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

*Abstract*—Machine Learning, a branch of artificial intelligence, enables systems to learn from data, identify patterns, and make decisions with minimal human intervention. It is a powerful method of data analysis that automates analytical model building. In this paper, our goal is to improve the reliability of a sensor system for human detection in a car wearing winter wear. To achieve this, we take readings using the Red Pitaya and feed the data into a chosen model, which in this case is the SVM algorithm. Then we frequently evaluated the classification accuracy and classification error rate using the proposed algorithm with the SVM. We found an average classification accuracy rate of 98.6% for a human with or without motion. The effectiveness of the sensor is evaluated by analyzing the F1 score, which is generated from the modified data in the confusion matrix.

*Keywords- Machine learning, SVM, Confusion Matrix, Red Pitaya, Ultrasonic Sensor,* passenger detection.

I. INTRODUCTION

The deaths of children in hot vehicles have lately become a significant societal problem. As a result, the European NCAP (New Car Evaluation Program) has suggested that CPD (Child Presence Detection) equipment be installed on all new vehicles beginning in 2020 [24].

Also, several nations' safety officials have thought about regulations that may compel CPD systems designed to find a child left in a car. A variety of sensors that can identify items within automobiles or keep track of the condition of the vehicle body are needed to enable such a system. [25] [26]

Electric cars are another target for passenger detection. The effectiveness of the battery affects the heating and cooling systems in electric cars [27]. Battery usage might be reduced if heating and cooling systems in cars could be automatically adjusted for each seat. Thus, technology to recognize occupants in each seat is needed to provide these activities.

The use of occupant detection in self-driving cars is another application. The presence or absence of passengers has a significant impact on how an autonomous vehicle drives.

In other words, when people are riding, their comfort and dependability become crucial factors [28]. Furthermore, the self-driving style might change based on whether the passenger is awake or asleep. Thus, it is crucial to evaluate the number of passengers and their state in each seat.

The performance capabilities of sensors to identify passengers are crucial for the many applications discussed above.

We can utilize many sensors for example pressure sensors, thermal infrared sensors, using camera, radar sensor-based or ultrasonic sensors. In this paper our focus is on Ultrasonic sensors.

Machine learning, a subfield of Artificial Intelligence (AI), allows computers to automatically learn and improve their performance based on a starting set of data. Computers can learn from experience by making decisions based on data received, removing the need for explicit human programming. The primary goal of machine learning is to enable computers to learn and adjust their behavior without the assistance of humans. [1]

Machine learning models rely on sensor input data to generate meaningful outputs that autonomous systems can use to perform complex tasks. A critical aspect of this process is machine perception, or the ability of machine learning algorithms to interpret raw data. The ability to recognize humans, for example, is required for autonomous systems to perform safety-critical tasks. Raw data pre-processing is a critical step in improving the accuracy and reliability of machine learning algorithms. Ultrasonic signals from both human and non-human targets are collected and analyzed in this context to form the input data for the systems. By understanding the significance of pre-processing raw data, we can enhance the performance and dependability of machine learning algorithms and enable more advanced autonomous systems. [2]

These are investigated in order to determine a pre-processing method that is efficient for feature extraction. In general, the following parameters are useful in separating humans from non-humans: maximum amplitude, reflected energy, and variation in Fourier transform shape over time. [2]

Our research on evaluating and verifying the dependability of a sensor mechanism designed for detecting humans in a car, specifically when they are dressed in cold weather attire. The objective is to distinguish and improve the efficiency of the sensor's ability to identify if a person is present in a car setting, particularly while wearing various garments for winter season such as scarves, coats, and hats. To collect our data, we employ a sensor and utilize Fourier transforms. After adjusting the output readings during the project and acquiring the requisite set of values and observations, we generate multiple confusion matrices for our study using these measurement datasets. We then use the data obtained from these matrices to train an RF model further, improving the dependability and accuracy of the sensor's output.

