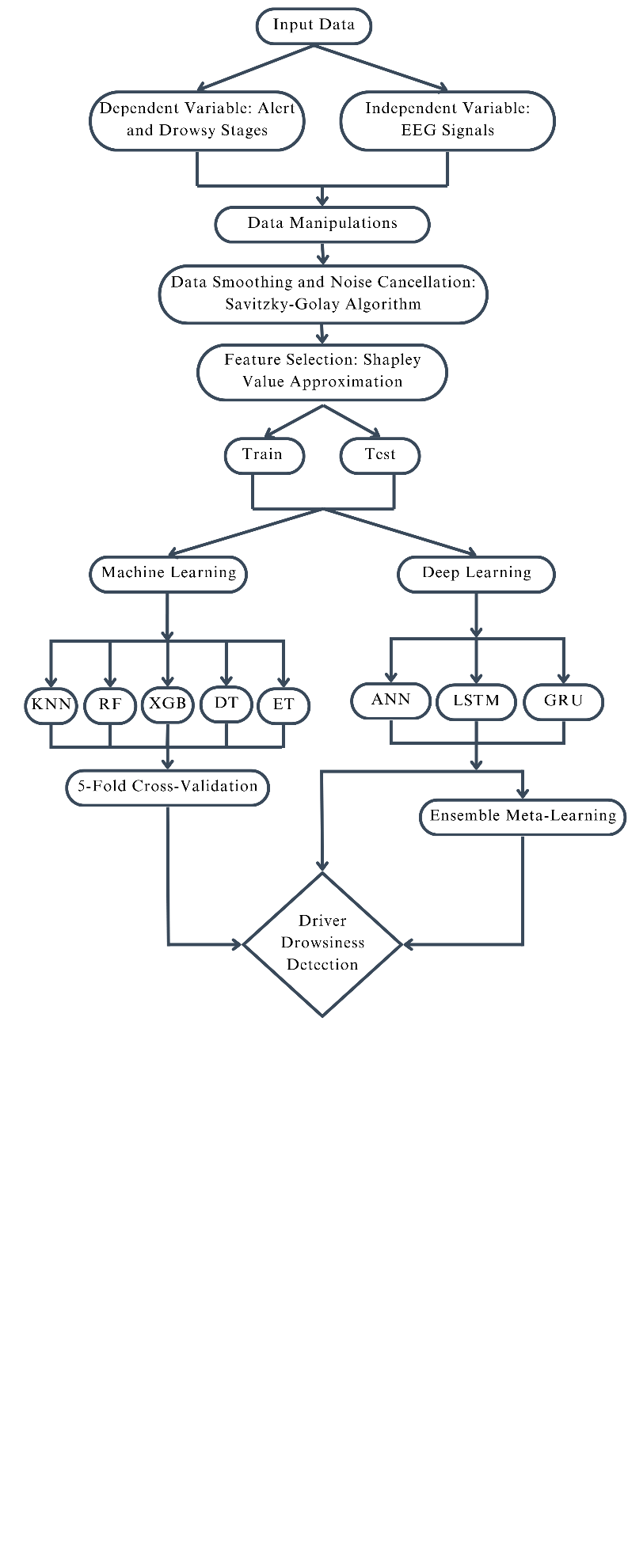
## Data extraction

The dataset has already been preprocessed by its authors in the following steps:

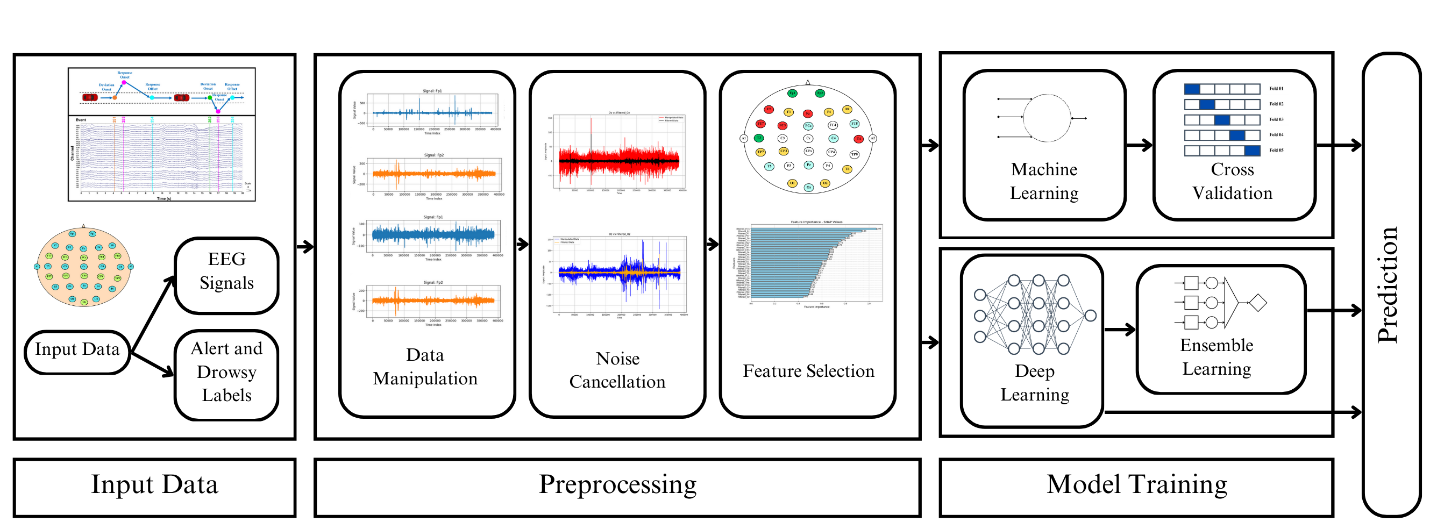
* The raw EEG signals were filtered by 1-Hz high-pass and 50-Hz low-pass finite impulse response (FIR) filters.
* For artifact rejection, apparent eye blink contamination was manually removed. Ocular and muscular artifacts were removed using the Automatic Artifact Removal (AAR) plug-in provided in EEGLAB.
* The original data was further down-sampled from 500 Hz to 128 Hz. 3s-long EEG samples were extracted prior to deviation onset for each trial.

**Reference Link:** <https://arxiv.org/abs/2106.00613>

# **System Workflow**



### **Figure:** System Workflow Diagram



### **Figure:** System Block Diagram

# **Data Manipulation**

## Data Manipulation Methodology

Segregation:

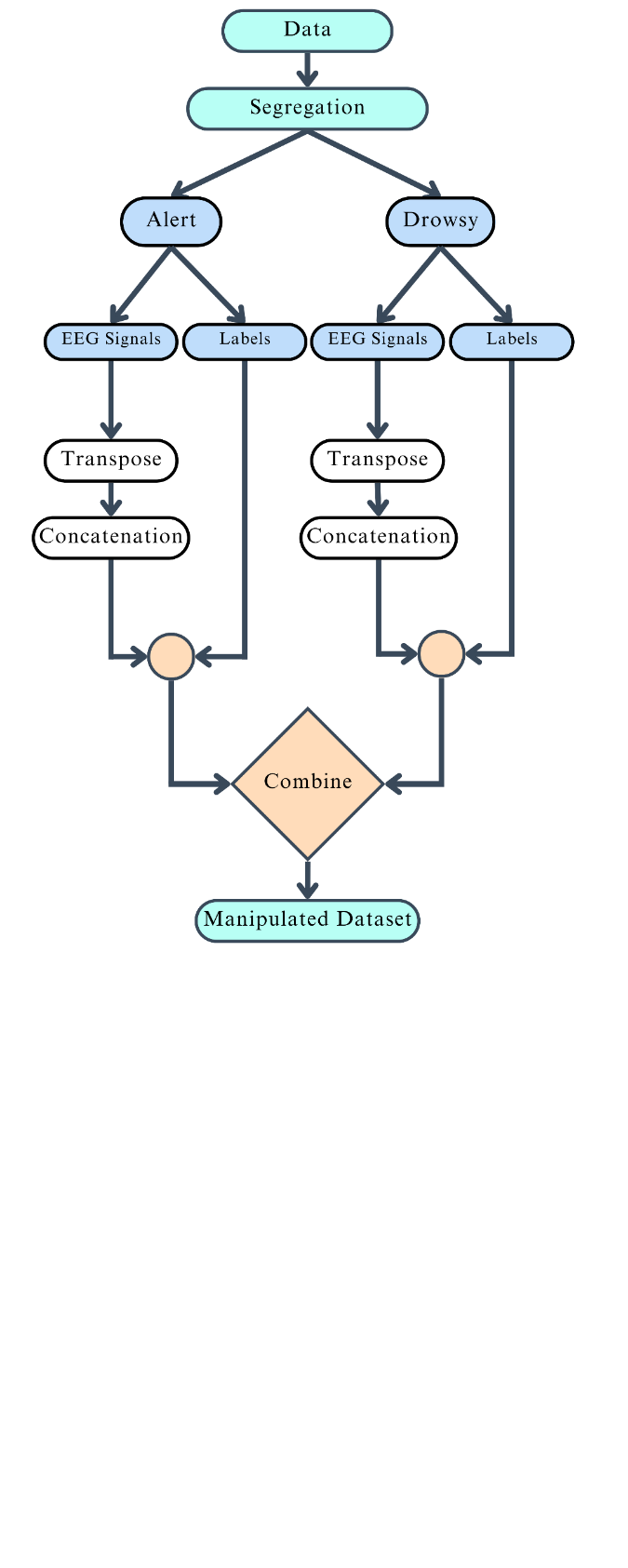
* The EEG samples are segregated into alert and drowsy states for all the subjects based on the 'substate' variable.
* The segregated alert and drowsy data are saved into new '.mat' files.

Manipulating alert and drowsy state data:

* The alert data is loaded and transposed, concatenating all EEG samples into a new data frame.
* Similar manipulation is performed on the drowsy state data.

Creating Combined Dataset:

* The manipulated alert and drowsy datasets are concatenated row-wise into a new data frame.

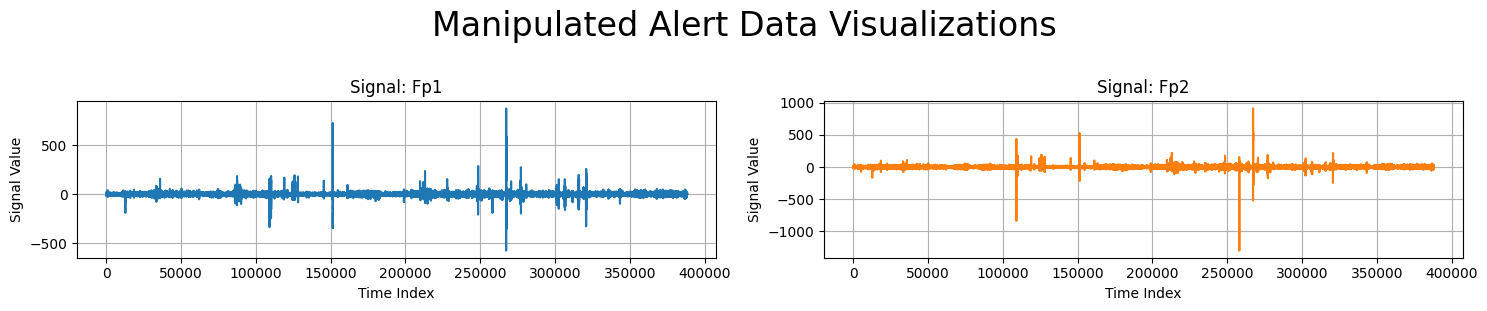


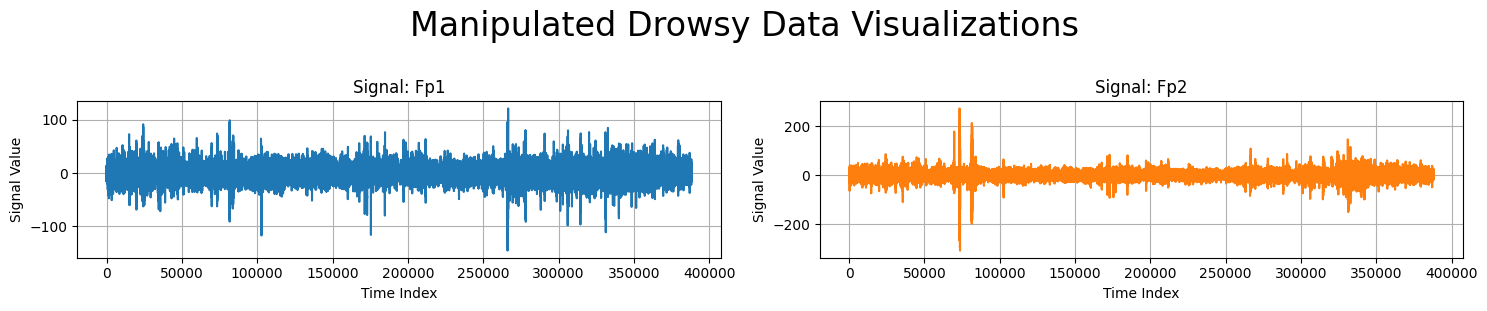
### **Figure:** Flowchart of the Data Manipulation Methodology.

## Data Manipulation Result

Prior to manipulation, the initial EEG dataset contained 2022 samples, each with 30 channels and 384 data points, recorded over 3 seconds at a sampling rate of 128Hz from 11 subjects. The shape of each alert and drowsy states dataset was (1011, 384, 30) [1].

After manipulation, the dataset was transformed into a flat structure with 776448 rows and 31 columns where the additional column represented the 'substate' variable, which contains alert (0) and drowsy (1) states. This manipulation resulted in a new data frame for both alert and drowsy states, each with a shape of (388224, 31).





**Figure:** Visualization of Manipulated Alert and Drowsy Data for Signal Fp1 and Fp2

## References

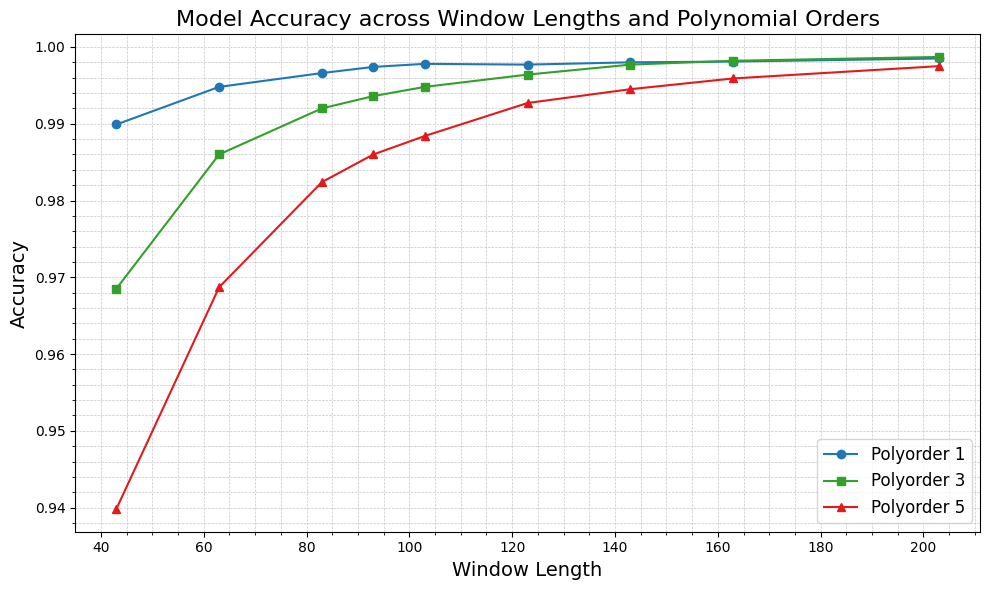
[1] J. Cui *et al.*, “A compact and interpretable convolutional neural network for cross-subject driver drowsiness detection from single-channel EEG,” *Methods*, vol. 202, pp. 173–184, Jun. 2022, doi: 10.1016/j.ymeth.2021.04.017.

