Parameter setting and reliability test of a sensor system for person detection in a car wearing summer wear

M. Eng. Information Technology
Machine Learning
Jiban Kumar Ray
1345968
jiban.ray@stud.fra-uas.de

M. Eng. Information Technology
Machine Learning
Md. Mijanur Rahaman
1324617
md.rahaman@stud.fra-uas.de

M. Eng. Information Technology
Machine Learning
Debasish Dutta
1343753
debasish.dutta@stud.fra-uas.de

***Abstract*— This project aims to optimize a Machine Learning model to accurately detect humans wearing summer clothing in the seat of a car using data from an ultrasonic sensor. The reliability of the sensor system is investigated and adjusted to ensure accurate detection. Confusion matrixes are used to evaluate performance. The project also compares additional measurements to the raw signal report. The experimental results will improve the accuracy of the ultrasonic sensor system and inform future research in the field of analyzing visual imagery with Machine Learning techniques.**

Keywords—Confusion Matrix, Random Forest Classifier, Machine Learning, Ultrasonic sensor, Data analysis, Data classification, FFT, ADC, Automotive sensor, Image processing.

I. Introduction

The field of machine learning has seen significant advancements in recent years, particularly in the development of autonomous intelligent systems. These systems are designed to perform tasks and make decisions independently, based on their ability to learn and adapt from experience. Machine learning has played a critical role in improving the efficiency and performance of AIS, particularly in applications like person detection. Person detection has become increasingly important in the automotive industry, as the number of vehicles on the road continues to rise. Regulations and laws have been introduced to improve safety standards, and autonomous vehicles have driven the need for more reliable and accurate person detection systems. [1] These systems not only detect the presence of passengers and pedestrians, but also enable autonomous alerts and information transmission to rescue authorities in case of an accident. In order to improve the performance of person detection systems, machine learning models are used to analyze and extract features from raw data received from sensors. [3] The pre-processing of data is critical in improving the reliability and accuracy of the models. Features such as maximum amplitude, reflected energy, and variation in FFT shape over time have proven to be effective in differentiating between humans and non-humans. One experiment in this field focused on fine-tuning the thresholds used in an existing model to optimize the classification between a human and an empty seat. This experiment was based on a replication of the system architecture from a previous study. Such experiments are vital in advancing the field of machine learning and autonomous systems, as they help refine and improve the performance of these systems in real-world applications.

II. Project Overview

The key motivation of this project is to determine the optimal settings for a sensor system that used for detecting people in a car who are wearing summer clothing. The work deals with investigating the reliability of the sensor system and ensure that it can accurately detect the presence of a person in a car, even when they are dressed in lightweight summer clothing that may be less visible or less easily detectable than heavier clothing. [2] Task accomplishment may involve testing and analyzing the data-generated from the ultrasonic sensor pointed at the car seat. Examine the reliability of the sensor system and test the accuracy through confusion matrixes that easily can differentiate among varying humans, empty seats and even a dummy doll. Then, develop and implement a suitable software program to systematically analyze reflected ultrasonic waves based on the profile of the objects. As a further scope, several additional measurements to compared to the report from the raw signals.

Overall, the experimental results of the project will help improving effectiveness and accuracy of the ultrasonic sensor, for detection of the clothes and classify which most applied ML techniques in analyzing visual imagery nowadays.

III. Experimental Setup

We used a cutting-edge sensor to collect data in a carefully controlled environment, taking measurements from every possible angle. After analyzing the data and optimizing our algorithms, we produced a highly accurate model that can predict environmental behavior. Our attention to detail at every step of the process ensured the success of our project.

A. SRF02 Ultrasonic Sensor with Red Pitaya

The setup used the SRF02 Ultrasonic Sensor, which is a single transducer ultrasonic rangefinder on a small PCB uses for both transmission and reception. Here single transducer resulting a higher minimum range than other dual transducer rangers. The minimum measurement range is around 15cm, and it can function with a 5V grounded power supply. The distance of the object from the sensor is calculated using this time and the speed of sound in air. The distinction between an empty seat and a human is made based on the frequency profile of the ultrasonic waves, as they bounce off harder surfaces with lower losses and incur more losses when bouncing off softer surfaces that tend to absorb some of these waves. [7] Because of the affordability Ultrasonic sensors are already being widely used in various applications such as in driver and parking assistance systems. Fig. 1 below represents the Ultrasonic Sensor and Red Pitaya card.