1. THEORITICAL BACKGROUND

An overview of the theoretical elements of the research will be given in this part. For instance, we shall outline the hardware requirements, the programming language utilized, and the developed underlying algorithm. The kind of microprocessor or STEM Lab board, processing speed, ADC and DAC bandwidth, and any further pertinent hardware details are included. The term "underlying algorithm" describes the precise mathematical or computational procedure employed to produce the intended result for the project. This could, among other things, be a machine learning algorithm or a signal processing algorithm.

1. *Ultrasonic Sensor*

Ultrasonic distance sensors can detect objects or the distance to objects within range without the need for physical contact. Ultrasonic sensors use sound waves which are above human hearing frequency (above 20 KHZ) to detect objects in their vicinity. They utilize a piezoelectric transducer to convert mechanical vibration into an electrical signal, which enables them to detect most objects, regardless of their material, color, or state (liquid, solid, granular). Ultrasonic sensors are less affected by condensing moisture than photoelectric sensors. However, they are not effective at targeting objects made of sound-absorbing materials like cloth, soft rubber, flour, or foam. An ultrasonic proximity sensor consists of four main components: transmitter/receiver, comparator, detector, and output circuit. The transducer emits a series of sonic pulses, and the reflected echo is converted into an analog or digital signal. Sound emitters and receivers, such as piezoelectric or magnetostrictive transducers, are used in ultrasonic sensors. [3] [4]

Some of the characteristics of ultrasonic sensors include:

* They are inexpensive.
* They are small and unobtrusive.
* They can measure distances across smoke, dust, and humidity.
* They do not have any lenses to clean. [29]

The sensor can generate an ultrasonic wave through the transducer using electrical signals to create mechanical vibrations. When the ultrasonic wave reflects off an object within the sensor's range, it causes a mechanical vibration that produces an electrical signal. This information, along with the characteristics of the environment, can be used to calculate the distance to the object or determine information about it. Ultrasonic sensors use either 'direct detection' or 'beam interruption detection' principles to measure distance. [5][6]

Diagram, shape

Description automatically generated

Fig. 1: Ultrasonic Time-of-Flight Measurement [6]

Ultrasonic devices are capable of detecting objects within their range, but they cannot distinguish between various shapes and sizes. This limitation, however, can be overcome by using two devices instead of just one. Both devices can be placed at different distances or next to each other. Examining the overlapped dark zone reveals the target item's form and size.

1. *Red Pitaya*

The Red Pitaya microchip STEM Lab board provides high bandwidth for the Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC). It is a popular option for radio frequency applications due to its feature, especially as a radio receiver and transmitter. Because of its wide variety of characteristics, the Red Pitaya is frequently used in several radio frequency applications. [8]

Diagram, schematic

Description automatically generated

Fig. 2. Red Pitaya [7]

For our project, the Red Pitaya version STEM 125-14 V1.0 was used, which was obtained from the Lab for Autonomous System and Intelligent Sensors at Frankfurt University of Applied Sciences. The device came with a preconfigured Linux system and the UDP\_Client program provided by the Lab, which were used to carry out the measurements.

1. *First Fourier Transform (FFT)*

The Red Pitaya server uses the idea of Fast Fourier Transform to give us output in the desired numerical discrete format. A general understanding of the Fast Fourier transform is required to gain a thorough understanding of the project deliverables. It breaks down a signal into its spectral components and outputs frequency data as a result. FFTs are used in machine and system defect analysis, quality control, and condition monitoring.