# **Noise Cancellation: Savitzky-Golay**

Determining the optimal window size and polynomial order for the Savitzky-Golay filter.

|  |  |  |  |
| --- | --- | --- | --- |
| Window Size | Polyorder = 1 | Polyorder = 3 | Polyorder = 5 |
| 43 | 0.9899 | 0.9685 | 0.9398 |
| 63 | 0.9948 | 0.9860 | 0.9687 |
| 83 | 0.9966 | 0.9920 | 0.9824 |
| 93 | 0.9974 | 0.9936 | 0.9860 |
| 103 | 0.9978 | 0.9948 | 0.9884 |
| 123 | 0.9977 | 0.9964 | 0.9927 |
| 143 | 0.9980 | 0.9977 | 0.9945 |
| 163 | 0.9981 | 0.9982 | 0.9959 |
| 203 | 0.9985 | 0.9987 | 0.9975 |

### **Table:** Altering window size and polynomial order with KNN accuracy to find the optimal values.

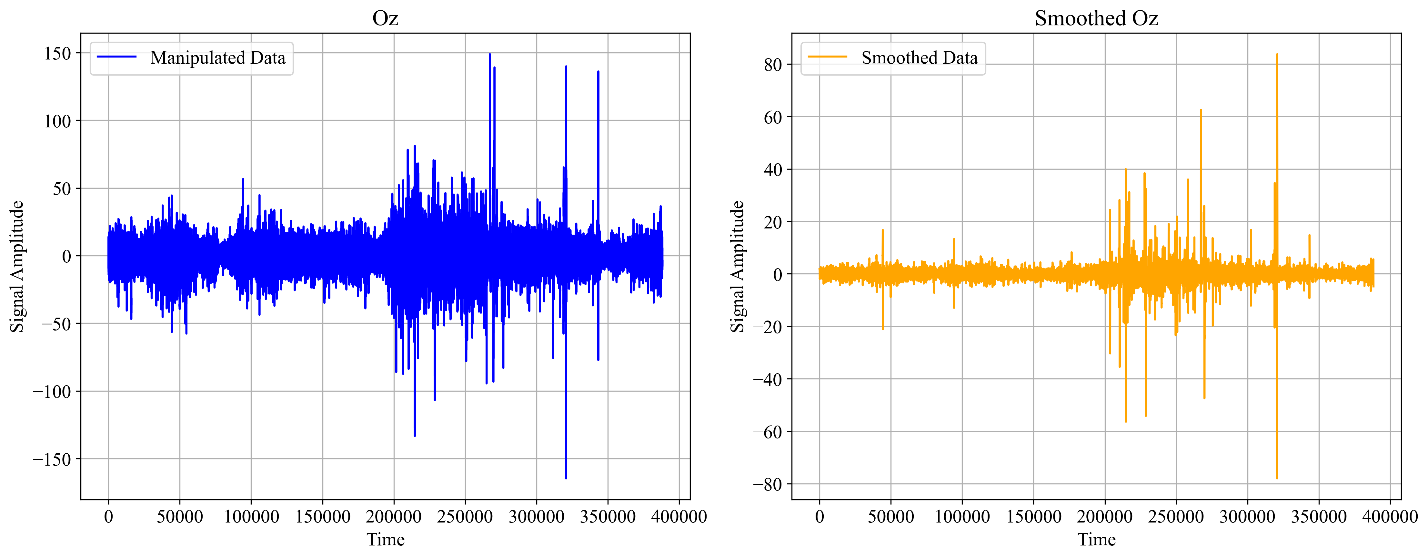


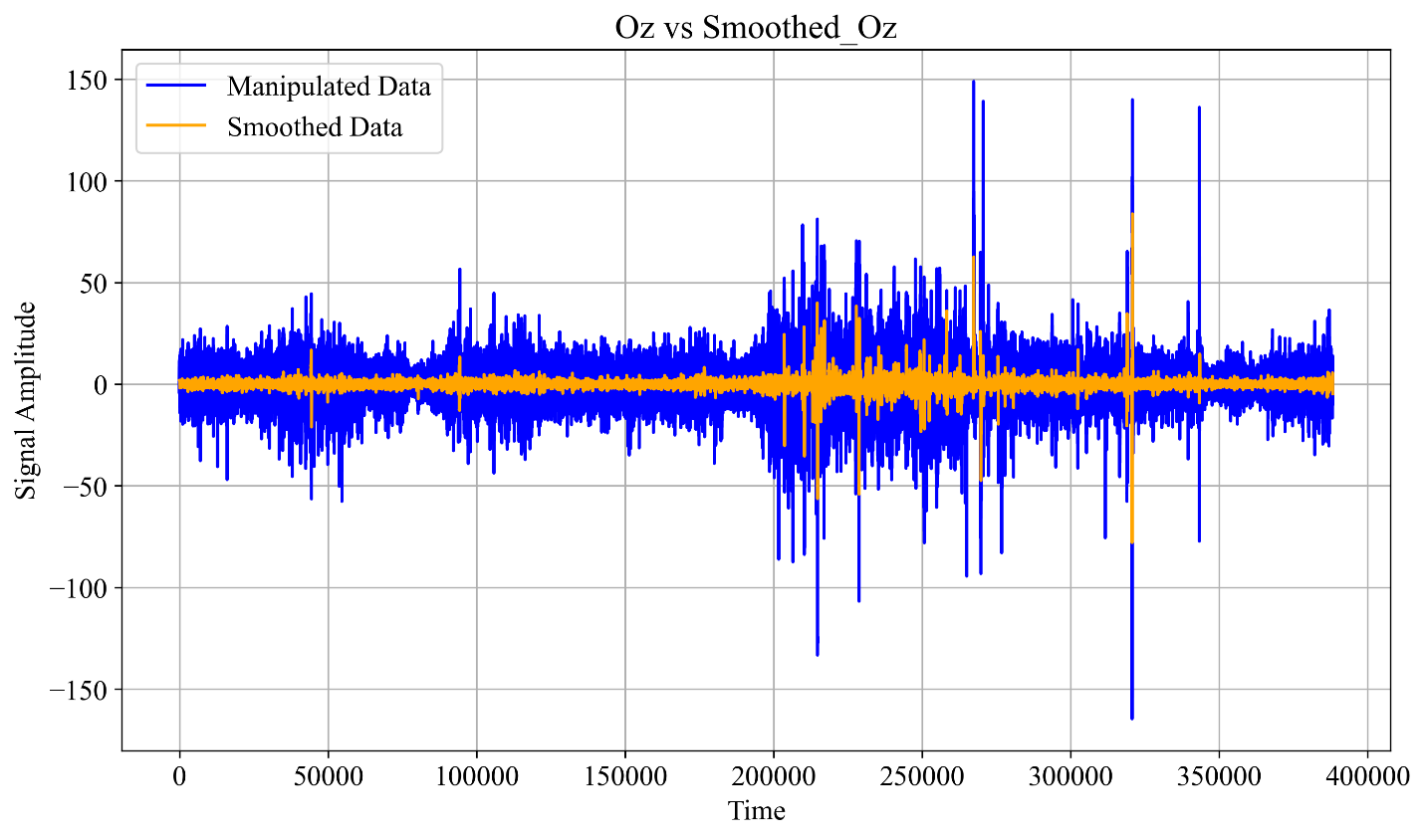
### **Figure:** KNN model accuracy across different window lengths and polynomial orders

## Signal Visualization

Polyorder = 3, Window Size = 203

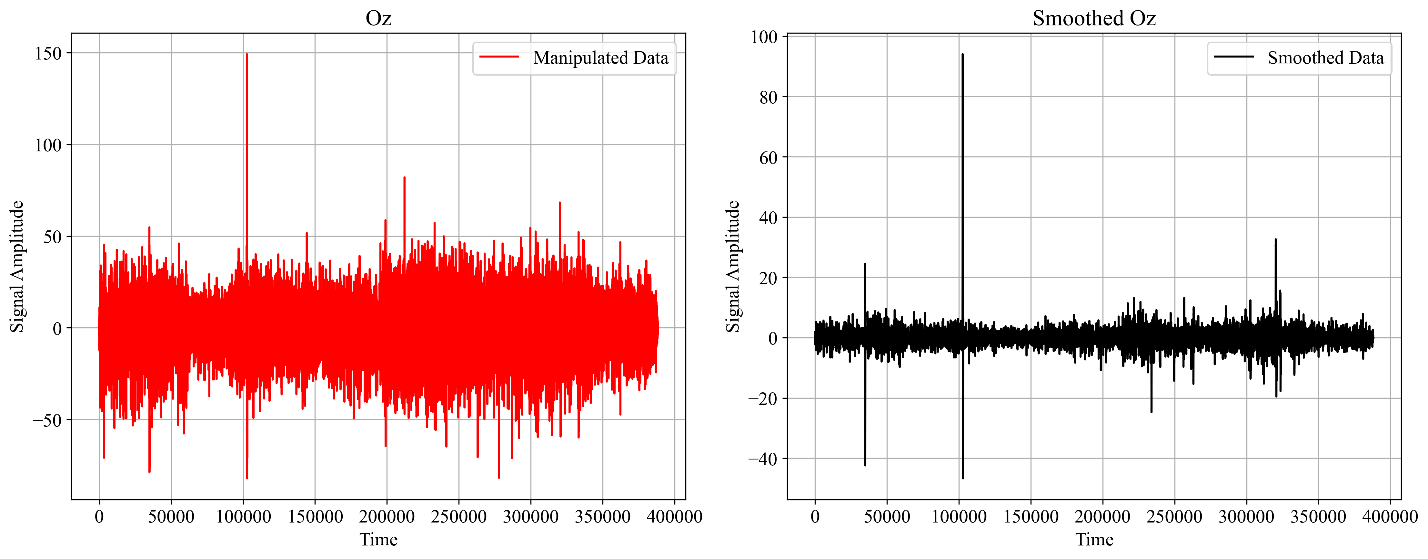
Alert Data

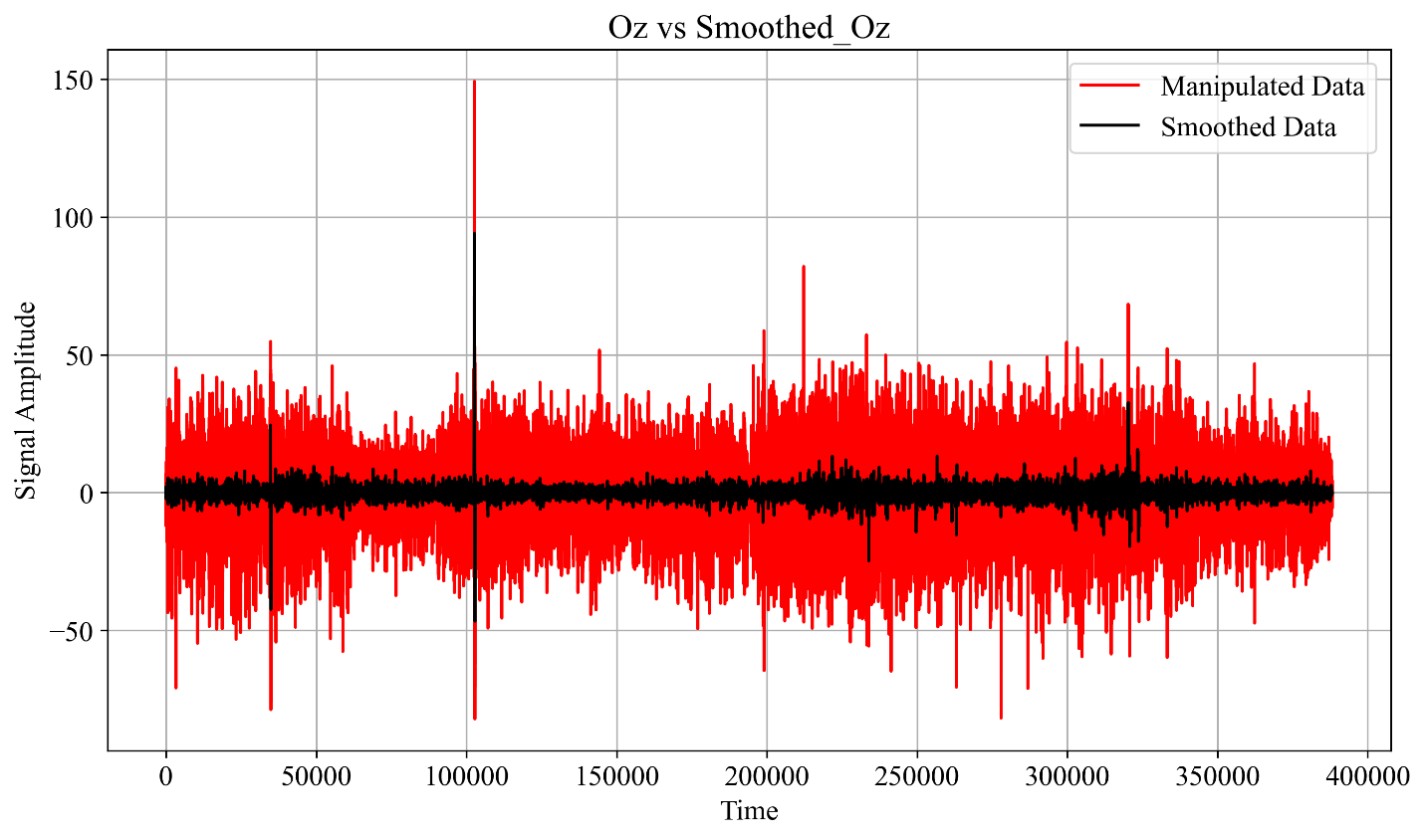




### **Figure:** Alert Data Visualization of unfiltered EEG signals (Oz) vs. filtered Oz signals.

Drowsy Data





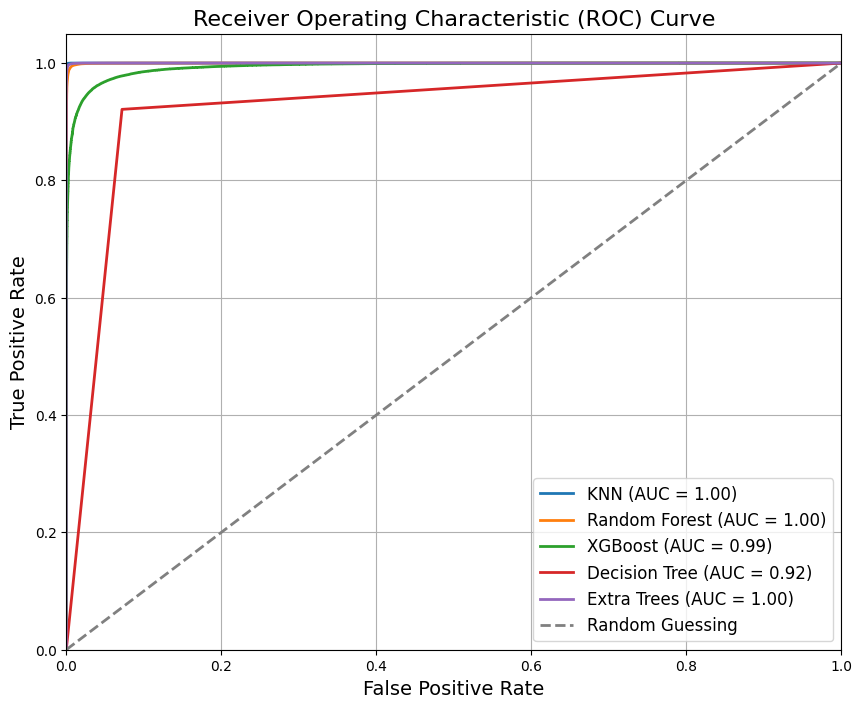
### **Figure:** Drowsy Data visualization of unfiltered EEG signals (Oz) vs. filtered Oz signals.