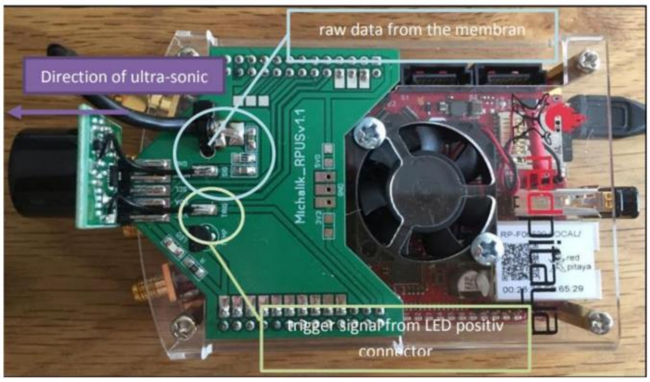

Fig. 1. Red Pitaya and Ultrasonic Sensor Layout

The sensor was connected to a Red Pitaya device, an open-source multifunction measurement tool that enables wireless data transmission to a laptop for further processing. It is an FPGA-based electronic board with two core ARM processors, and are commonly used in measurement systems, signal processing, and digital sensory implementation. It has the embedded Linux system that provides ease and flexibility in radio systems, vector network-analyzer, etc. During this project, Red Pitaya version STEM 125-14 V1.0 was used for measuring the data though UDP_Client program. The Red Pitaya device provided by the Lab for Autonomous System and Intelligent Sensors at the Frankfurt University of Applied Sciences.

B. UDP Client Software

The UDP_Client program is a vital tool in acquiring data from the Red Pitaya in this project. The program was developed by Daniel Schäfer from the Frankfurt University of Applied Sciences, and it boasts a user-friendly Graphical User Interface (GUI) as shown in Fig. 2. Through the use of this program, the team is able to efficiently gather data from the Red Pitaya, which is crucial for the success of the project.

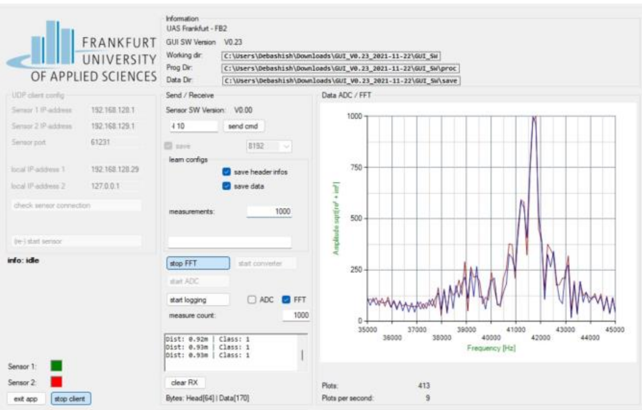

Fig. 2. UDP_Client GUI software

IV. Measurement System and Data Collection

In this project, data measurements were conducted using a car parked in an enclosed area of Frankfurt University of Applied Sciences. The collected data includes FFT and ADC data in a text file format with useful information such as data length, classification results, and sampling frequency.

A. Measurement System

The car was a white Ford Fiesta v16 for data collection as shown in the Fig. 3 below. The focus of the measurement was from the side seat shown in Fig. 4 to the driving seat, and an ultrasonic sensor was attached to the car console using a Red Pitaya device. The sensor was mounted at a distance of 72 cm from the center of the console and 29.5 cm from the car floor, with an angle of 22 degrees from the reference line. The car's side seat was adjustable.

Fig. 3. The Car Ford Fiesta v16 used.

Fig. 4. The side view with measurements of the car seat

B. Data Collection

The data was collected using a software developed by Master of Information Technology Engineering seniors, which can collect and observe data behavior in different situations and plot time-series and FFT/ADC graphs. The collected data was exported separately for FFT and ADC data, which is available in a text file format, along with data headers containing useful information such as data length, classification results, and sampling frequency. The data contains rows of amplitude values ranging from 0 to 1000 and columns of varying frequency range. The data is separated on a date-wise basis and can be found at [10]. As discussed in the Experimental Setup, the data was collected using UDP_Client GUI of the software. The software was set to measure analog data or raw data from the ultrasonic sensor.

C. Data Validation

The confusion matrix in this study displays a tabular representation of the different actual and predicted conditions measured. It includes various metrics that are used to evaluate the performance of the classification model, such as True Positive Rate, False Positive Rate, False Negative Rate, True Negative Rate, Accuracy, Precision, False Discovery Rate, False Omission Rate, and F1-score. F1-score indicates the accuracy of measured data set. Table I below as a reference example.

TABLE I. EVALUATION OF THE DATA SCORE

Label	***Score %***
Accuracy	87.4%
TPR	94.93%
TNR	77.33%
FPR	22.66%
FNR	5.06%
F1-Score	84%

Below Fig. 5 shows three persons image which we took before entering inside the car for data collection.

Fig. 5. Person Outfit and Specification

Following Table II summarizes all 9 individuals' specification.

TABLE II. PERSON SPECIFICATION

Person	***Height (cm)***	***Weight (Kg)***	***Outfit (Type)***
Per-1	164	62	Summer
Per-2	178	82	Summer
Per-3	155	61	Thin Wear
Per-4	178	66	Summer
Per-5	156	74	Summer
Per-6	171	70	Thin Wear
Per-7	174	62	Summer
Per-8	167	68	Summer
Per-9	165	87	Thin Wear

Each measurement data was collected first as FFT and then separately as ADC with 1000 sets of measurements at a time. Individual people wear different thin outfits. For the first step a total of 51,000 measurements which also includes an empty seat and a dummy doll. Again, we collected another 40,000 data for the second time with similar person with similar outfits. The confusion matrix measured for both first time data and the second time data. The in-depth analysis is represented in the result segment.