The Discrete Fourier Transformation (DFT) is best performed with precision using the FFT method. This method involves repeatedly sampling a signal to separate it into separate frequency components, each of which is made up of a single sinusoidal signal with a different frequency, amplitude, and phase. [9]

1. *Algorithm*

By using data instead of explicit programming, machine learning enables computers or other machines to learn and make decisions. For example, the collection of Wikipedia entries is divided into various categories. There are two different types of Machine Learning such as

1. Supervised Learning
2. Unsupervised Learning.

* Supervised Learning

In supervised machine learning, a human expert uses training data to instruct a computer how to link particular inputs with predetermined outputs. The computer then applies this understanding to forecast potential patterns based on new data. The objective of supervised learning is to create an overarching rule that can precisely map inputs to outputs. The methods used in classification and regression are two common types of supervised learning. [10]

Our experiment involves using a random forest algorithm to train a model using a generated dataset obtained from Red Pitaya. The objective is to detect people wearing winter clothing under various applied conditions.

* Unsupervised Learning

Unsupervised learning techniques involve teaching a computer using unlabeled data without the need for a human expert. The learning algorithm must identify the structure in the input data on its own, without any guidance from labels or predetermined outcomes. Unsupervised learning algorithms use various methods and procedures to identify patterns and rules and summarize data points in order to provide insights and better represent the data to users. Experts are informed about the different types of patterns present in the data through the algorithms used in this data-driven learning approach. [10]

1. *Random Forest (RF) Algorithm*

The Random Forest (RF) algorithm creates decision trees using provided data samples, gathers predictions from each tree, and then selects the best option through voting. It also provides insight into the significance of each feature. The Random Forest algorithm creates a forest of trees by integrating multiple decision trees, and the more trees in the forest, the more accurate the random forest classifier becomes. The Random Forest algorithm takes less time to train than other algorithms and can predict outcomes accurately and quickly even with large datasets. It can also keep its accuracy even when a large amount of data is missing.

The Random Forest algorithm has two stages: the first stage involves creating the random forest by constructing N decision trees, while the second stage involves making predictions for each tree generated in the first stage. The algorithm's working process can be illustrated using a diagram and a set of steps.

To implement the Random Forest algorithm, the following steps can be followed:

1. Randomly select K data points from the training set.
2. Create subsets of the data points selected in step a) and generate decision trees based on these subsets.
3. Determine the number N of decision trees to be created.
4. Repeat steps a) and b) to generate N decision trees.
5. When making predictions for new data points, gather forecasts from each decision tree and assign the data to the category with the most votes.

The RF algorithm is versatile and can be used for both classification and regression problems. However, the way they predict values differs. Classification algorithms provide probabilities for class labels, while regression methods estimate discrete values like integers. The evaluation process for these methods can vary depending on certain conditions.

* Classification predictions can be evaluated for accuracy, but regression predictions cannot. However, the root mean square method can be used for regression evaluation, whereas it cannot be used for classification evaluation.
* The RF algorithm is known for its high accuracy and resilience due to its use of many decision trees. It avoids overfitting by averaging out the biases of the decision trees.
* The random forest classifier can handle missing values by either replacing continuous variables with median values or by determining the proximity-weighted average of missing variables.
* Feature selection can be achieved using the RF classifier, which identifies the most important characteristics from the available features in the training dataset.

Diagram

Description automatically generated

Fig 3. Random Forest ML Algorithm [11]

1. *Support Vector Machine*

In order to classify time signals, SVM (Support Vector Machine) was utilized, which is a powerful machine learning approach for classification. SVM is a binary classifier that aims to locate a hyperplane that separates the data points of different classes as much as possible, while also minimizing the distance between the hyperplane and each class [15].

Chart, diagram, scatter chart

Description automatically generated

Fig. 4. The Transformation

While many previous classifiers use hyperplanes to distinguish between classes, SVM goes beyond this approach by mapping the predictors to a higher dimensional space, allowing for linear separation of data that cannot be separated by a hyperplane in its original dimensionality. Typically, classification errors arise when either an inappropriate kernel function is chosen or there are instances that belong to different classes. The process of determining the optimal location for the decision plane is essentially an optimization problem, which involves using a kernel function to create nonlinear boundaries through linear transformations [16][17]. The set of data used to train the support vector classifier, denoted by P, can be defined as:

*P =* {(*xi, yi*)|*xi* ∈ *ℝ k, yi* ∈ {−1*,* 1}}*n* (1)

In order to perform classification, we use a set of training data consisting of input feature vectors (*xi*) and corresponding desired outputs (*yi*). The objective of classification is to create a hyperplane that can separate the data into two classes. There can be many hyperplanes that can do this, but the Support Vector Machine aims to find the hyperplane with the widest margin to achieve better performance. If the data cannot be separated linearly, then we need to use a nonlinear transformation to map the input space into a higher dimensional space [18].