Machine Learning Evaluation with Savitzky-Golay Filter

(Polyorder: 3, Window Size: 203)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Accuracy | Precision | Recall | F1-score | AUC-ROC | Training Times (s) |
| KNN | 0.998693 | 0.999265 | 0.998119 | 0.998691 | 0.998693 | 0.07 |
| Random Forest | 0.992491 | 0.996493 | 0.988456 | 0.992458 | 0.992490 | 617.77 |
| XGBoost | 0.959991 | 0.971537 | 0.947716 | 0.959478 | 0.959986 | 6.72 |
| Decision Tree | 0.924586 | 0.927525 | 0.921084 | 0.924293 | 0.924585 | 57.02 |
| Extra Trees | 0.995943 | 0.998394 | 0.993481 | 0.995931 | 0.995942 | 154.29 |

### **Table:** Machine Learning Evaluation with Filtered Dataset

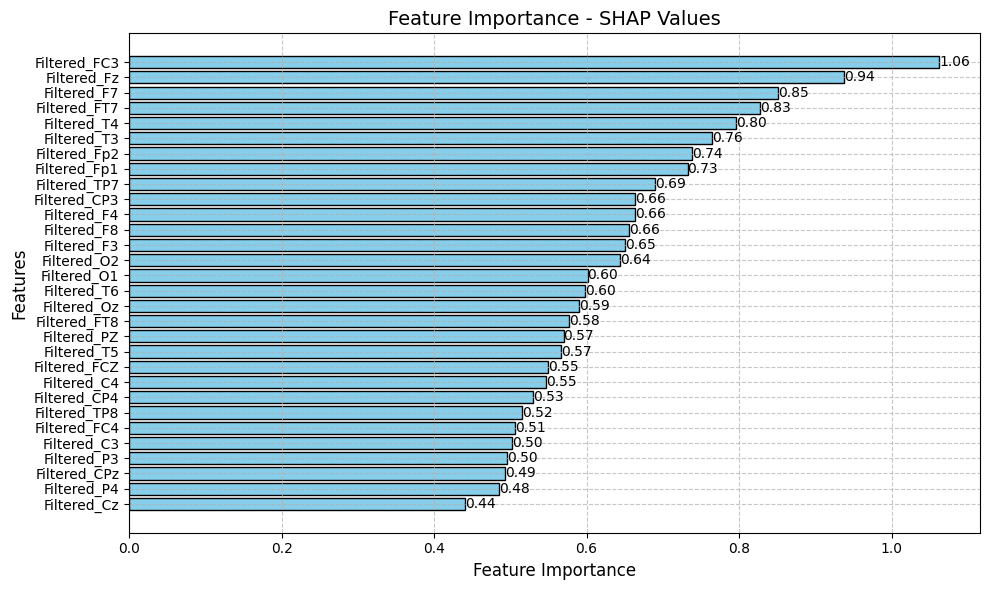


### **Figure:** ROC Curve for Machine Learning Results with Filtered Dataset.

# **Feature Selection - Shapley Approximation (shap)**

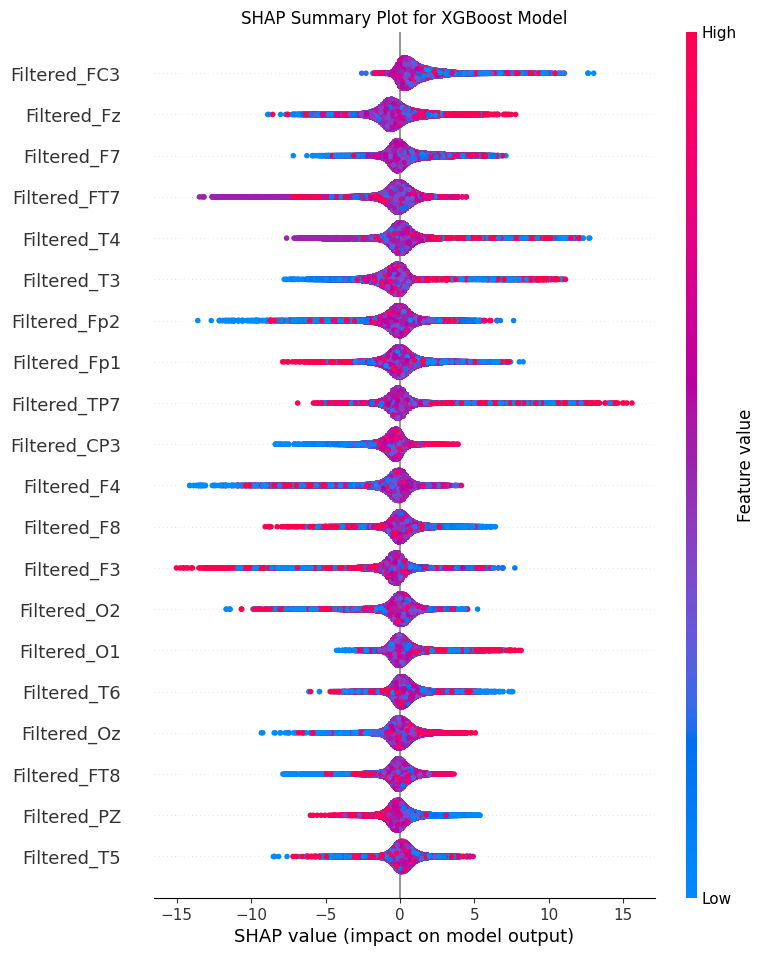
The dataset is split into training and testing sets, with 80% for training and 20% for testing. XGBoost classifier is trained on the training data. GPU acceleration is utilized for faster computation, and the number of trees is set to 5000. SHAP is employed for interpreting the XGBoost model predictions. SHAP values are calculated for the test dataset to understand the contribution of each feature to the model's output. Feature importance is determined based on the absolute mean SHAP values across all instances in the test dataset.

**Feature Importance Plot:** A horizontal bar plot is created to illustrate the importance of each feature, with annotations showing the numerical values of importance.

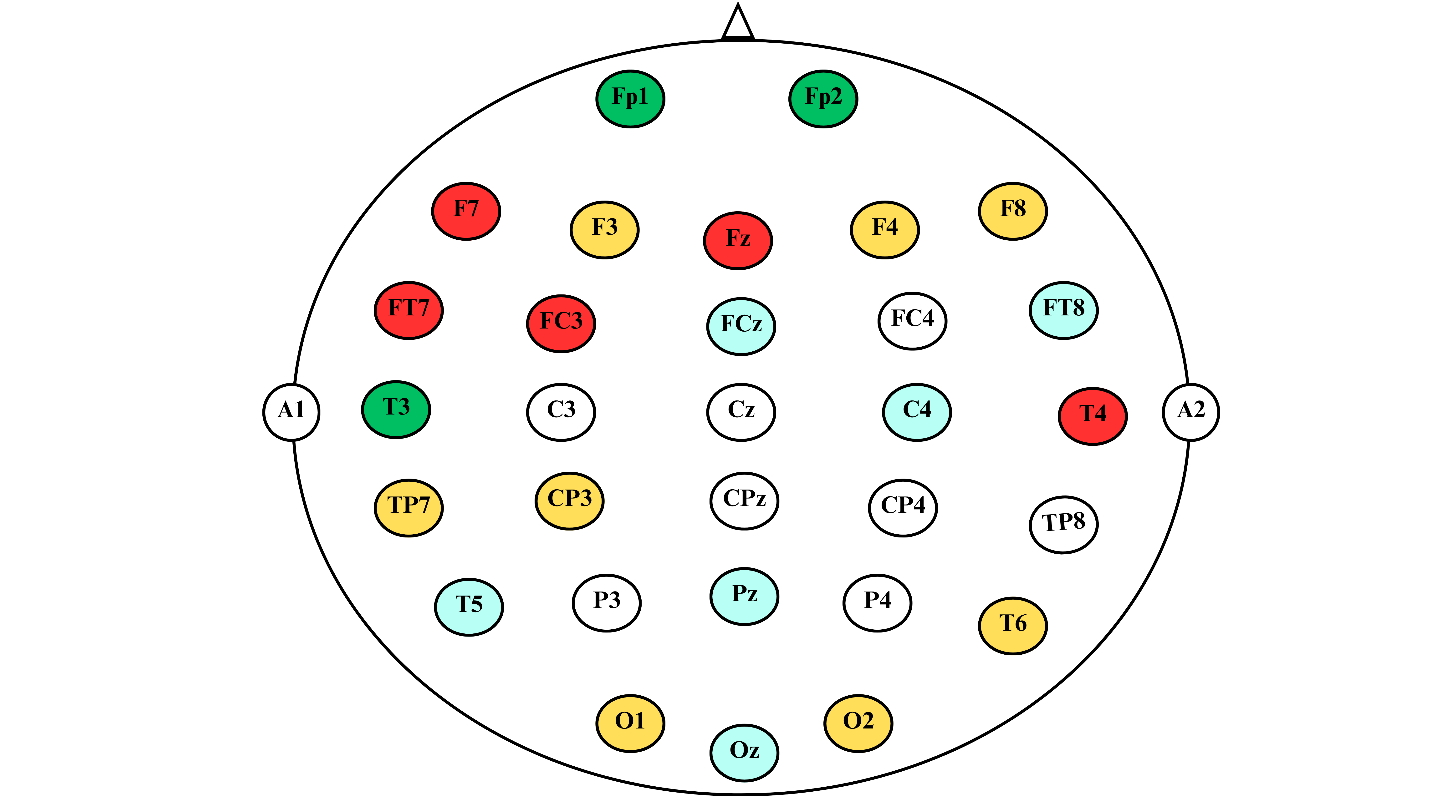


### **Figure:** Feature Importance SHAP Values

**SHAP Summary Plot:** A summary plot is generated to visualize the distribution of SHAP values for each feature.



### **Figure:** Summary Plot for Feature Impact Distribution.



### **Figure:** Electrode Layout on EEG Cap Sorted by Feature Importance Values

**Table of Feature Importance with Corresponding Denotations**

|  |  |  |  |
| --- | --- | --- | --- |
| Importance | Total Features | Features | Denotations |
| ≥ 0.80 | 5 | FC3, Fz, F7, FT7, T4 |  |
| ≥ 0.70 | 8 | FC3, Fz, F7, FT7, T4, T3, Fp2, Fp1 |  |
| ≥ 0.60 | 16 | FC3, Fz, F7, FT7, T4, T3, Fp2, Fp1, TP7, CP3, F4, F8, F3, O2, O1, T6 |  |
| ≥ 0.55 | 22 | FC3, Fz, F7, FT7, T4, T3, Fp2, Fp1, TP7, CP3, F4, F8, F3, O2, O1, T6, Oz, FT8, PZ, T5, FCZ, C4. |  |
| ≥ 0.40 | 30 | FC3, Fz, F7, FT7, T4, T3, Fp2, Fp1, TP7, CP3, F4, F8, F3, O2, O1, T6, Oz, FT8, PZ, T5, FCZ, C4, CP4, TP8, FC4, C3, P3, CPz, P4, Cz |  |

### **Table:** Feature Importance with Corresponding Denotations

## References

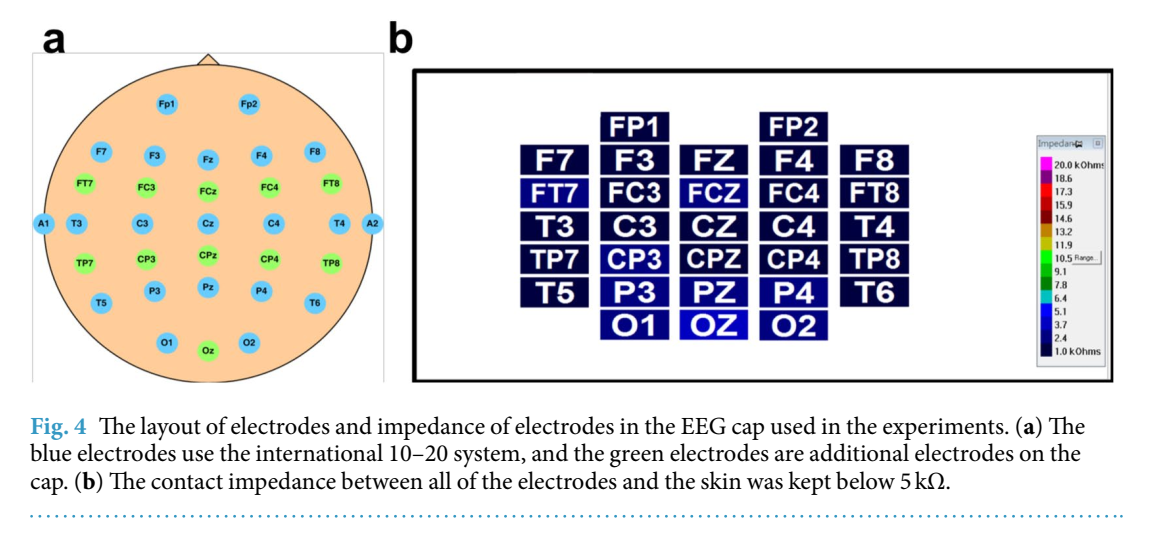


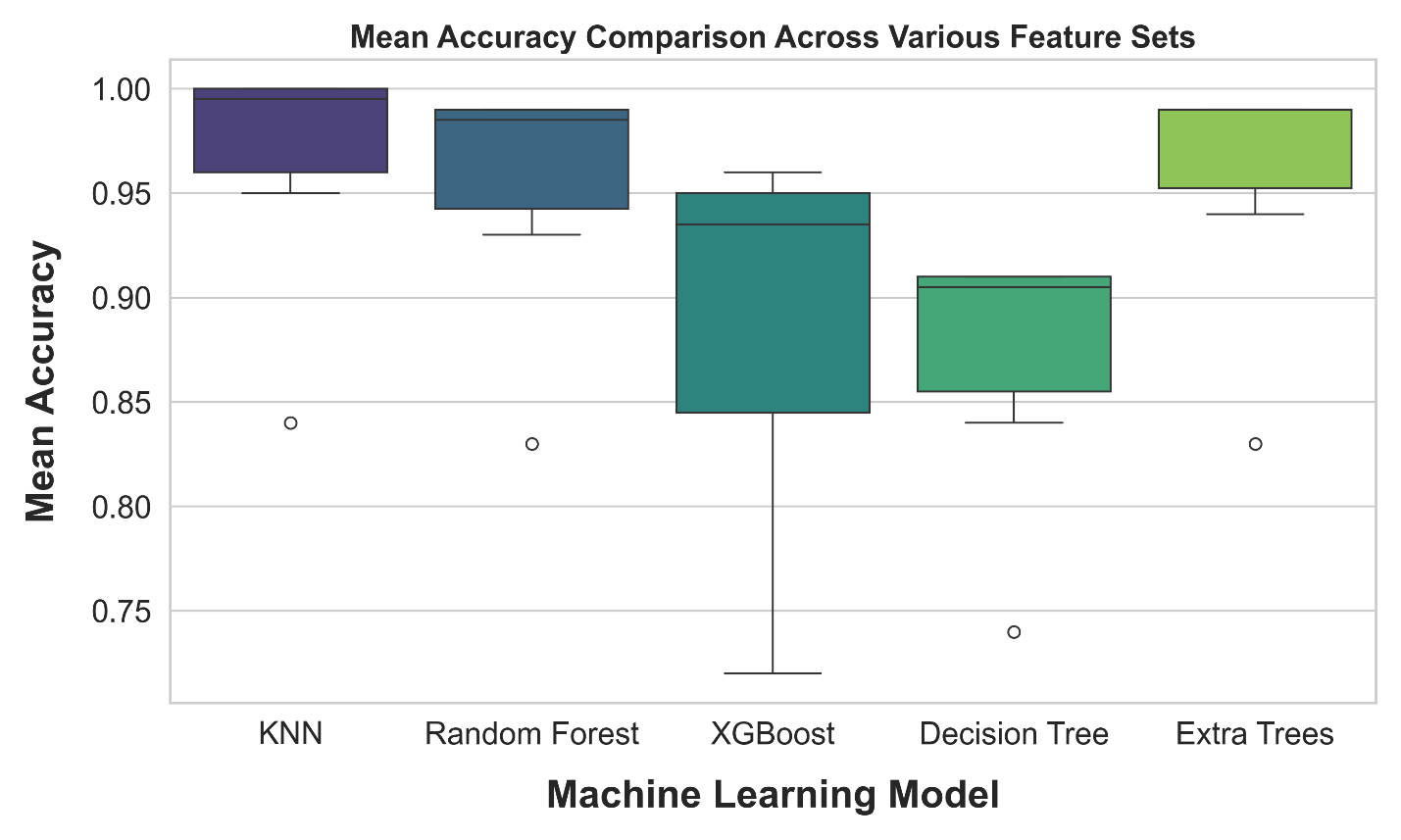
Image Source: <https://www.nature.com/articles/s41597-019-0027-4>

# **Machine Learning Results**

## Mean Accuracy

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of Features | KNN | Random Forest | XGBoost | Decision Tree | Extra Trees |
| 5 Features | 0.84 | 0.83 | 0.72 | 0.74 | 0.83 |
| 8 Features | 0.95 | 0.93 | 0.82 | 0.84 | 0.94 |
| 16 Features | 0.99 | 0.98 | 0.92 | 0.90 | 0.99 |
| 22 Features | 1.00 | 0.99 | 0.95 | 0.91 | 0.99 |
| 27 Features | 1.00 | 0.99 | 0.95 | 0.91 | 0.99 |
| 30 Features | 1.00 | 0.99 | 0.96 | 0.91 | 0.99 |