Precision measures the ability of the model to correctly predict positive cases out of all positive predictions, while recall measures the proportion of correct positive predictions out of all true positive cases. F1-Score is a combination of both Precision and Recall and balances between them.

D. Random Forest Classifier

Random Forest Classifier is a popular machine learning algorithm used for building classification models, shown in Fig. 6. It is an ensemble learning method that constructs a multitude of decision trees at training time and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. [4] The algorithm works in two major steps as given below:

- It randomly selects 'n' samples from the dataset.
- Builds a decision tree based on the selected samples. Each node in the tree is split based on the best split among a random subset of features.

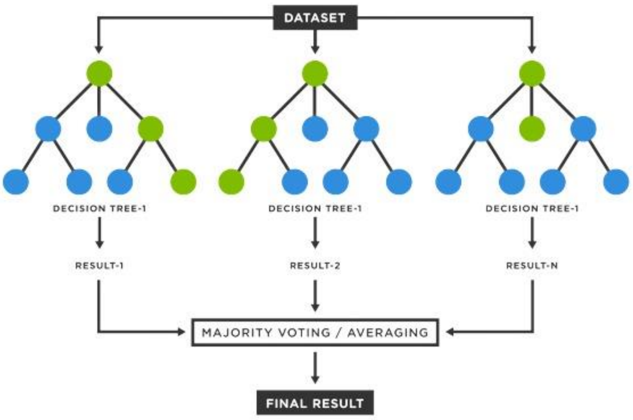

Fig. 6. Random Forest Algorithm

Repeat steps 1 and 2 'k' times to get 'k' decision trees. To classify a new object, let each decision tree classify the object and assign the object to the class that has the most votes among the 'k' trees. Fig. 7 shows the working flow.

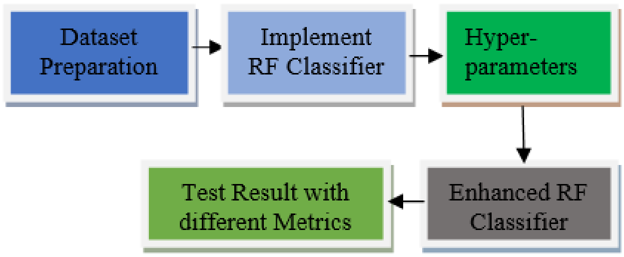

Fig. 7. Workflow of Random Forest Classifier

Random Forest Classifier includes robustness against overfitting. It can handle high-dimensional data with a large number of features which helps to handle missing data and provide feature importance ranking. But using Random Forest Classifier can be computationally expensive to train and make predictions for large datasets. It may not perform well when there are strong linear relationships between features and the target variable. It may not perform well with unbalanced datasets. So, we had to be careful about using the classifier keeping aside all the cons. Overall, Random Forest Classifier is a powerful and versatile algorithm that can be used for a wide range of classification tasks.

V. Experimental Method

In this segment, we will discuss the approaches we move forward to accomplish the project objective. The experimental work begins with train the classifier and to do so we used Random Forest Algorithm which we discussed in the section under Measurement Systems and Data Collection.

A. *Image Processing*

For the beginning of our experiment of FFT and ADC data, we write a code to detect summer wear and winter wear from the given image. A sample of individuals shown in Fig. 8 below:

Fig. 8. Image with Outboxing

B. *Programming Language*

We used Panda at Jupyter notebook, which is a popular data processing package in the Programming Language Python. Our programing code was written using python programming language version 3.9. Details of the code can be found in Github repository named MLGroup-4. The dataset was split into training and testing dataset. We are using Random Forest Classifier to run our dataset. To improve performance of our model we also use hyperperperameters to gain the highest possible accuracy.

After processing of collected data with torchvision for all 9 individuals and a dummy is shown in Fig. 9 below.

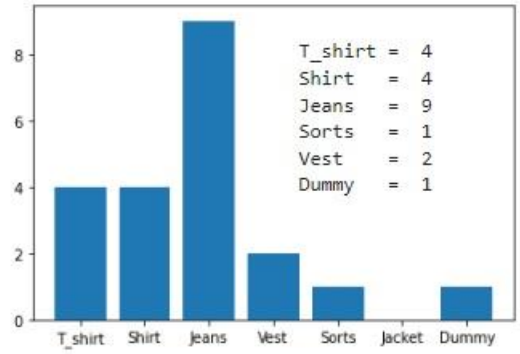

Fig. 9. Identification and Classification of Clothing

C. *Dataset Processing*

To make a proper environmental setup at first import all required packages and the load the dataset. The process involves compressing the dataset and reduce the size for efficient functioning. In this case we first compress it, upload into google drive and the again decompress this file to run.

```
os.chdir('Folderpath')
with rarfile.RarFile('rar file name', 'r') as rf:
    rf.extractall()
```