*φ*(*x*) : *x* ⊂ *ℝ k* → *ℝ m, K << m* (2)

Vapnik proposed a family of functions that includes the hyperplane f(x) with the universal approximation form for hyperplanes that can be linearly separated.

*f* (*x*) = (3)

If our objective is to make the data linearly separable in the feature space, we should aim to find the hyperplane described below,

∀i: *yi = +1* (4)

∀i: *yi = -1* (5)

The points that are closest to the hyperplane are crucial in determining the class boundaries, which are also known as margins. The distance between the margins of the two classes is 2 divided by the magnitude of the hyperplane's weight vector (||w||). In order to obtain the best possible solution for the Support Vector Machine, various kernel functions are used, including the radial basis (RBF) kernel function which has been extensively studied [19][20][21].

In Support Vector Machine has two primary parameters that need to be selected properly for the RBF kernel function. One of these parameters is the penalty cost (C), which has a significant impact on classification accuracy. If the value of C is too high, the classification accuracy during the training phase will be very high, but the accuracy during the testing phase will be very low. On the other hand, if the value of C is too low, the classification accuracy will be unsatisfactory, and the model will become ambiguous. Various techniques are utilized in SVM to improve the possibility of the correct selection of parameter values. Some researchers suggest that if the objective of the model is to reduce the error rate of the classifier and the cost of measurement is uniform for all features, then the classifier should be designed to achieve maximum classification accuracy. The optimal features are then selected as the characteristics adopted by the classifier [22][23].

In our project, we have decided to use Support Vector Machines (SVMs) over the Random Forest algorithm as it provides us with improved accuracy.

*G. Confusion Matrix*

A confusion matrix is a popular machine-learning evaluation method for classification models. It presents several metrics that aid in understanding and analyzing the performance of a classification model on a given set of test data with known true or expected values. The matrix allows you to compare the model's predicted values to the actual target values. It provides a detailed overview of the model's performance and the types of errors it makes, making it a useful tool for summarizing a classification algorithm's performance [12.] Relying solely on the accuracy of a classification model can be misleading, especially when the dataset has an uneven distribution of observations across classes or has more than two classes. The confusion matrix provides the outcomes of the classifier, presenting information on the model's prediction and any classification errors made [13]. The features of a confusion matrix are asfollows:

* The matrix is structured according to the number of prediction classes, with a 2x2 table for two classes and a 3x3 table for three classes, and so on.
* It includes separate dimensions for predicted values, actual values, and the total number of predictions.
* The actual values represent the true values of the observations, while the predicted values are the values anticipated by the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual Class | |
| Positive (P) | Negative (N) |
| Predicted Class | Positive (P) | True Positive (TP) | False Positive (FP) |
| Negative (N) | False Negative (FN) | True Negative (TN) |

Fig 4: Confusion Matrix layout [13]

The terms Positive (P) and Negative (N) refer to the total number of cases in a dataset that is either positive or negative. The definitions of four terms commonly used in classification evaluations are as follows:

1. *True Positive (TP)*

When the predicted value matches the actual value, the model predicted a positive outcome, and the actual outcome was positive [13].

1. *True Negative (TN)*

This occurs when the predicted and actual values are both negative. The model predicted a negative outcome, which was confirmed by the actual outcome [13].

1. *False Positive (FP)- Type 1 error*

A Type 1 error occurs when the predicted value does not match the actual value. The model predicted a positive outcome, but the reality was different [13].