### **Table:** Mean accuracy comparison across various feature sets.

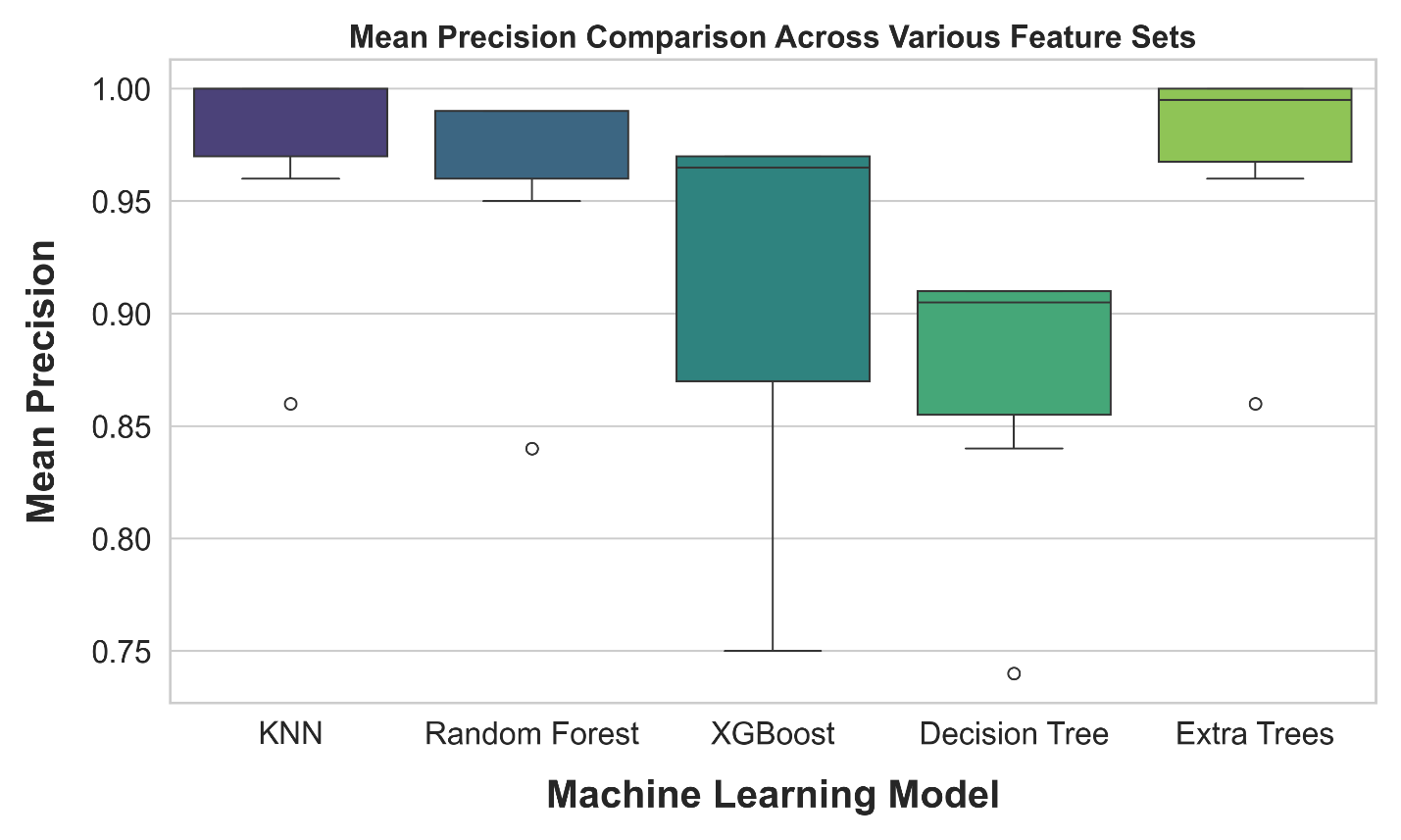


### **Figure:** Mean accuracy comparison across various feature sets.

## Mean Precision

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of Features | KNN | Random Forest | XGBoost | Decision Tree | Extra Trees |
| 5 Features | 0.86 | 0.84 | 0.75 | 0.74 | 0.86 |
| 8 Features | 0.96 | 0.95 | 0.84 | 0.84 | 0.96 |
| 16 Features | 1.00 | 0.99 | 0.97 | 0.90 | 0.99 |
| 22 Features | 1.00 | 0.99 | 0.96 | 0.91 | 1.00 |
| 27 Features | 1.00 | 0.99 | 0.97 | 0.91 | 1.00 |
| 30 Features | 1.00 | 0.99 | 0.97 | 0.91 | 1.00 |

### **Table:** Mean precision comparison across various feature sets.

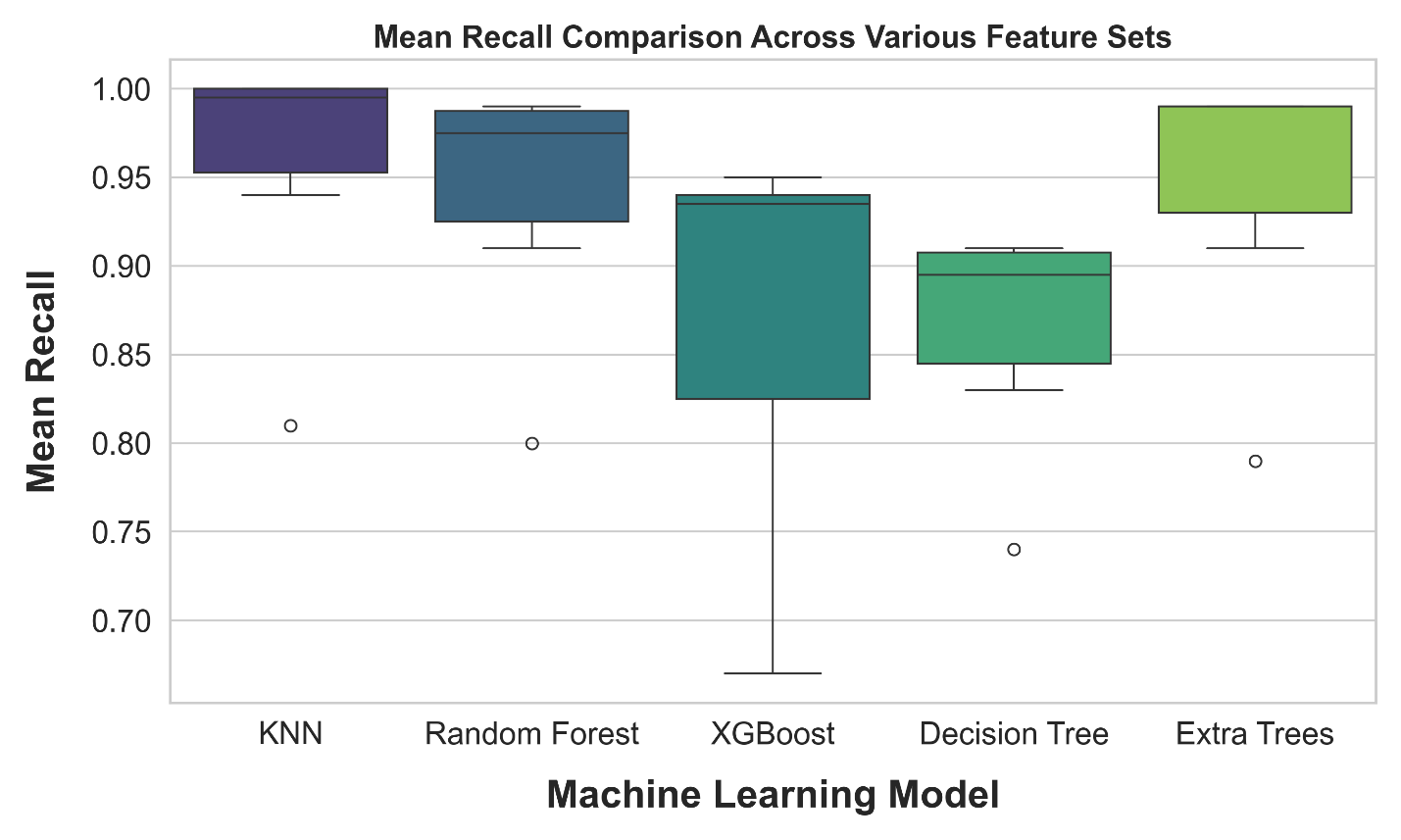


### **Figure:** Mean precision comparison across various feature sets.

## Mean Recall

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of Features | KNN | Random Forest | XGBoost | Decision Tree | Extra Trees |
| 5 Features | 0.81 | 0.80 | 0.67 | 0.74 | 0.79 |
| 8 Features | 0.94 | 0.91 | 0.79 | 0.83 | 0.91 |
| 16 Features | 1.00 | 0.97 | 0.94 | 0.89 | 0.99 |
| 22 Features | 0.99 | 0.98 | 0.93 | 0.90 | 0.99 |
| 27 Features | 1.00 | 0.99 | 0.94 | 0.91 | 0.99 |
| 30 Features | 1.00 | 0.99 | 0.95 | 0.91 | 0.99 |

### **Table:** Mean recall comparison across various feature sets.

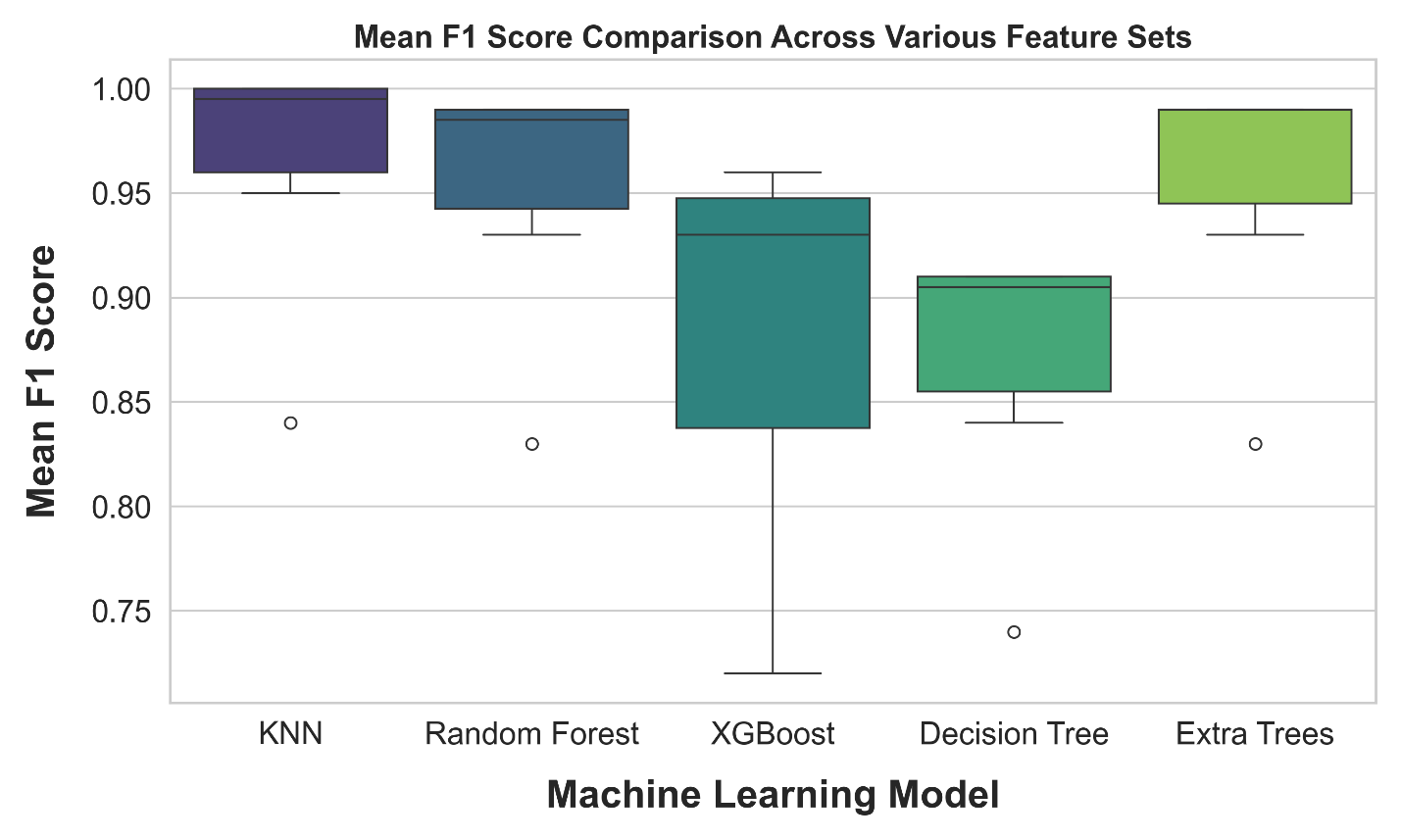


### **Figure:** Mean recall comparison across various feature sets.

## Mean F1 Score

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of Features | KNN | Random Forest | XGBoost | Decision Tree | Extra Trees |
| 5 Features | 0.84 | 0.83 | 0.72 | 0.74 | 0.83 |
| 8 Features | 0.95 | 0.93 | 0.81 | 0.84 | 0.93 |
| 16 Features | 0.99 | 0.98 | 0.92 | 0.90 | 0.99 |
| 22 Features | 1.00 | 0.99 | 0.94 | 0.91 | 0.99 |
| 27 Features | 1.00 | 0.99 | 0.95 | 0.91 | 0.99 |
| 30 Features | 1.00 | 0.99 | 0.96 | 0.91 | 0.99 |