In this step, preprocessing the data to prepare and clean for making it usable for analysis or modeling. It is essential because raw data often contains inconsistencies, errors, and missing values that can negatively impact the quality of the analysis or modeling results. It involves cleaning the unwanted column and row from the data set. The first row of FFT data is sampling frequency and first column is header lengths. So, we clean first row and column by using python code before processing. Then fill missing value or NAN value with 0. Below code is run for the preprocessing.

```
# Preprocessing the data
cols = list(pd.read_csv(File_name, skiprows=[0], nrows =1,delimiter="\t"
))
df = pd.read_csv(File_name, skiprows=[0],usecols =[i for i in cols if i != '
V0.2'], delimiter="\t")
df.fillna(0, inplace=True)
```

Next is to create a target column in column number 2 with respect to the use case of datatype. In the preprocessed dataset column 3 represents the type of the data whether it is FFT and ADC, where the numeric value 2 indicates FFT type and 3 indicates for ADC. Now labeling the data by using rename in column 2, where 1 is for FFT data and 2 for the ADC data. After that dropping the predicted column values from dataset and classify it for X and predicted column for Y.

```
df['Target'] = 2
df.rename(columns = {'2':'Predicted'}, inplace = True)
X = df.drop(['Predicted'],axis=1).values
y = df.Predicted
```

D. Feature Extraction

Feature extraction is the process of selecting and transforming raw data into a set of features that can be used as inputs to a machine learning algorithm. In the case of a Random Forest Classifier trained on FFT data to detect the presence of a human in a seat. We extract some feature from our dataset. Its included Mean, Standard deviation, Skewness, Kurtosis, Frequencies, Peak Frequencies, Dominant frequency, Spectrum, Spectral entropy, Total energy, Spectral centroid, and Feature vector from our FFT data shown in below Fig. 10.

```
Mean: 64.00
Standard deviation: 0.00
Skewness: 9.85
Kurtosis: 95.01
Frequencies: [ 0.00000000e+00  9.99700090e-05  1.99940018e-04 ... -2.99910027e-04
 -1.99940018e-04 -9.99700090e-05]
Peak frequencies: [9.99700090e-05 3.99880036e-04 5.99820054e-04 ... 9.99991003e+01
 9.99993002e+01 9.99996001e+01]
Dominant frequency: 9.997000899730081e-05
Spectrum: [ 1.97486195e+10+7.33416528e-07j -9.09682248e+06+1.12961571e+07j
  1.04478567e+07-2.04653080e+07j ...  2.38814899e+06-7.13841140e+06j
  1.04478567e+07+2.04653080e+07j -9.09682248e+06-1.12961571e+07j]
Spectral entropy: 64     -3.407979e+08
```

Fig. 10. Extracted Features from the FFT data of a single person.

E. Random Forest Model Training

The dataset is then split into two subsets, one for training the model and the other for testing or validation of the model's performance. Before running RF classifier, splitting train and test subset which is given below.

```
X_train, X_test, y_train, y_test = train_test_split(
   X, y, test_size=.1, random_state=42)
```

After that running dataset by using hyperperameter n_estimators=100 and random state=.42. The classifier command is shown below.

```
clf = RandomForestClassifier(n_estimators=100, random_state=42)
```

Fit the classifier to train splitting train data and making a prediction on the test data with this command.

```
clf.fit(X_train, y_train)
pred = clf.predict(X_test)
```

VI. Model Evaluation And Result

In this segment, we will thoroughly evaluate the model with relevant results. A confusion matrix is a table that is used to evaluate the performance of a classification model. It compares the predicted class labels of a model with the true class labels and displays the number of correct and incorrect predictions made by the model. The confusion matrix has four categories:

- True positives (TP): The model correctly predicted the positive class.
- False positives (FP): The model predicted the positive class, but it was negative.
- True negatives (TN): The model correctly predicted the negative class.
- False negatives (FN): The model predicted the negative class, but it was positive.

A. Without Classifier Training with 10,000 data

At the beginning, without classifier training he confusion matrix is generated as shown Fig. 11 below.

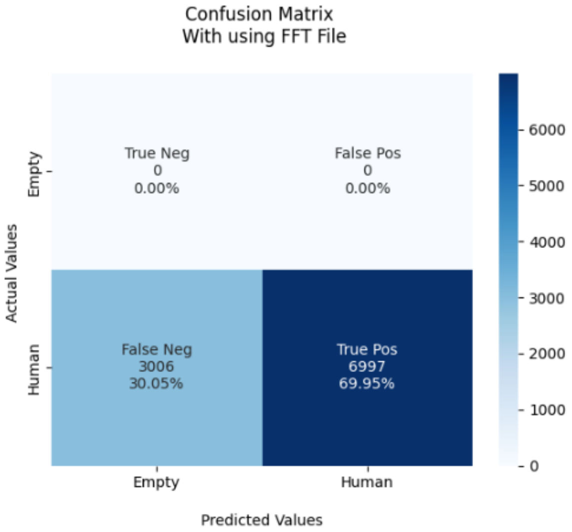

Fig. 11. Confusion Matrix without Classifier Training.

Below Fig. 12 represents classification report including F-1 score, precision and Validation Accuracy while the classifier are not trained.