1. *False Negative (FP)- Type 2 error*

Also known as a Type 2 error, this occurs when the predicted value does not match the actual value. The model predicted a negative outcome, but the actual outcome was positive. These errors help to provide additional parameters for the prediction model [14].

* True Positive Rate (TPR)

The True Positive Rate (TPR) represents the likelihood or probability of correctly identifying actual positive cases in the data as positive, using a given test or model. In other words, TPR indicates how well the test or model is able to detect positive cases among all the positive cases present in the data[13].

(6)

* True Negative Rate (TNR)

The True Negative Rate (TNR), which is also known as Specificity, represents the probability or likelihood of correctly identifying actual negative cases in the data as negative, using a given test or model. TNR indicates how well the test or model is able to identify negative cases among all the negative cases present in the data[13].

(7)

* False Positive Rate (FPR)

The False Positive Rate is the probability of a positive outcome being indicated when the actual value is negative, resulting in a false alarm [13].

(8)

* False Negative Rate (FNR)

The False Negative Rate, which is also known as the "miss rate," represents the probability that a real positive result may be missed by the test [13].

(9)

* Negative Predictive Value (NPV)

The Negative Predictive Value is determined by dividing the number of true negative results by the total number of negative results and is expressed as a ratio [13].

(10)

* Positive Predictive Value (PPV)

The Positive Predictive Value, which is also known as "precision," is determined by dividing the number of true positive results by the total number of positive results and is expressed as a ratio [13].

(11)

* False Omission Rate (FOR)

The False Omission Rate is a measure of performance that indicates the probability of a true positive result being missed when a negative prediction is made [13].

(12)

* False Discovery Rate (FDR)

False Discovery Rate is determined by dividing the number of false positive results by the total number of positive results and is expressed as a ratio [13].

(13)

If we want to see the classification metrics like Precision, Recall and F1 score.

* F1 Score

The F1 score is a statistical metric that considers both precision and recall values, with an ideal value of 1 indicating a high-quality classifier[13].

(14)

* Accuracy (ACC)

Accuracy is determined by dividing the number of data sets that are correctly classified by the total number of observed data sets, and is expressed as a ratio [13].

(15)

1. ENVIRONMENT SETUP

Setting up the environment is the initial step when building or re-running a project on a different machine. This section explains the technical requirements for constructing or operating the project. Python was selected due to its multitude of frameworks, modules, and packages that can aid developers in saving time. The extensive technology stack of Python facilitates the provision of a broad range of libraries for computational intelligence. Object-oriented and functional programming are among the programming paradigms supported by Python.

* *Installing Python and Anaconda Platform*

The first step in the process is to install Python 3.8 version on the system. After that, the machine needs to have the Anaconda environment installed, which simplifies package management and deployment. The 'Individual Edition' of Anaconda has been installed for this purpose.

* *Packages Installation*

After successfully completing the previous step, the next task is to install individual packages required for the project. Anaconda provides a cloud-based repository to install various machine-learning packages. The installation of Anaconda includes Anaconda Prompt, a command-line interface that helps in installing the necessary packages.

To install packages required for the project, the 'conda' command is necessary. It can efficiently install, update, and execute packages along with their dependencies. The 'conda install' command is used to install any package. For this project, multiple packages including pandas, NumPy, matplotlib, sklearn, and panda were required, and all of them were installed using the 'conda install' command in Anaconda Prompt. Once all the necessary packages were installed, they were imported into the script as demonstrated below:

from os import listdir

from os.path import isdir, isfile, join

from posixpath import basename

import pandas as pd

import tensorflow as tf

import datetime

from tensorflow.keras.preprocessing import image\_dataset\_from\_directory

from sklearn.metrics import confusion\_matrix

* *Python Script*

To write Python scripts, an Integrated Development Environment (IDE) is needed. In this project, the Python script was written using Jupyter Notebook, which was set up through Anaconda. Jupyter Notebook is a very user-friendly web-based tool that can be launched by entering "jupyter notebook" in the Anaconda command prompt after it has been installed.