### **Table:** Mean F1 Score comparison across various feature sets.

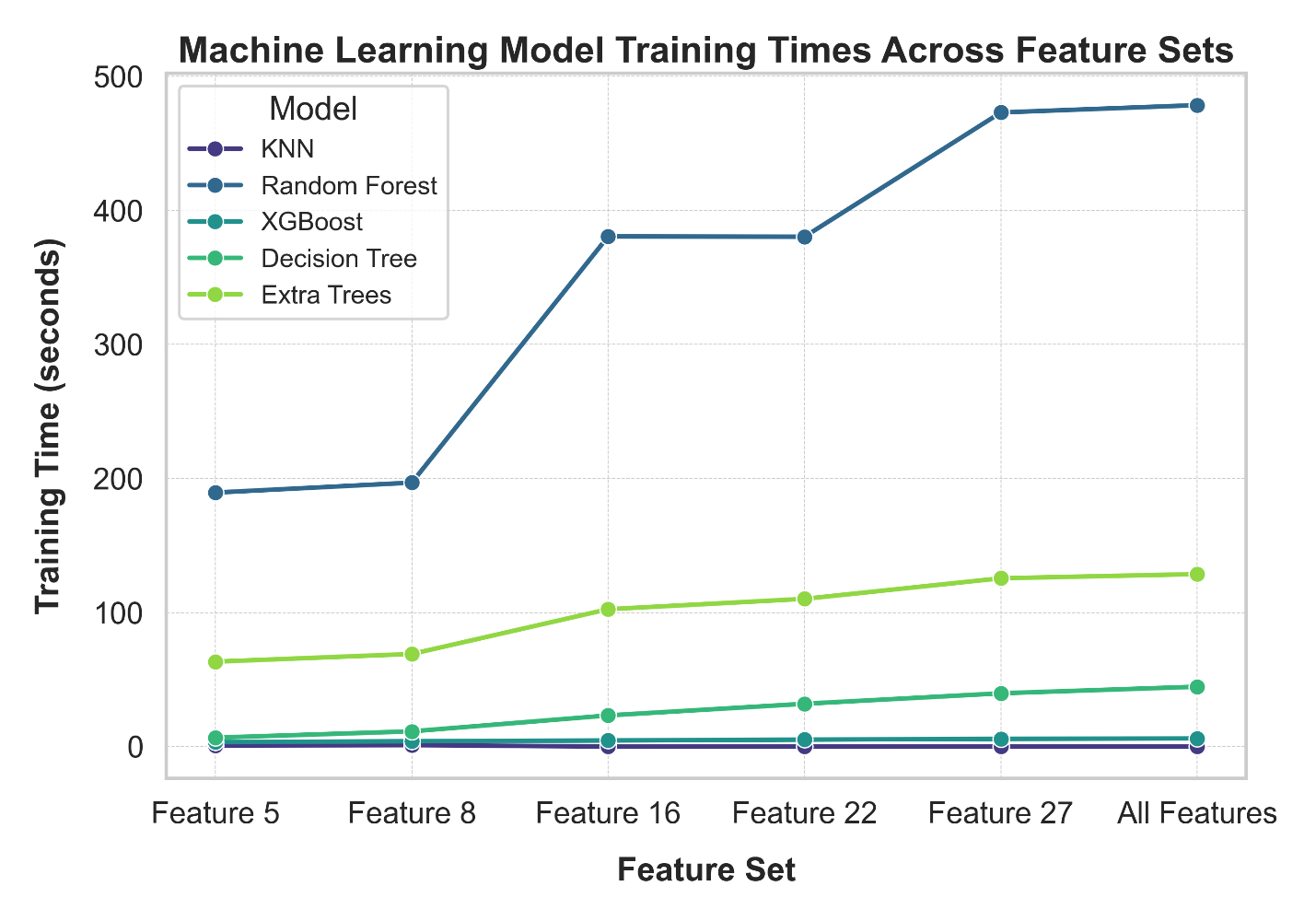


### **Figure:** Mean F1 Score comparison across various feature sets.

## Training Time of Machine Learning Model

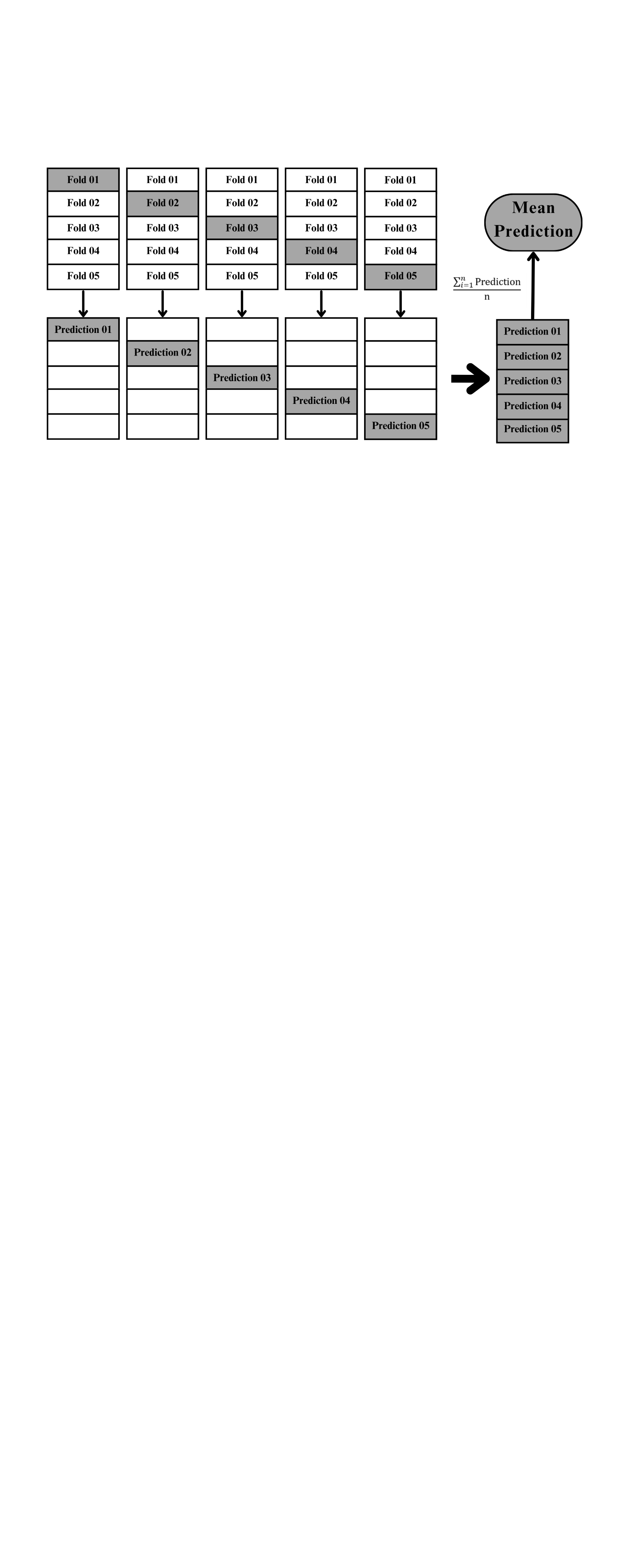
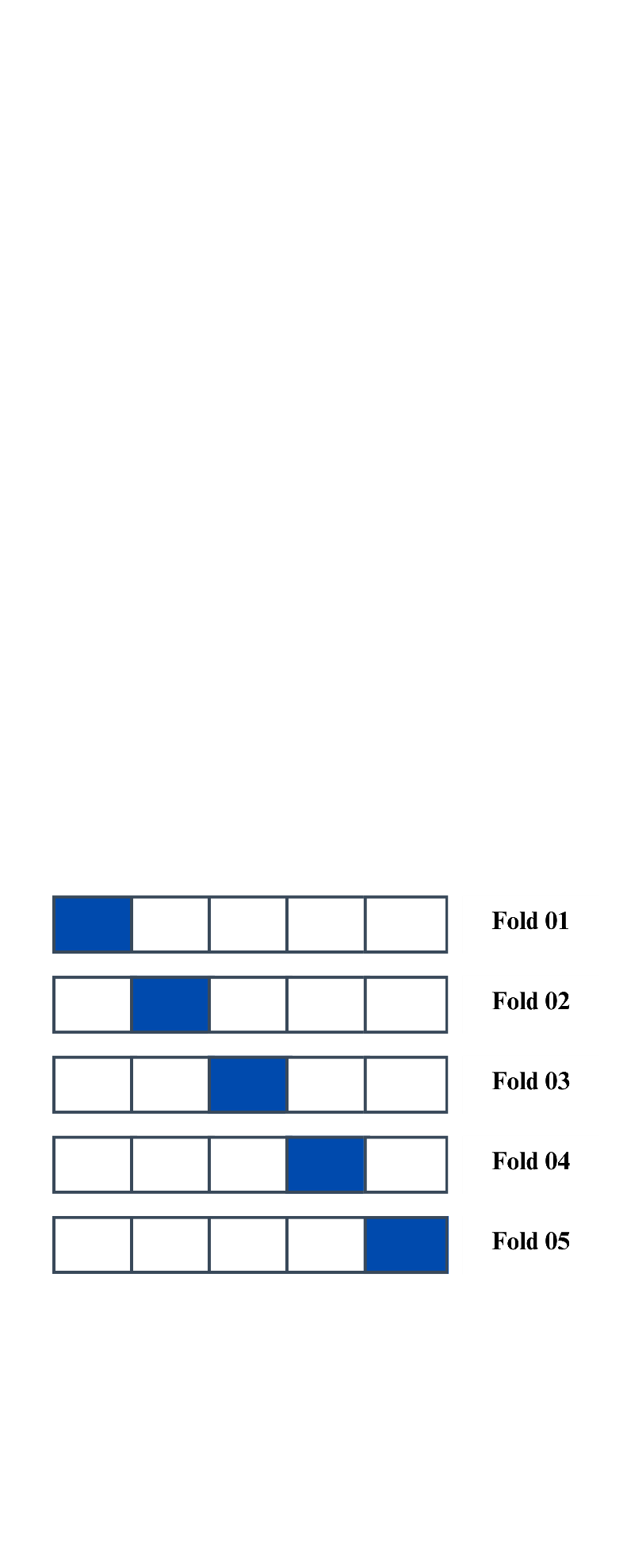
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of Features | KNN | Random Forest | XGBoost | Decision Tree | Extra Trees |
| 5 Features | 0.79 | 189.50 | 3.36 | 6.71 | 63.40 |
| 8 Features | 1.11 | 196.92 | 4.02 | 11.41 | 69.11 |
| 16 Features | 0.04 | 380.56 | 4.59 | 23.29 | 102.52 |
| 22 Features | 0.05 | 380.34 | 5.22 | 31.97 | 110.24 |
| 27 Features | 0.05 | 473.02 | 5.79 | 39.78 | 125.60 |
| 30 Features | 0.05 | 478.50 | 6.10 | 44.69 | 128.68 |

### **Table:** Machine learning model training time across various feature sets



### **Figure:** Machine learning model training time across various feature sets

## Cross Validation



### **Figure:** Cross Validation Workflow

# **Deep Learning**

## **Artificial Neural Network Architecture**

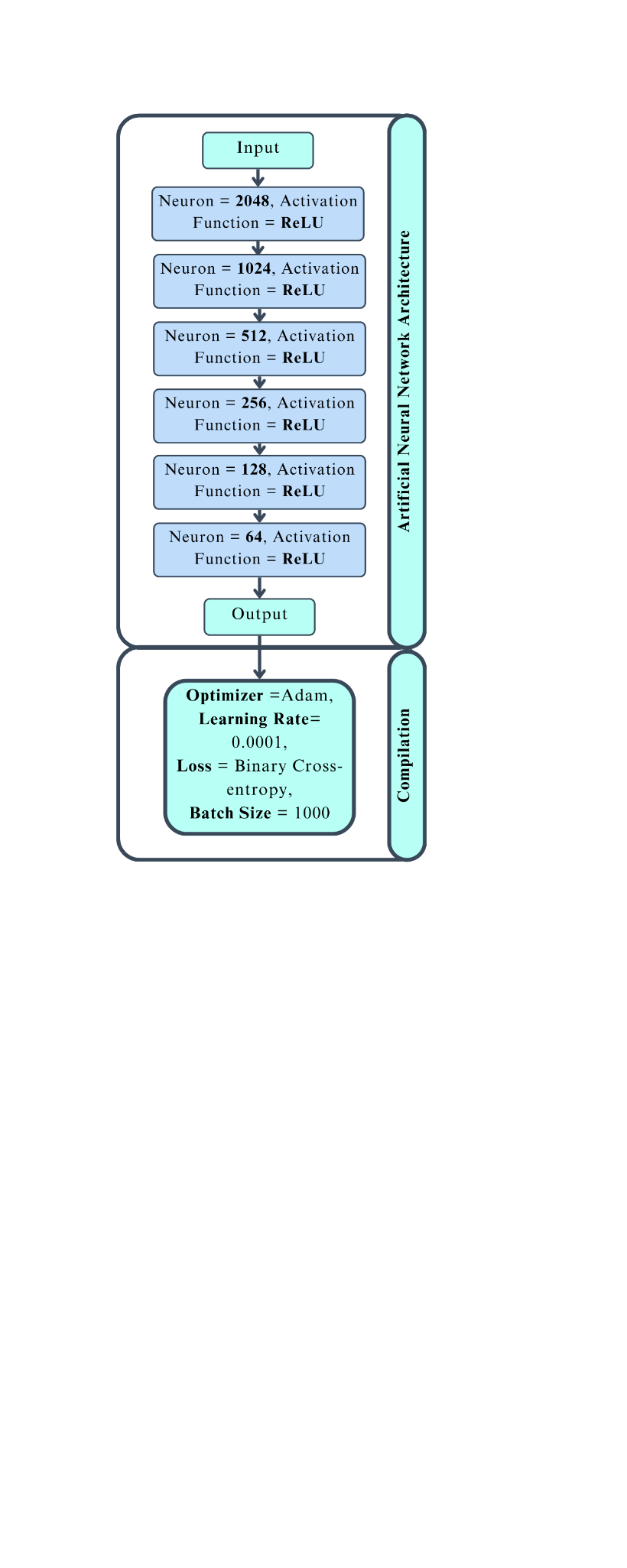
## The input and hidden layers

The ANN architecture contains eight layers: an input layer, six hidden layers, and an output layer. The input layer was defined based on the shape of the training dataset. The six hidden layers contain 2048, 1024, 512, 256, 128, and 64 neurons each. We set the activation function for each of the six hidden layers as "ReLU." Finally, we add a single neuron with a sigmoid activation function at the output layer.

## Compilation and Training

We compile the model using the Adam optimizer, with a learning rate 0.0001. The loss function we use is binary cross-entropy ('binary\_crossentropy'). Accuracy was the primary evaluation metric during training.

We set the batch size to 1024 and the epoch to 500. However, we implemented early stopping with a patience setting of 30 epochs to avoid overfitting. Therefore, the model will stop when the validation loss stops decreasing.



### **Figure:** Artificial Neural Network Architecture

## **Long short-term memory Architecture**

## The input and hidden layers

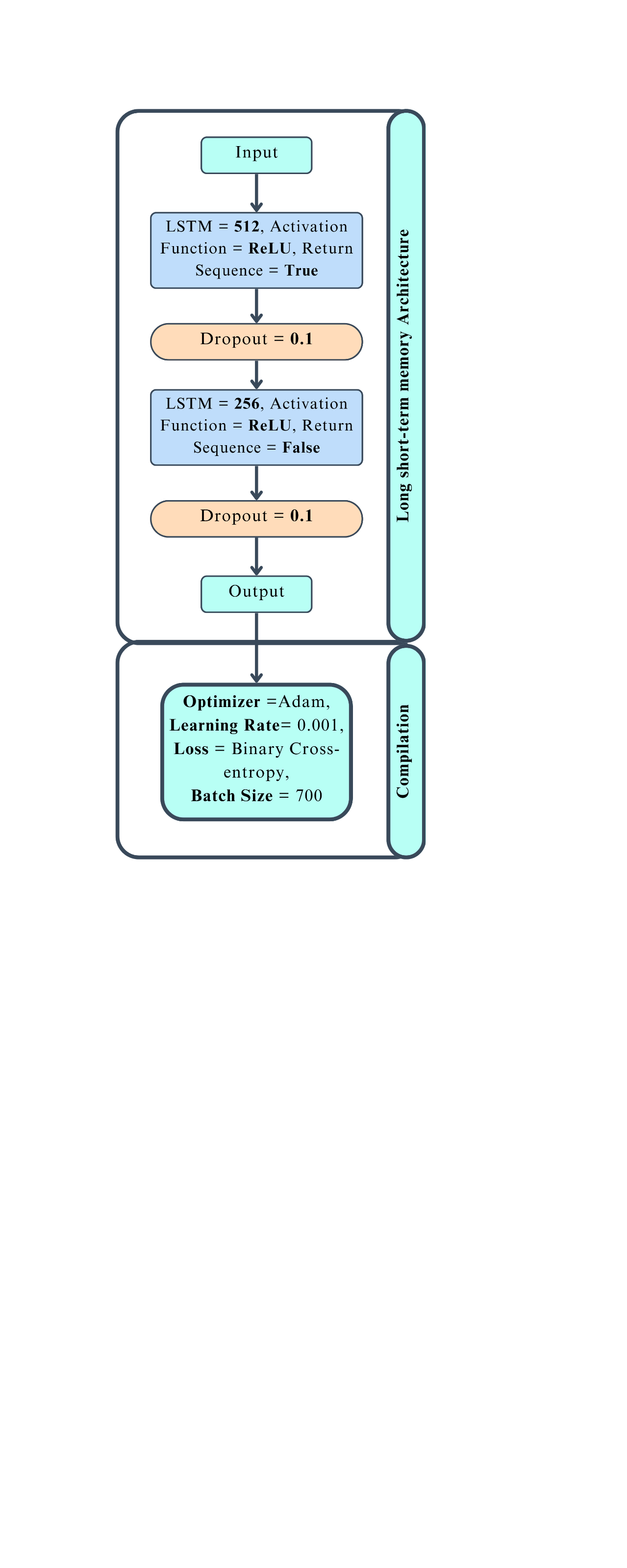
The first layer of our LSTM architecture has 512 units and uses the ReLU activation function. It also has return\_sequences=True. Next, we introduce a dropout layer with a 0.1 dropout rate.

The second layer has 256 units and uses the ReLU activation function. Return\_sequences=False generates the final state in the sequence. Another LSTM layer with a 0.1 dropout rate follows the second one.

Finally, the output layer included a dense layer with a single neuron and a sigmoid activation function.

## Compilation and Training

As for compilation, we used the Adam optimizer with a learning rate of 0.001. We chose Adam because of his adaptive learning rate properties. The loss function we use is binary cross-entropy ('binary\_crossentropy'). We evaluate the model based on its accuracy. We set the batch size to 700, trained the model for a maximum of 500 epochs, and set the early stopping at 30.