Predicted Values

	precision	recall	f1-score	support
1	0.00	0.00	0.00	0
2	1.00	0.70	0.82	10003
accuracy			0.70	10003
macro avg	0.50	0.35	0.41	10003
weighted avg	1.00	0.70	0.82	10003

Fig. 12. Classification Report of Without Classifier Training.

B. FFT Results of Classifier Training with 10,000 data

The confusion matrix below in Fig. 13 is generated from the first 10,000 dataset without empty or dummy data.

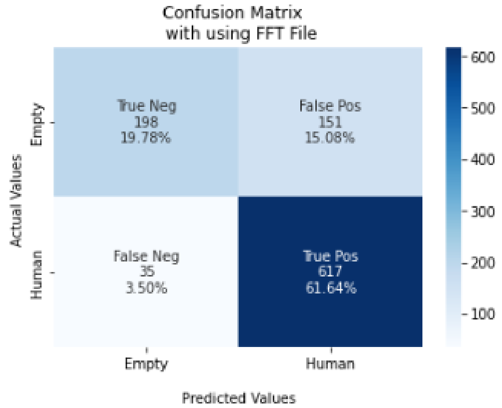

Fig. 13. Confusion Matrix of 10,000 data.

Below Fig. 14 represents the extracted features from RF classification report including F-1 score, precision and Validation Accuracy.

	precision	recall	f1-score	support
1	0.85	0.57	0.68	349
2	0.80	0.95	0.87	652
accuracy			0.81	1001
macro avg	0.83	0.76	0.77	1001
weighted avg	0.82	0.81	0.80	1001

Validation accuracy: 81.42 %

Fig. 14. RF Classification Report of 10,000 data.

Fig. 15 below depicts the accuracy graph of the classifier which generated from the first 10,000 dataset.

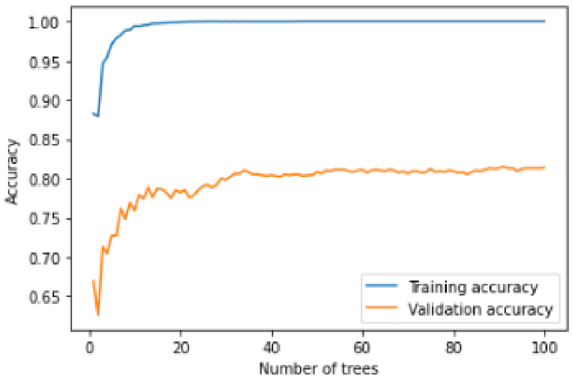

Fig. 15. Accuracy graph of 10,000 data.

The error overview shown in Fig. 16 below.

```
Average Traing Errors = 0.004667333111185163
Average Validation Errors = 0.2048151848151848
Difference Between Traing and Validation Errors = 0.20 or 20.01%
```

Fig. 16. Error Overview of 10,000 data.

C. FFT Results of Classifier Training with 30,000 data

The confusion matrix below in Fig. 17 is generated from the second 30,000 dataset.

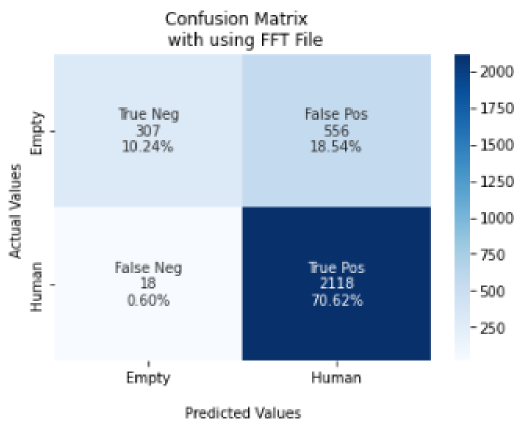

Fig. 17. Confusion Matrix of 30,000 data.

Below Fig. 18 represents the extracted features from RF classification report including F-1 score, precision and Validation Accuracy.

	precision	recall	f1-score	support
1	0.94	0.36	0.52	863
2	0.79	0.99	0.88	2136
35167	1.00	1.00	1.00	2
accuracy			0.81	3001
macro avg	0.91	0.78	0.80	3001
weighted avg	0.84	0.81	0.78	3001

Validation accuracy: 0.8087304231922693

Fig. 18. RF Classification Report of 30,000 data.

Fig. 19 below depicts the accuracy graph of the classifier.

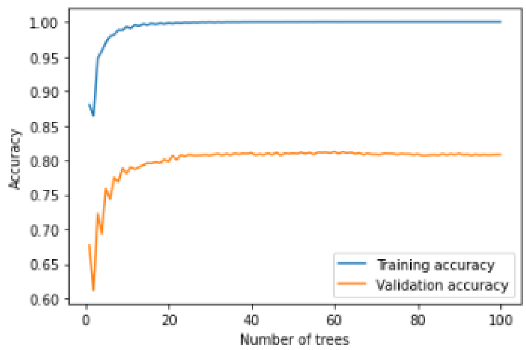

Fig. 19. Accuracy Graph of 30,000 data.