After starting Jupyter Notebook, a new Python file needs to be created to write scripts for a binary classification model. The models were trained, tested, and saved within the Jupyter Notebook once they were written.

1. IMPLEMENTATION

We were assigned a task, and we performed our actions accordingly. Initially, we familiarized ourselves with the measuring equipment and the method for collecting measurements. We installed the GUI\_V0.23\_2021-11-22 to measure and save the dataset. Then, we started collecting data by using different types of winter wear, with a dataset consisting of a minimum of 40 instances of staying in the car. Next, we adjusted the thresholds for the classification results. We started the experiments with X=80 and Y=1000, gradually increasing the X value by 10 and the Y value by 500 to obtain the desired results.

The same steps were followed for the dummy human and empty seats, resulting in at least 40,000 measurement scans recorded.

To achieve the project's objective, we collected readings and created a code that is divided into two sections. Firstly, we created a Confusion Matrix, and secondly, we deleted rows from the measurement files.

In order to perform the classification tasks, we needed to train the algorithms. The implemented classification was designed to distinguish between "Person Detected" and "Not Person Detected".

## Confusion Matrix Created

A widely used tool for evaluating the performance of machine learning classification is the Confusion Matrix. It is a table that illustrates how effectively a classification model performs on a set of test data when the real or predicted values are known. The matrix compares the predicted values of the machine learning model to the actual goal values. This allows us to have a comprehensive view of the model's performance and the types of errors it makes. We have created code for plotting the matrix, as shown in Figure 5.

**Fig.3 . Code snippet for plotting the matrix**

In this code, there are two possible values: 0, which indicates that no person has been detected in the office environment, and 1, which indicates that a person has been detected.

1. DISCUSSION AND CONCLUSION

In the paper, we proposed the method to classify time signals with SVM algorithm using ultrasonic sensor for capturing data. We trained the model with 2 known labeled objects as person and empty data, which has overall Number of Instances 67K which we acquired in 67 events. Each event contains 1000 measurements divided in:

* 45 events with Person.
* 22 events empty.

We trained our model with 80 percent of this instances. After performing training we evaluated the model. It achieves high evaluation score, accuracy 0.88, precision 0.91, recall 0.90, and F1-Score 0.90. The trained model is then used to evaluated by the remained dataset which is 20 percent of whole dataset.

Our accuracy was still not 100 percent which we think was as the result of training with unbalanced dataset. Our dataset with events with person were almost 2 times bigger than the dataset with empty seat. In the next experiment training the model on a balanced set of data could be a great approach.

1. COMMITMENT

The commitments of each member in this experiment can be found at [30]