### **Figure:** Long short-term memory Architecture

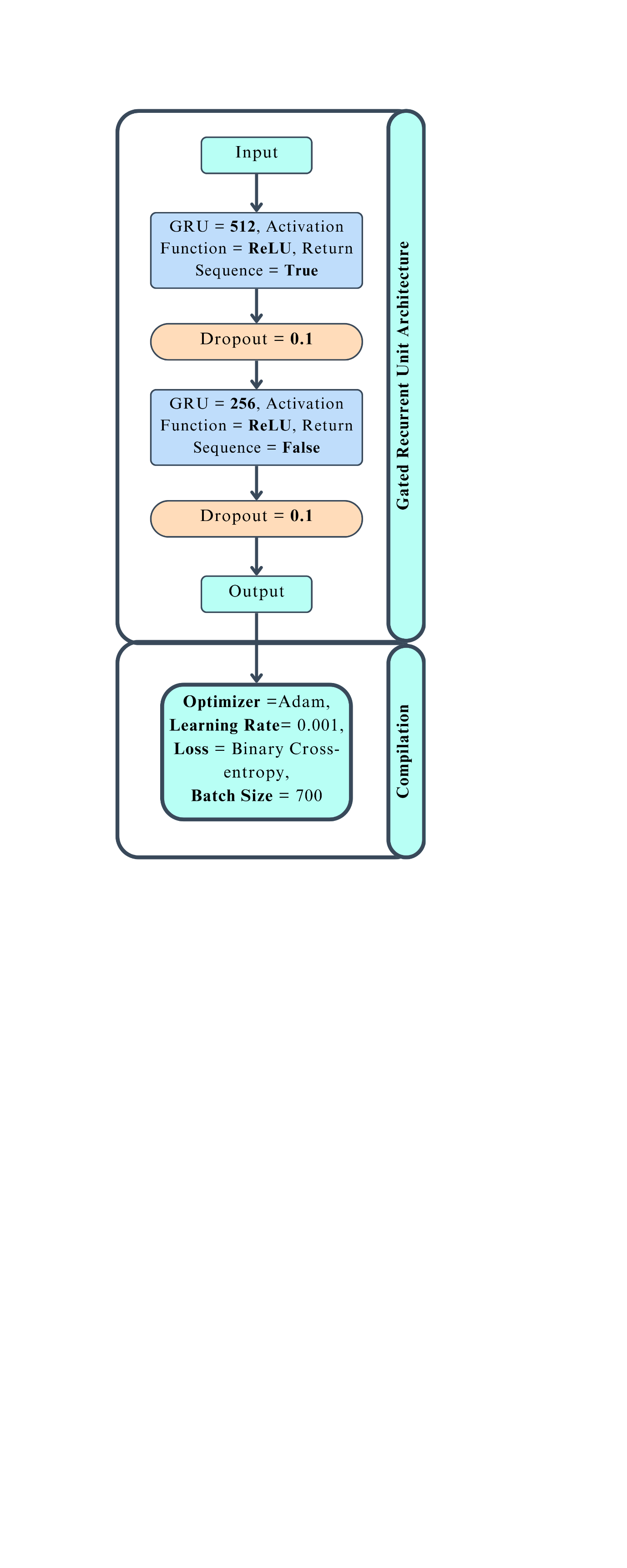
## **Gated Recurrent Unit Networks Architecture**

## The input and hidden layers

The architecture includes two GRU layers: the first has 512 units using ReLU activation and return\_sequences=True, followed by a Dropout layer with a 0.1 rate to prevent overfitting; the second has 256 units with ReLU activation and outputs only the last state, followed by another Dropout layer. We use a dense output layer with a single neuron and sigmoid activation for binary classification.

### Compilation and Training

We configure the model using the Adam optimizer, setting the learning rate to 0.0001. We defined the loss function as binary cross-entropy ('binary\_crossentropy'). Training involves a batch size of 700, up to 500 epochs, and early stopping with 30-epoch patience to prevent overfitting. Throughout the training process, we use the accuracy metric for evaluation.



### **Figure:** Gated Recurrent Unit Networks Architecture

## **Ensemble Learning**

## Ensemble Learning Architecture

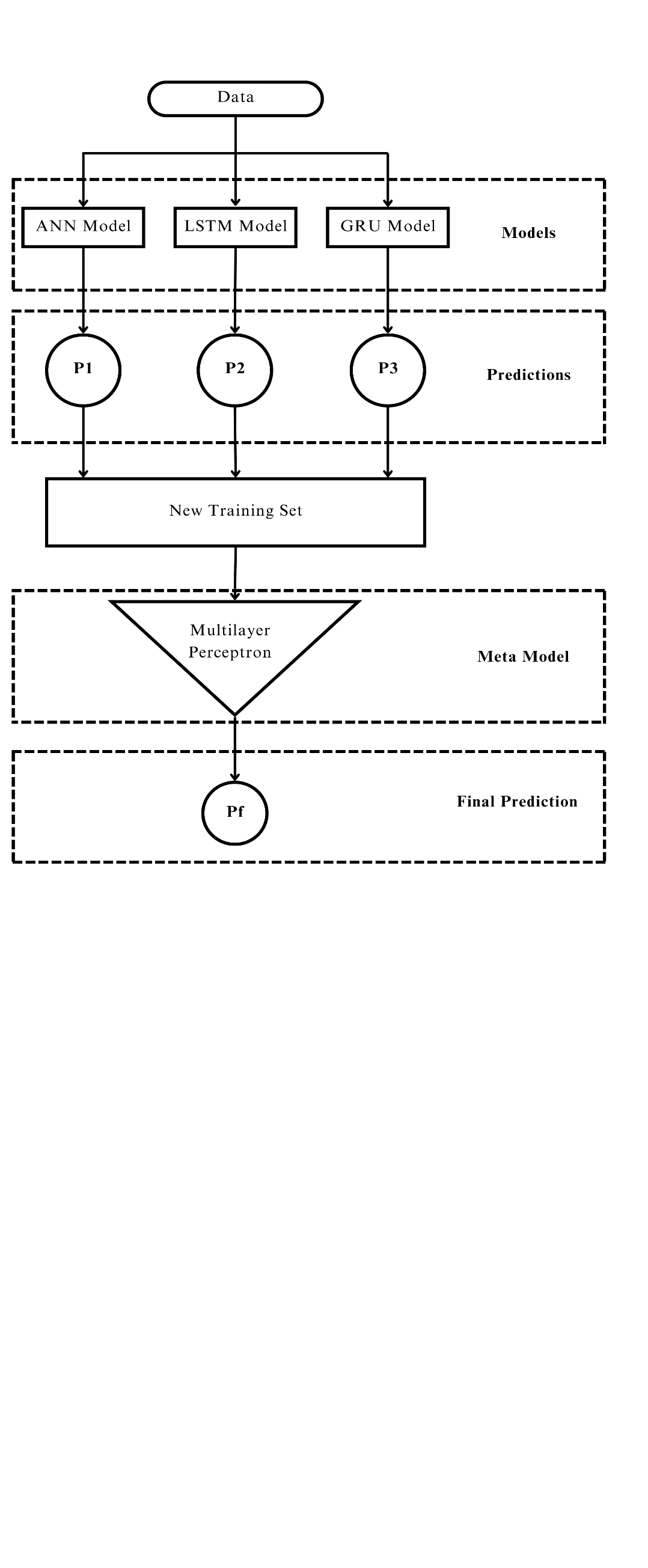
Stacking ensemble learning integrates the strengths of multiple base learners to improve classification performance. Here, we use predictions from three different neural network models: Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), and then make the final predictions.

We generate predictions from these three models separately for the training and testing datasets. We then stack these predictions horizontally to create new feature sets. This stacking step combines the outputs from different models into a unified input for the meta-model.

### Meta-Model Training

We used a Multi-Layer Perceptron (MLP) classifier for the meta-model. We configure this MLP classifier with 100 hidden neurons and a maximum of 1000 iterations for convergence. The training process involves fitting the MLP to the stacked predictions from the base models, using the true labels as the dependent variable. We also record the total training time for evaluation purposes.

After training, we make predictions on the test set and record the time it takes to make these predictions. Finally, we evaluate the ensemble model performance based on accuracy, precision, recall, F1-score, and AUC-ROC score.

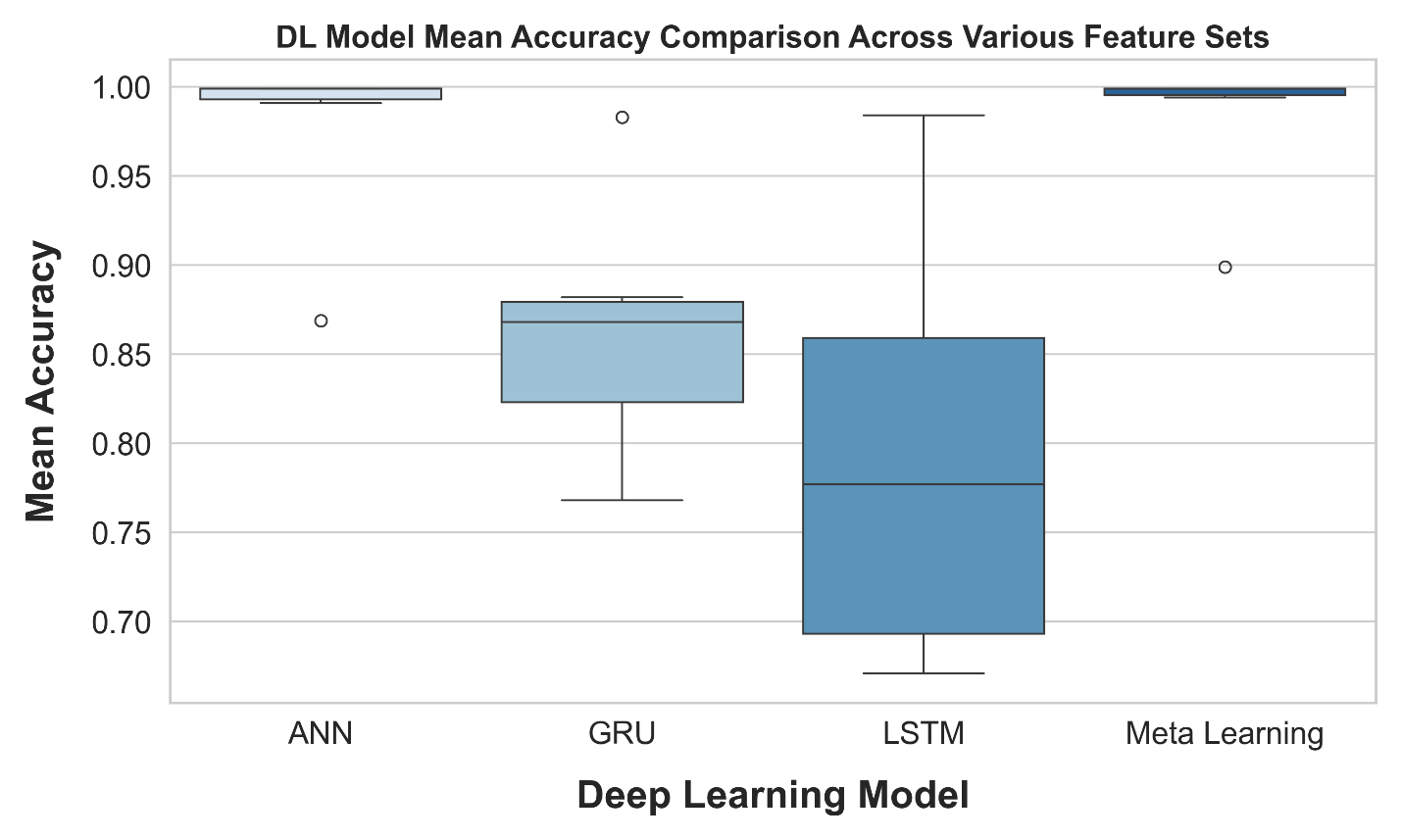


### **Figure:** Ensemble Learning Architecture

# Deep Learning Results

## Mean Accuracy

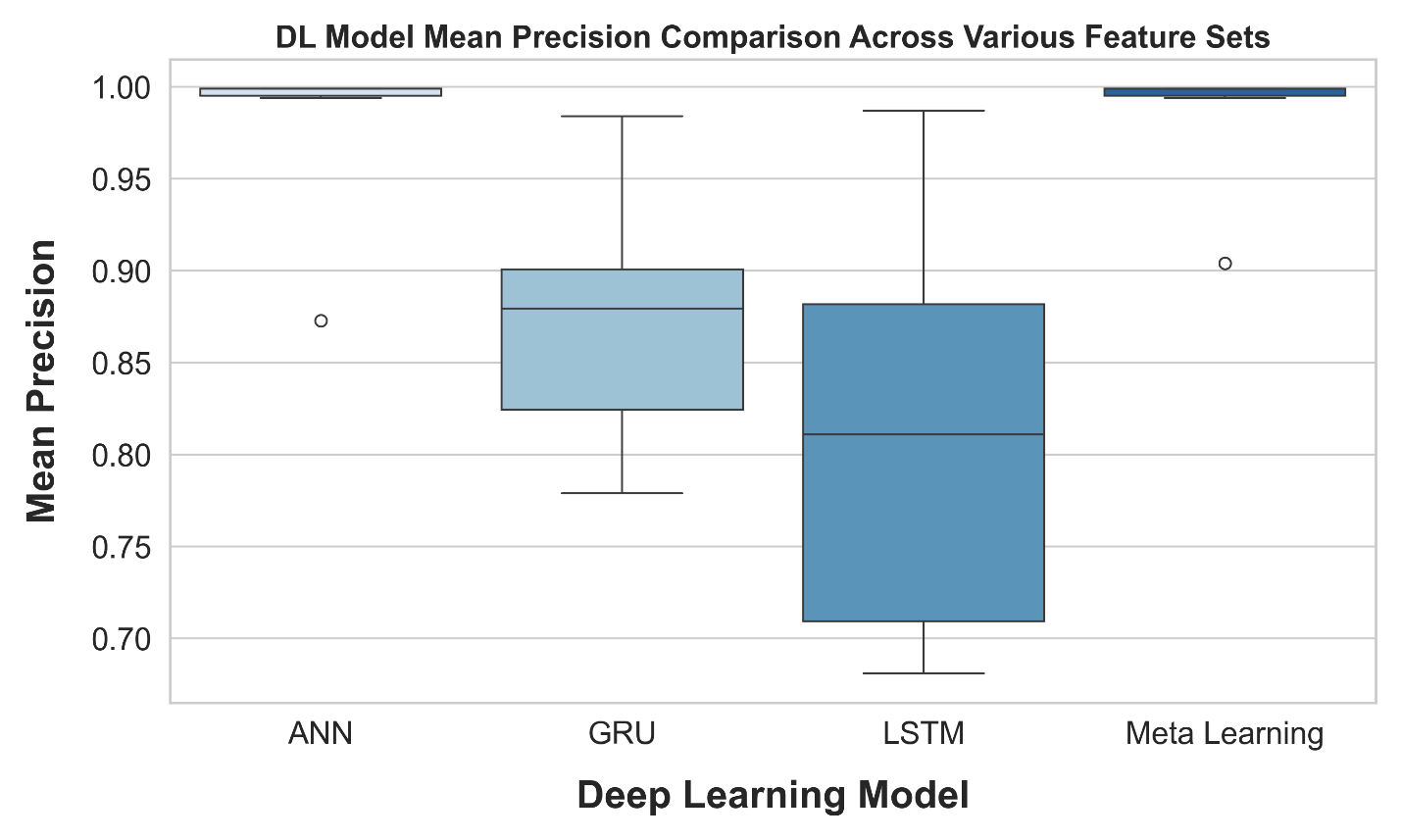
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of Features | ANN | GRU | LSTM | Meta Learning |
| 5 Features | 0.869 | 0.882 | 0.878 | 0.899 |
| 8 Features | 0.991 | 0.983 | 0.984 | 0.994 |
| 16 Features | 0.999 | 0.871 | 0.751 | 0.999 |
| 22 Features | 0.999 | 0.768 | 0.803 | 0.999 |
| 27 Features | 0.999 | 0.865 | 0.671 | 0.999 |
| 30 Features | 0.999 | 0.809 | 0.674 | 0.999 |