The error overview shown in Fig. 20 below.

```
Average Traing Errors = 0.005153321976149904
Average Validation Errors = 0.19997000999666778
Difference Between Traing and Validation Errors = 0.19 or 19.48%
```

Fig. 20. Error Overview of 30,000 data.

D. FFT Results of Pre-trained Classifier Person by Person

The confusion matrix now generated for individual person with pre-trained classifier and classified measurement data person by person. Fig. 21 below is an example confusion matrix for a single person of Per-9 as of Table II.

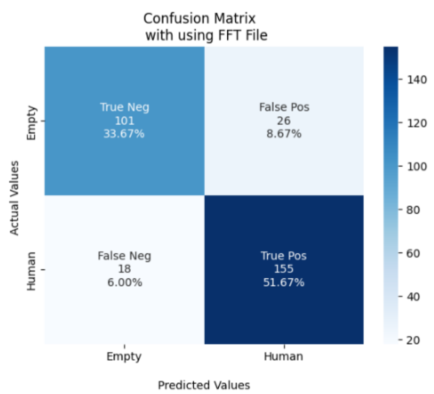

Fig. 21. Confusion Matrix of single person (Per-9).

Below Fig. 22 represents the extracted features from RF classification report including F-1 score, precision and Validation Accuracy.

```
              precision    recall  f1-score   support

           1       0.85      0.80      0.82       127
           2       0.86      0.90      0.88       173

    accuracy                           0.85       300
   macro avg       0.85      0.85      0.85       300
weighted avg       0.85      0.85      0.85       300

Validation accuracy: 85.33 %
```

Fig. 22. RF Classification Report of Single Person (Per-9).

Fig. 23 below depicts the accuracy graph of the classifier for Per-9.

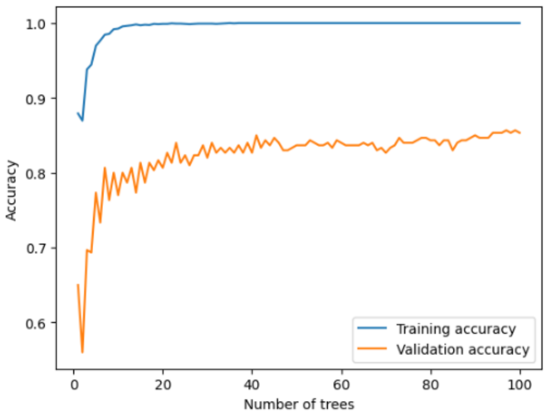

Fig. 23. Accuracy Graph of Single Person (Per-9).

The error overview generated for Per-9 shown in Fig. 24 below.

```
Average Traing Errors = 0.0050463134494257 08
Average Validation Errors = 0.17723333333333334
Difference Between Traing and Validation Errors = 0.17 or 17.22%
```

Fig. 24. Error Overview of Per-9.

E. FFT Results of Pre-trained Classifier for Empty Seat

The confusion matrix now generated for empty seat with pre-trained classifier shown in Fig. 25 below.

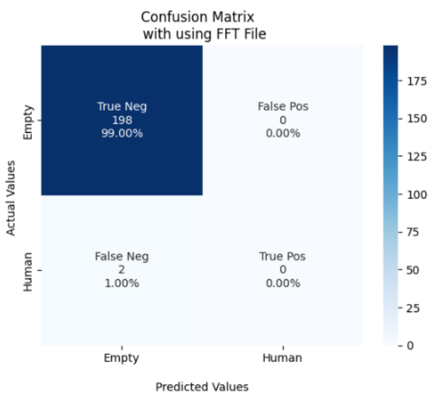

Fig. 25. Confusion Matrix of Empty Seat.

Below Fig. 26 represents the extracted features from RF classification report including F-1 score, precision and Validation Accuracy.

```
              precision    recall  f1-score   support

           1       0.99      1.00      0.99       198
           2       0.00      0.00      0.00         2

    accuracy                           0.99       200
   macro avg       0.49      0.50      0.50       200
weighted avg       0.98      0.99      0.99       200

Validation accuracy: 99.00 %
```

Fig. 26. RF Classification Report of Empty Seat.

Fig. 27 below depicts the accuracy graph of the classifier for empty seat.

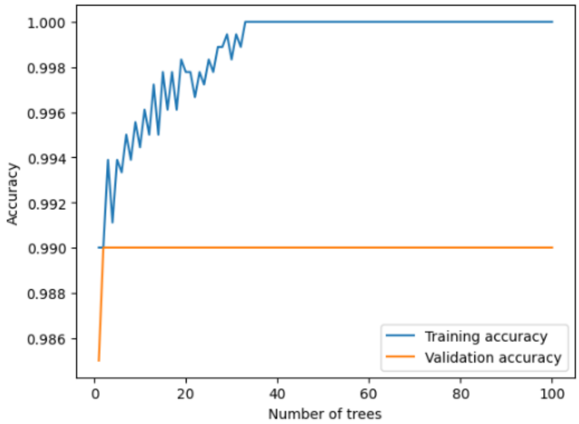

Fig. 27. Accuracy Graph of Empty Seat.