1. "Machine Learning," [Online]. Available: https://en.wikipedia.org/wiki/Machine\_learning#Artificial\_intellig ence.
2. I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," Advances in Computational Approaches for Artificial Intelligent, Image Processing, IoT and Cloud Applications, 22 March 2021.
3. L. K. B. M. L. B. Busslinger A, "A comparative in vitro study of a magnetostrictive and a piezoelectric ultasonic scaling instrument," Periodontal & Implant Science, no. 28 (7), pp. 642 - 649, 2001.
4. M. R. M. K. R. M. H. Yousefimanesh, "A comparison of magnetostrictive and piezoelectric ultrasonic scaling devices: an in vitro study," Periodontal & Implant Science, no. 42 (6), pp. 243 - 247, 2012.
5. A. W. M. Toa, Ultrasonic Sensing Basics, Texas Instrument, 2020.
6. Turck Ultrasonic sensors, Catalog, 2007.
7. Red Pitaya, "https://www.mouser.in/new/red-pitaya/red-pitaya-starter-kit/", 2015
8. "Red pitaya," 2017. [Online]. Available: https://www.redpitaya.com/f145/specifications. [Accessed 03 05 2021].
9. C. Loan, Computational Frameworks for the Fast Fourier Transform, 1992.
10. E. K. L. M. F. P. B. T. Tim Menzies, "Chapter 24 - Using Goals in Model-Based Reasoning," in Sharing Data and Models in Software Engineering, 2015, pp. 321 - 353.
11. Ernest Yeboah Boateng, Joseph Otoo, Daniel "Abaye, Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review", Journal of Data Analysis and Information Processing, 2020
12. J. Brownlee, "What is a Confusion Matrix in Machine Learning," 18 November 2016. [Online]. Available: <https://machinelearningmastery.com/confusion-matrix-machinelearning/>.
13. "Confusion Matrix," [Online]. Available: <https://en.wikipedia.org/wiki/Confusion_matrix>.
14. M. Y Ting, Sammut, Claude; Webb, Geoffrey I. (eds.). Encyclopedia of machine learning., 2011
15. C.J.C. Burges, A tutorial on support vector machines for pattern recog- nition, Data Min. Knowl. Discov. 2 (1998) 121–167.
16. R. Burbidge, M. Trotter, B. Buxton, S. Holden, Drug design by machine learning: support vector machines for pharmaceutical data analysis, Comput. Chem. 26 (1998) 5–14.
17. R. Begg, D.T.H. Lai, M. Palaniswami, Computational Intelligence in Biomedical Engineering, CRC Press Taylor Francis Group, Boca Raton, FL, 2008.
18. V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 2000.
19. H.T. Lin, C.J. Lin, A Study on Sigmoid Kernels for SVM and the Training of Non-PSD Kernels by SMO-Type Methods, Technical Re- port, Department of Computer Science and Information Engineering, University of National, Taiwan (March 2003) 1–32.
20. K.R. Muller, S. Mike, G. Ratsch, K. Tsuda, B. Scholkopf, An introduc- tion to kernel-based learning algorithms, IEEE Trans. Neural Netw. 12 (2001) 181–201.
21. M. Pardo, G. Sberveglieri, Classification of electronic nose data with support vector machines, Sensors Actuators B Chem. 107 (2005) 730–737.
22. Y. Bazi, F. Melgani, Toward an optimal SVM classification system for hyperspectral remote sensing images, IEEE Trans. Geosci. Remote Sens. 44 (2006) 3374–3385.
23. A.P. Engelbrecht, Computational Intelligence, An Introduction, Second edi-tion, John Wiley Sons Ltd., England, 2007.
24. Available online: https://www.euroncap.com/en/vehicle-safety/the-ratings-explained/child-occupant-protection/ (accessed on 20 February 2023).
25. Automotive News Europe. Child-Detection Safety Technology May Get Mandate. Available online: https://europe.autonews.com/automakers/child-detection-safety-technology-may-get-mandate ((accessed on 22 February 2023).
26. Ismail, N.H.F.; Husain, N.A.; Mansor, M.S.F.; Baharuddin, M.M.; Zaki, N.M.; Husain, M.A.; Ma’aram, A.; Wiyono, A.S.; Chaiyakul, T.; Ahmad, Y. Child Presence Detection System and Technologies. J. Soc. Automot. Eng. Malays. 2019, 3, 290–297.
27. Chongpyo, C.; Gangchul, K.; Youngdug, P.; Wookhyun, L. The development of an energy-efficient heating system for electric vehicles. In Proceedings of the 2016 IEEE Transportation Electrification Conference and Expo, Asia—Pacific (ITEC), Busan, Korea, 1–4 June 2016; pp. 883–885.
28. Elbanhawi, M.; Simic, M.; Jazar, R. In the passenger seat: Investigating ride comfort measures in autonomous cars. IEEE Intell. Transp. Syst. Mag. 2015, 7, 4–17. [CrossRef]
29. Available online: <https://cerebrumx.ai/blogs/how-car-ultrasonic-sensor-data-could-change-automotive-iot/> (accessed on 22 February 2023).
30. <https://github.com/SusmitaSumaiya/SAML>