### **Figure:** Box Plot of Mean Accuracy Across Feature Sets in Deep Learning Models

## Mean Precision

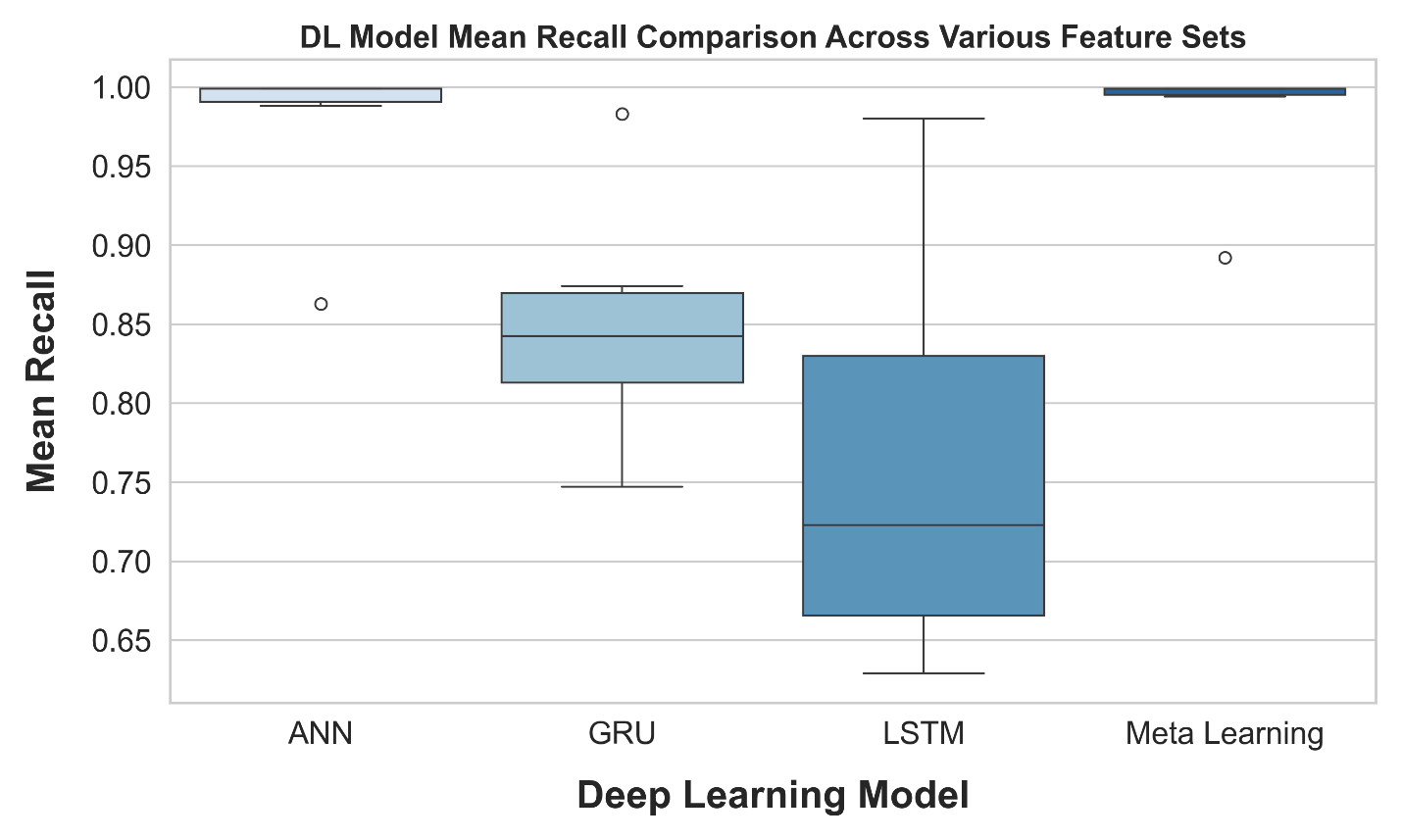
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of Features | ANN | GRU | LSTM | Meta Learning |
| 5 Features | 0.873 | 0.888 | 0.894 | 0.904 |
| 8 Features | 0.994 | 0.984 | 0.987 | 0.994 |
| 16 Features | 0.999 | 0.905 | 0.777 | 0.999 |
| 22 Features | 0.999 | 0.779 | 0.845 | 0.999 |
| 27 Features | 0.999 | 0.871 | 0.687 | 0.999 |
| 30 Features | 0.999 | 0.809 | 0.681 | 0.999 |



### **Figure:** Box Plot of Mean Precision Across Feature Sets in Deep Learning Models

## Mean Recall

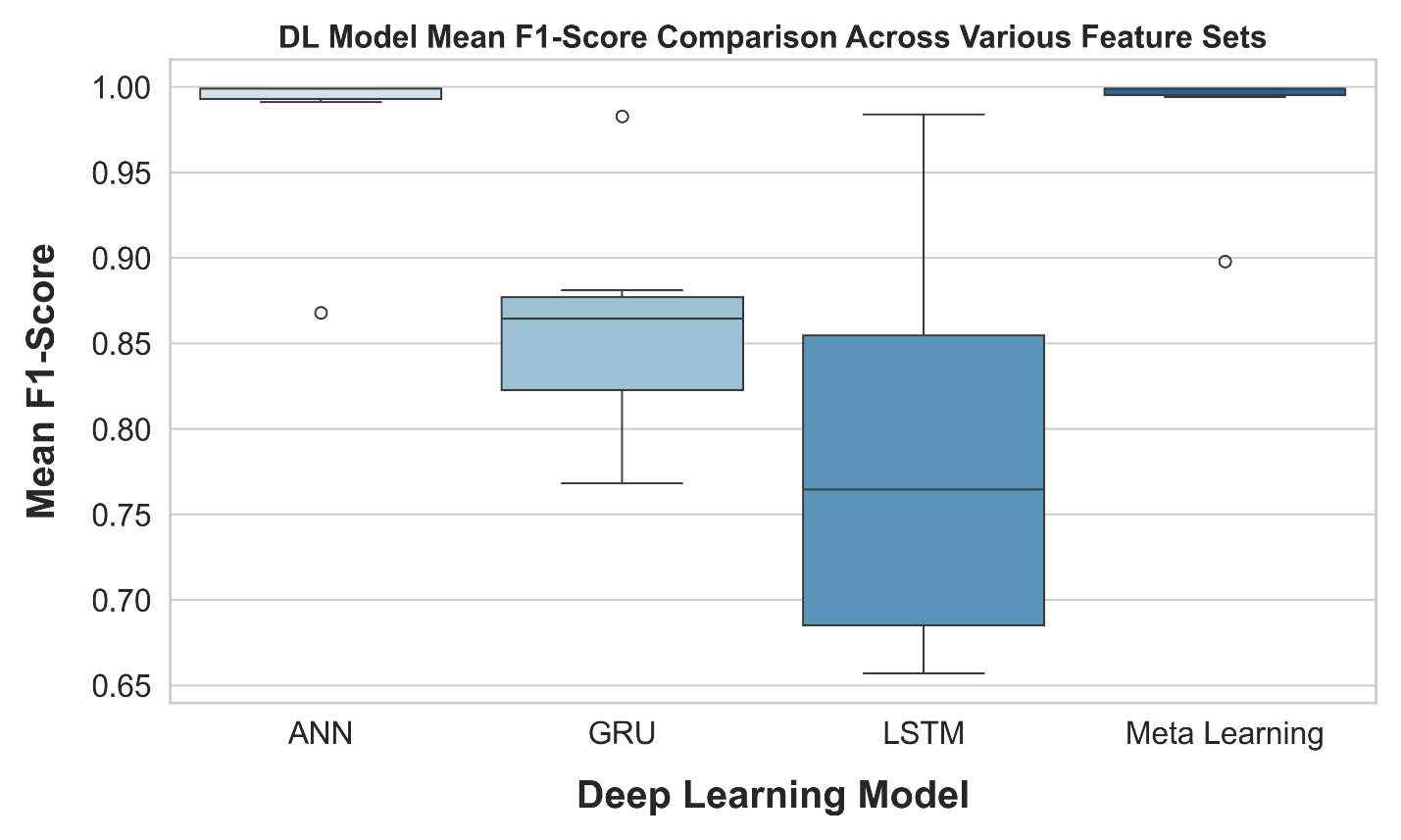
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of Features | ANN | GRU | LSTM | Meta Learning |
| 5 Features | 0.863 | 0.874 | 0.859 | 0.892 |
| 8 Features | 0.988 | 0.983 | 0.980 | 0.994 |
| 16 Features | 0.999 | 0.828 | 0.704 | 0.999 |
| 22 Features | 0.999 | 0.747 | 0.742 | 0.999 |
| 27 Features | 0.999 | 0.857 | 0.629 | 0.999 |
| 30 Features | 0.999 | 0.808 | 0.653 | 0.999 |



### **Figure:** Box Plot of Mean Recall Across Feature Sets in Deep Learning Models

## Mean F1-Score

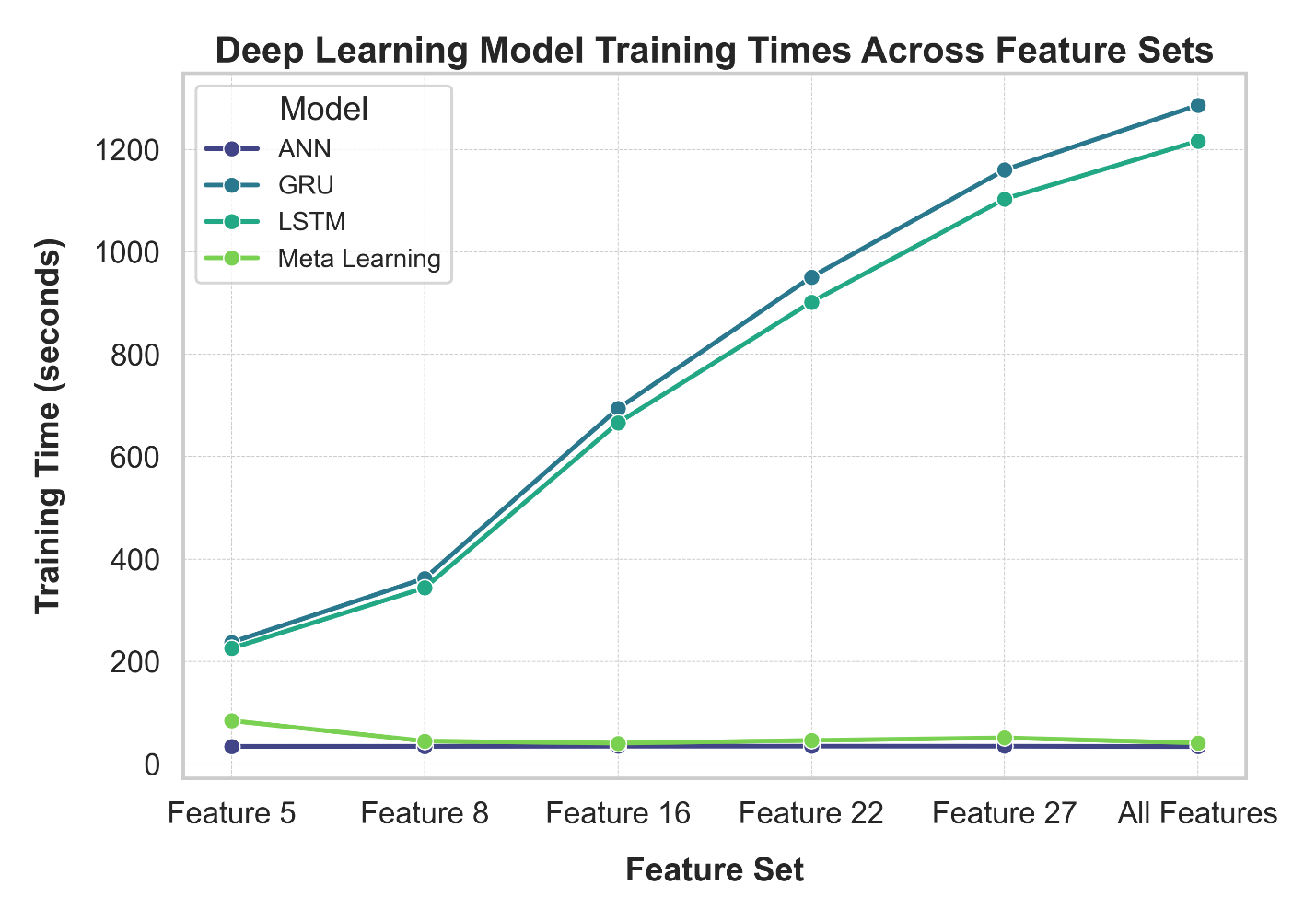
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of Features | ANN | GRU | LSTM | Meta Learning |
| 5 Features | 0.868 | 0.881 | 0.876 | 0.898 |
| 8 Features | 0.991 | 0.983 | 0.984 | 0.994 |
| 16 Features | 0.999 | 0.865 | 0.739 | 0.999 |
| 22 Features | 0.999 | 0.768 | 0.790 | 0.999 |
| 27 Features | 0.999 | 0.864 | 0.657 | 0.999 |
| 30 Features | 0.999 | 0.809 | 0.667 | 0.999 |



### **Figure:** Box Plot of Mean F1 Score Across Feature Sets in Deep Learning Models

## Training Time (in seconds)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of Features | ANN | GRU | LSTM | Meta Learning |
| 5 Features | 34.34 | 237.50 | 225.93 | 84.64 |
| 8 Features | 34.47 | 362.58 | 344.09 | 44.75 |
| 16 Features | 34.70 | 694.13 | 666.29 | 40.59 |
| 22 Features | 34.82 | 950.87 | 902.08 | 46.06 |
| 27 Features | 34.70 | 1,160.16 | 1103.39 | 51.05 |
| 30 Features | 34.00 | 1,286.49 | 1216.40 | 41.10 |