The error overview generated for empty seat shown in Fig. 28 below.

```
Average Traing Errors = 0.0012222222222222224
Average Validation Errors = 0.010050000000000009
Difference Between Traing and Validation Errors = 0.01 or 0.88%
```

Fig. 28. Error Overview of Empty Seat.

F. FFT Results of Pre-trained Classifier for Dummy Doll

The confusion matrix now generated for a dummy doll shown in Fig. 29 below.

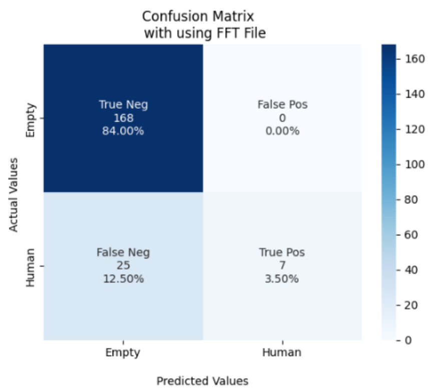

Fig. 29. Confusion Matrix of Dummy Doll.

Below Fig. 30 represents the extracted features from RF classification report including F-1 score, precision and Validation Accuracy.

	precision	recall	f1-score	support
1	0.87	1.00	0.93	168
2	1.00	0.22	0.36	32
accuracy			0.88	200
macro avg	0.94	0.61	0.64	200
weighted avg	0.89	0.88	0.84	200

```
Validation accuracy: 87.50 %
```

Fig. 30. RF Classification Report of Dummy Doll.

Fig. 31 below depicts the accuracy graph of the classifier for Dummy Doll.

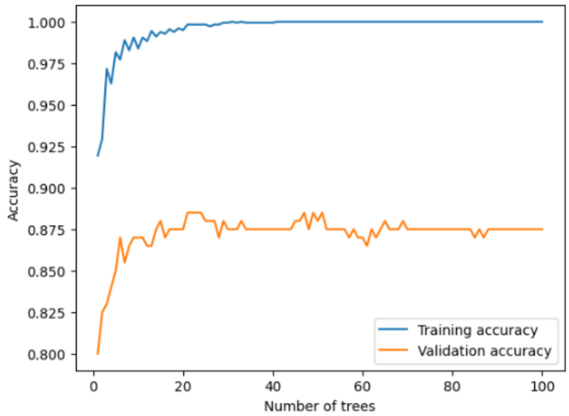

Fig. 31. Accuracy Graph of Dummy Doll.

The error overview generated for a dummy doll shown in Fig. 32 below.

```
Average Traing Errors = 0.004000000000000001
Average Validation Errors = 0.12725
Difference Between Traing and Validation Errors = 0.12 or 12.32%
```

Fig. 32. Error Overview of Dummy Doll.

G. ADC Results of Pre-trained Classifier for 10,000 dataset

The confusion matrix now generated with ADC data measurement for the 10,000 datasets, shown in Fig. 33.

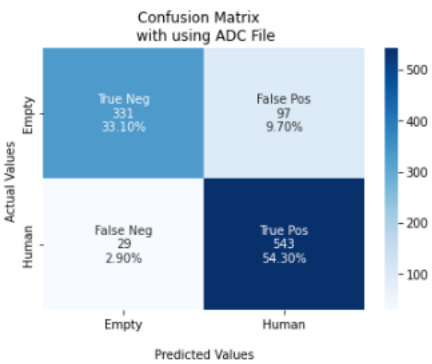

Fig. 33. ADC data Confusion Matrix.

Below Fig. 34 represents the extracted features from RF classification report including F-1 score, precision and Validation Accuracy.

	precision	recall	f1-score	support
1	0.85	0.95	0.90	572
2	0.92	0.77	0.84	428
accuracy			0.87	1000
macro avg	0.88	0.86	0.87	1000
weighted avg	0.88	0.87	0.87	1000

```
Validation accuracy: 87.40 %
```

Fig. 34. ADC data RF Classification Report.

Fig. 35 below depicts the accuracy graph of ADC data.

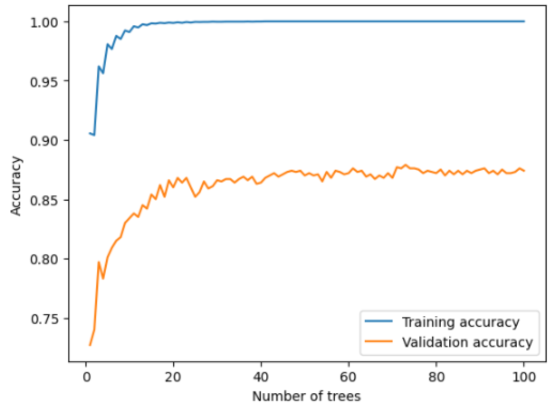

Fig. 35. ADC data Accuracy Graph.

The error overview generated for a dummy doll shown in Fig. 36 below.

```
Average Traing Errors = 0.0038986441431429174
Average Validation Errors = 0.13916000000000003
Difference Between Traing and Validation Errors = 0.14 or 13.53%
```

Fig. 36. ADC data Error Overview.