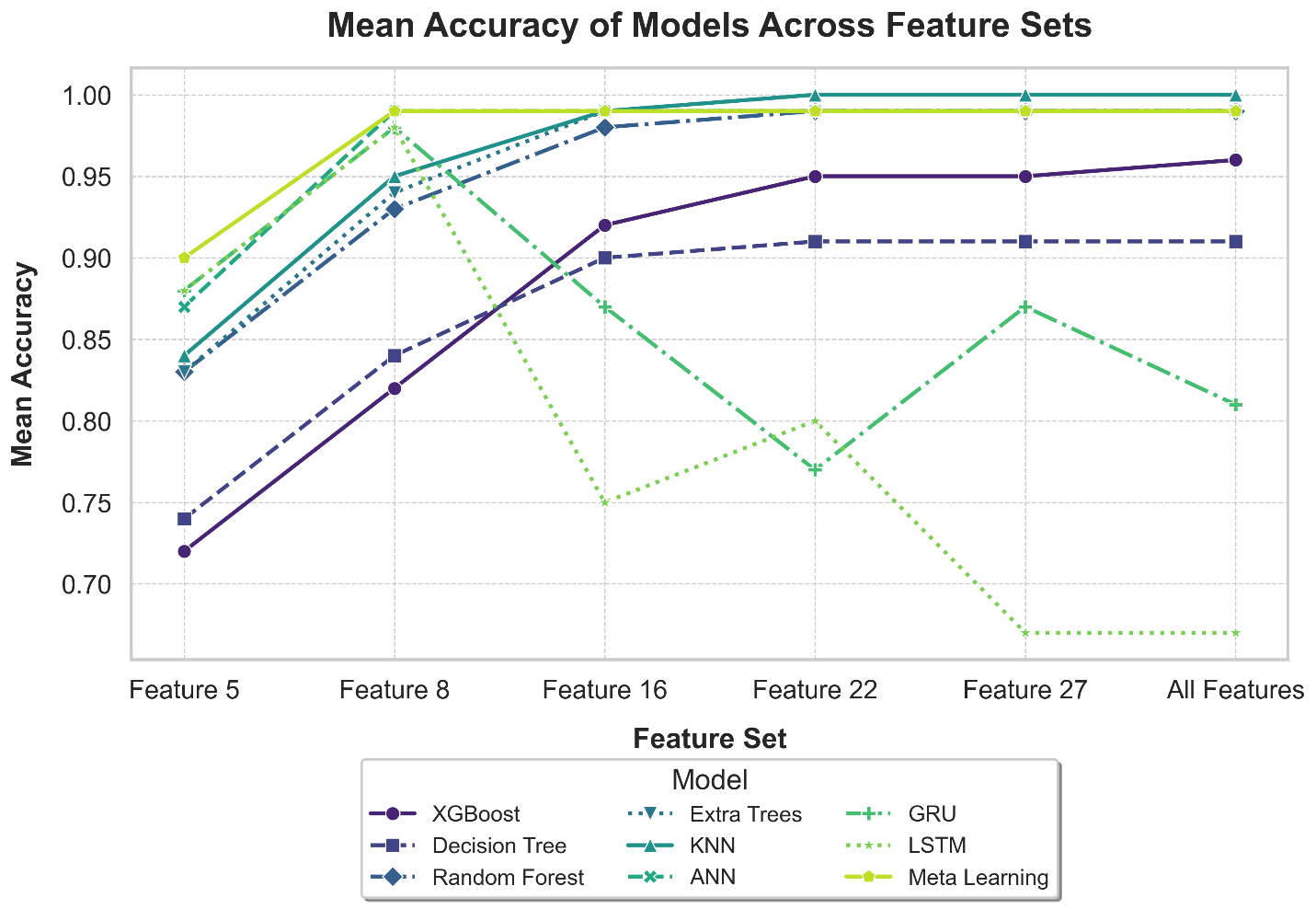


### **Figure:** Comparison of Deep Learning Model Training Times Across Feature Sets.

# **Overall Performance Comparison**

## Mean Accuracy

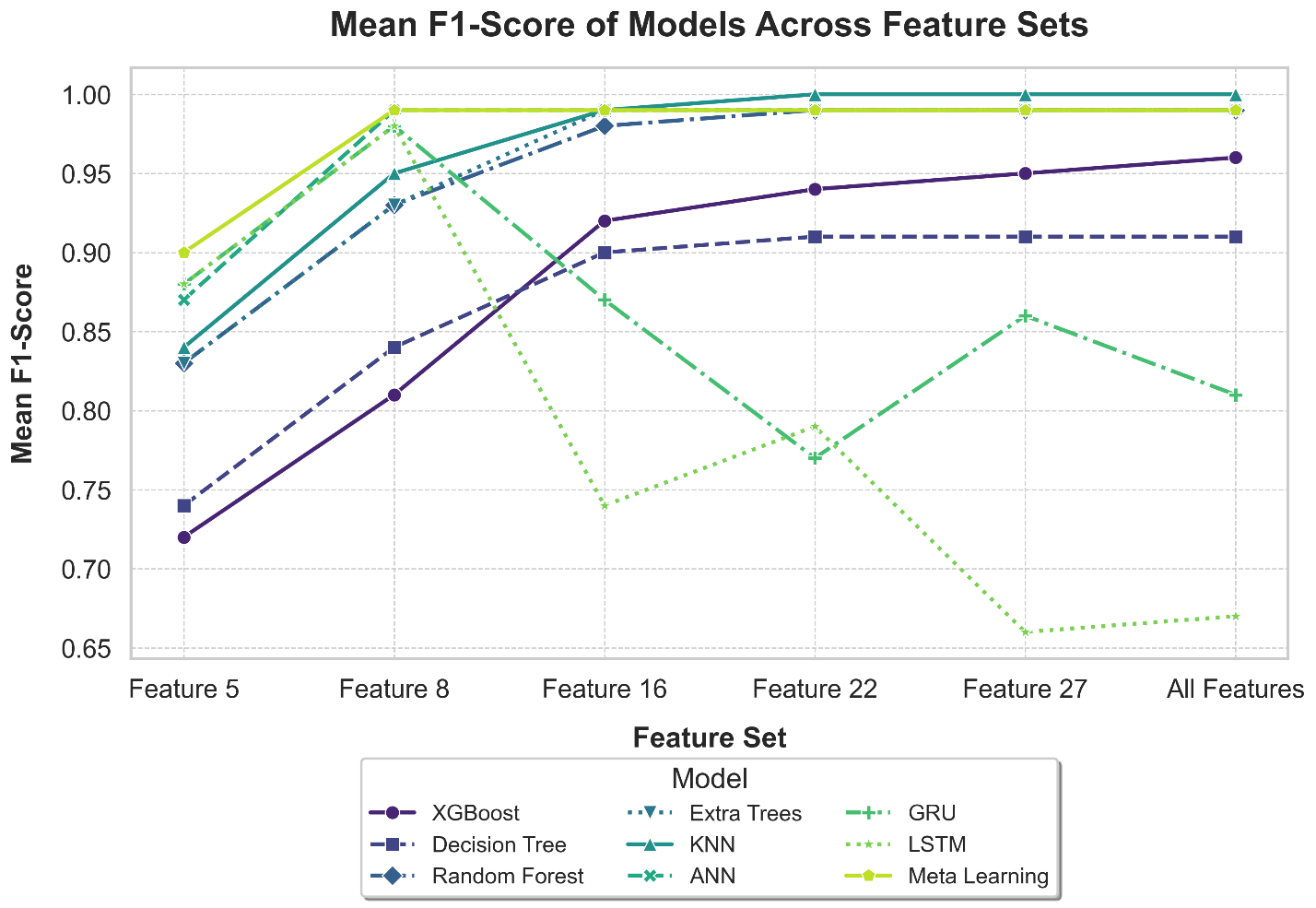
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Number of Features | XGBoost | Decision Tree | Random Forest | Extra Trees | KNN | ANN | GRU | LSTM | Meta Learning |
| 5 Features | 0.72 | 0.74 | 0.83 | 0.83 | 0.84 | 0.87 | 0.88 | 0.88 | 0.90 |
| 8 Features | 0.82 | 0.84 | 0.93 | 0.94 | 0.95 | 0.99 | 0.98 | 0.98 | 0.99 |
| 16 Features | 0.92 | 0.90 | 0.98 | 0.99 | 0.99 | 0.99 | 0.87 | 0.75 | 0.99 |
| 22 Features | 0.95 | 0.91 | 0.99 | 0.99 | 1.00 | 0.99 | 0.77 | 0.80 | 0.99 |
| 27 Features | 0.95 | 0.91 | 0.99 | 0.99 | 1.00 | 0.99 | 0.87 | 0.67 | 0.99 |
| 30 Features | 0.96 | 0.91 | 0.99 | 0.99 | 1.00 | 0.99 | 0.81 | 0.67 | 0.99 |



### **Figure:** Comparison of Mean Accuracy Across Models with Varying Feature Sets

## Mean F1-Score

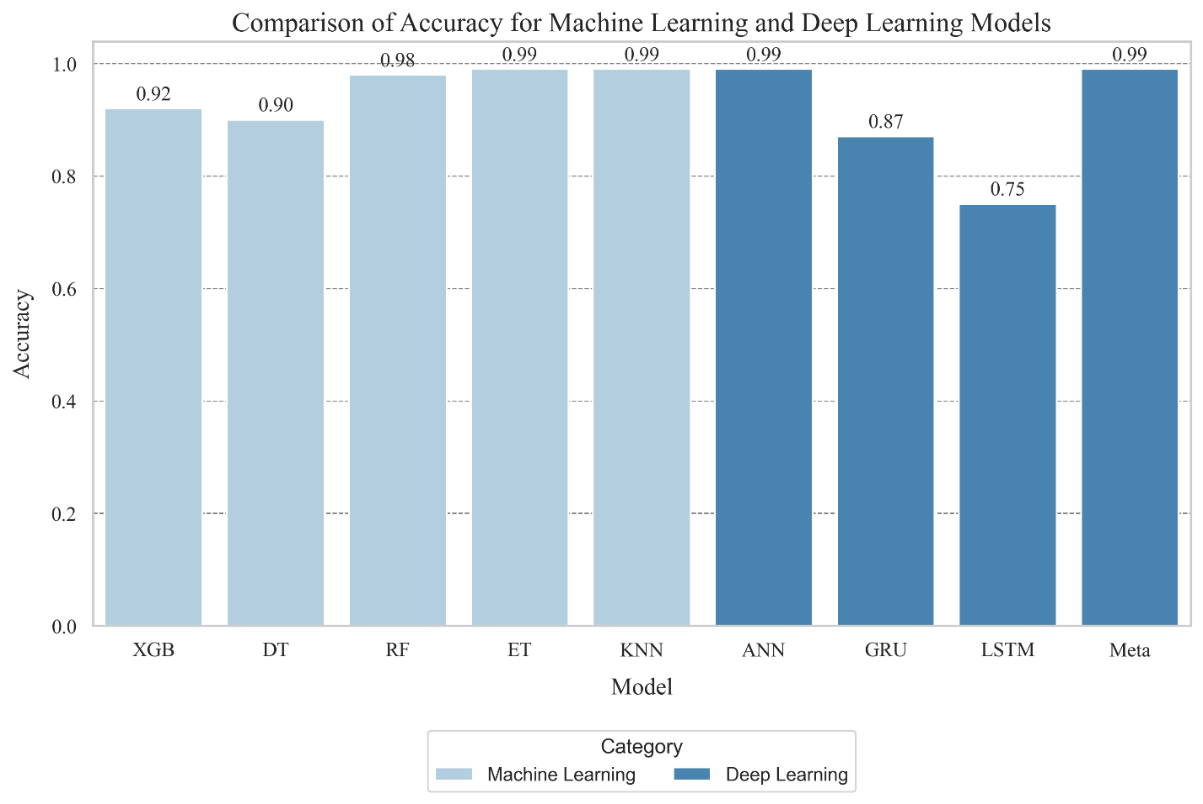
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Number of Features | XGBoost | Decision Tree | Random Forest | Extra Trees | KNN | ANN | GRU | LSTM | Meta Learning |
| 5 Features | 0.72 | 0.74 | 0.83 | 0.83 | 0.84 | 0.87 | 0.88 | 0.88 | 0.90 |
| 8 Features | 0.81 | 0.84 | 0.93 | 0.93 | 0.95 | 0.99 | 0.98 | 0.98 | 0.99 |
| 16 Features | 0.92 | 0.90 | 0.98 | 0.99 | 0.99 | 0.99 | 0.87 | 0.74 | 0.99 |
| 22 Features | 0.94 | 0.91 | 0.99 | 0.99 | 1.00 | 0.99 | 0.77 | 0.79 | 0.99 |
| 27 Features | 0.95 | 0.91 | 0.99 | 0.99 | 1.00 | 0.99 | 0.86 | 0.66 | 0.99 |
| 30 Features | 0.96 | 0.91 | 0.99 | 0.99 | 1.00 | 0.99 | 0.81 | 0.67 | 0.99 |



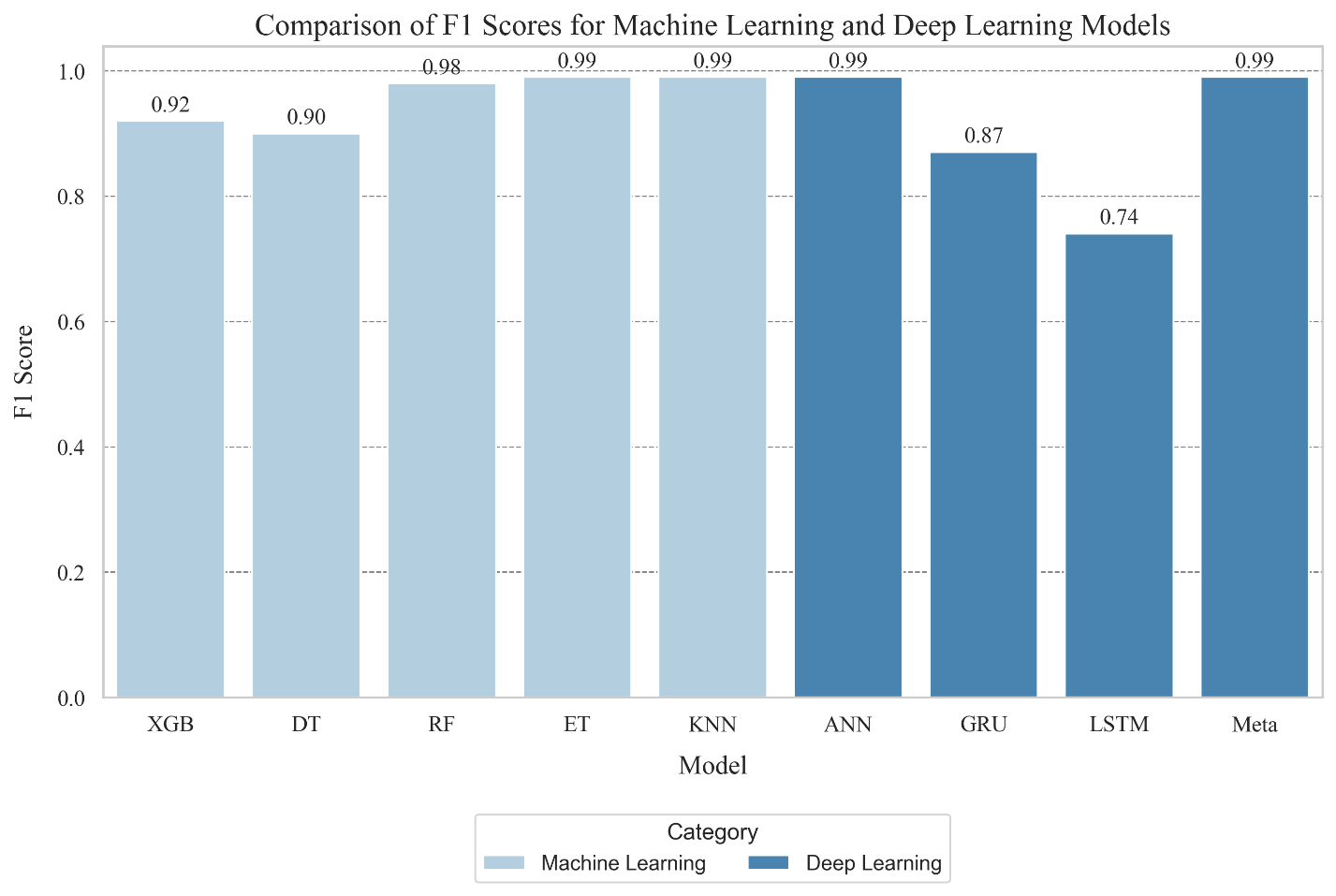
### **Figure:** Comparison of Mean F1 Score Across Models with Varying Feature Sets

## Performance Comparison Using 16 Features

|  |  |  |
| --- | --- | --- |
| Model | Accuracy | F1 Score |
| XGBoost | 0.92 | 0.92 |
| Decision Tree | 0.90 | 0.90 |
| Random Forest | 0.98 | 0.98 |
| Extra Trees | 0.99 | 0.99 |
| KNN | 0.99 | 0.99 |
| ANN | 0.99 | 0.99 |
| GRU | 0.87 | 0.87 |
| LSTM | 0.75 | 0.74 |
| Meta Learning | 0.99 | 0.99 |



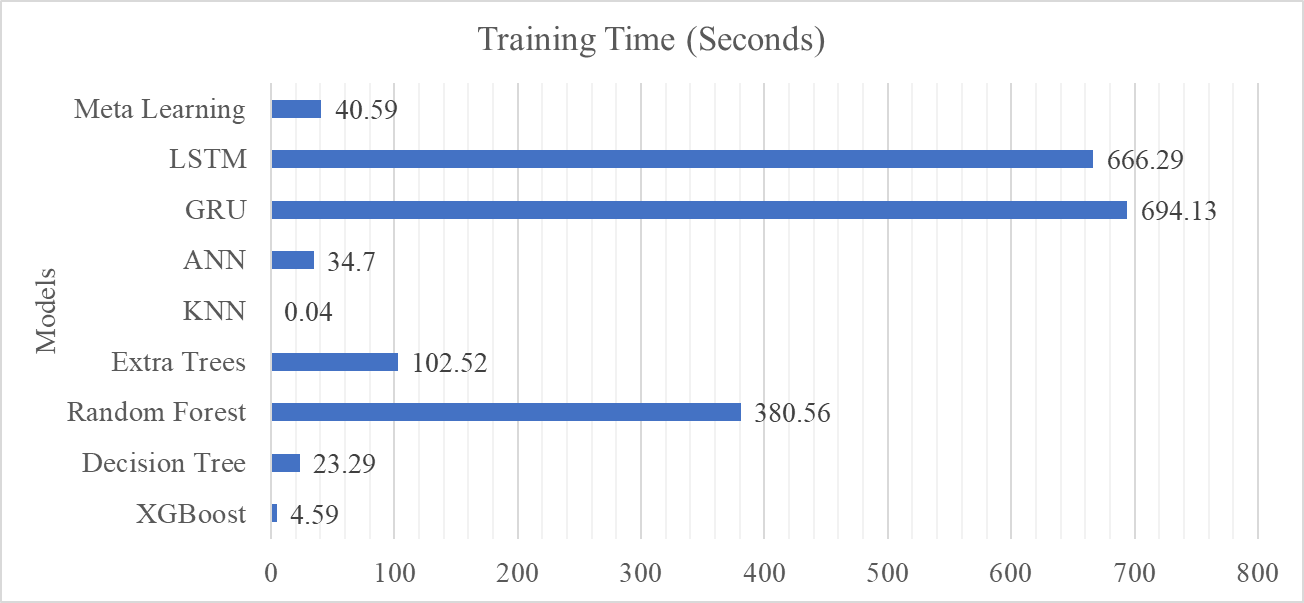
### **Figure:** Bar Plots Comparing Accuracy of Machine Learning and Deep Learning Models



### **Figure:** Bar Plots Comparing F1 Score of Machine Learning and Deep Learning Models

## Training Time Comparison Using 16 Features

|  |  |
| --- | --- |
| Model | Training Time (Seconds) |
| XGBoost | 4.59 |
| Decision Tree | 23.29 |
| Random Forest | 380.56 |
| Extra Trees | 102.52 |
| KNN | 0.04 |
| ANN | 34.70 |
| GRU | 694.13 |
| LSTM | 666.29 |
| Meta Learning | 40.59 |

****

### **Figure:** Training Time Comparison of All Models Using 16 Features