VII. Discussion

The experiment setup included various variations such as changing the distance of the seat from the sensor, adjusting the backrest inclination, wearing different clothes and accessories, opening/closing the car door during measurement, and making bodily movements. The backrest inclination did not have a significant impact on the classification results as the sensor was directed towards the stomach position of the human. However, keeping the car door open during measurement had a significant impact on the classification results due to the ultrasonic waves being affected by surrounding objects or humans, resulting in an erratic frequency profile. The empty seat positioned closest to the sensor reflected more energy, resulting in a higher frequency peak, whereas the farthest seat position reflected less energy and had a lower frequency peak.

VIII. Conclusion

In this project, we have explored the potential of machine learning models for classifying humans wearing summer clothes in a car seat using data from an ultrasonic sensor. Our findings suggest that the reflected ultrasonic waves show different patterns with different subjects and setups, which can have a significant impact on the classification performance of the models.

We have also identified various strategies for improving the classification results, such as balancing variance and bias, adjusting model parameters, and acquiring more training data. Through our experiments, we observed a bias towards misclassifying an empty seat as a human, which we addressed by adjusting the initial model parameters.

However, we recognize that there is still much to understand about the features in the data and how they relate to human presence in a car seat. We believe that by using larger and more complex pretrained models, and by training them with our specific experimental data through collective effort, we can continue to make progress towards more accurate and reliable classification results.

In conclusion, this project has demonstrated the potential of machine learning for classifying humans in a car seat using ultrasonic data. Our findings suggest that there are still challenges to be addressed, but by continuing to explore and refine our models, we can make significant advances in this area. We hope that our work can contribute to the development of more robust and effective classification techniques for applications such as automotive safety and security.

IX. Acknowledgement

We would like to express our sincere appreciation to Professor Andreas Pech for his invaluable guidance and support throughout this project. His extensive knowledge and expertise in the field of machine learning have been instrumental in shaping our understanding and helping us to develop our ideas. We also extend our gratitude to the Information Technology department for providing us with the necessary infrastructure and resources to conduct our research and complete our work. Their technical support and access to software and hardware have been crucial in enabling us to carry out our experiments and analyze our results. Moreover, we would like to acknowledge the efforts of the authors of the references and literature that we consulted during our research. Their works have provided us with valuable insights and perspectives on the topic and have helped us to develop a comprehensive understanding of the subject matter. We are also grateful for the opportunity to participate in the "Machine Learning" lectures, which have been a significant source of motivation and inspiration for our work. The lectures have provided us with a solid foundation in the fundamentals of machine learning, and we have been able to build upon this knowledge to develop our own ideas and insights.

Finally, we would like to express our appreciation to all those who have supported us in this project, whether through their advice, feedback, or encouragement. We could not have completed this work without their contributions, and we are grateful for their support.

References

[1] A. H. Pech, P. M. Nauth and R. Michalik, “A new Approach for Pedestrian Detection in Vehicles by Ultrasonic Signal Analysis,” IEEE EUROCON 2019 -18th International Conference on Smart Technologies, 2019, pp. 1-5, doi: 10.1109/EUROCON.2019.8861933.

[2] P. M. Nauth, A. H. Pech and R. Michalik, “Research on a new Smart Pedestrian Detection Sensor for Vehicles,” 2019 IEEE Sensors Applications Symposium (SAS), 2019, pp. 1-5, doi: 10.1109/SAS.2019.8705978.

[3] K.-H. Liu, T.-Y. Chen, C.-S. Chen, "MVC: A Dataset for View-Invariant Clothing Retrieval and Attribute Prediction", Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, pp. 313-316, 2016.

[4] "Random Forest," [Online]. Available: https://www.tibco.com/reference-center/what-is-a-random-forest.

[5] D. Gabor, "Theory of Communication," Journal of Institution of Electrical Engineers, vol. 93, no. 3, pp. 429-457, 1946.

[6] Ajitesh Kumar, “Accuracy, Precision, Recall & F1-Score”, published on 3 September 2021, accessed on 20 September 2021.

[7] R. ML-AIS, “ML-AIS-Sensor-Project-2021/Data at main · ML-AIS/MLAIS-Sensor-Project-2021,” GitHub, Apr. 30, 2022. [Online]. Available: https://github.com/ML-AIS/ML-AIS-Sensor-Project-2021/tree/main/Data.

[8] Y. S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo and J. Choe, “CutMix: Regularization Strategy to Train Strong Classifirs With Localizable Features,” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6022-6031, doi: 10.1109/ICCV.2019.00612.

[9] Q. Dong, S. Gong, X. Zhu, Multi-Task Curriculum Transfer Deep Learning of Clothing Attributes, 2016.

[10] R. Jiban Kumar, D. Dutta and M. Rahman ML Project data “Collected FFT-ADC data, Programing Code, Output Reports, Detailed Documentation”. 31 March 2023. Available at: https://github.com/MdMijanur-Rahaman/FIUS-Sonar-Sensor-project-of-Machine-Learning-.git