Detecting Interaction Failures through Emotional Feedback and Robot Context

Fernando Loureiro[1]

***Abstract*—During human-robot interactions, robots may break social norms (Social Norm Violations - SNV) or perform erroneous behaviours due to sensor and actuator errors, and software issues (Technical Failures - TF). If robots are unaware of these errors, the interaction may become unpleasant or even risk user safety. While interacting, humans show various types of social signals that translate their inner state, which is concurrently estimated by other humans that detect social norm violations and react to them. To detect social errors and classify them as Social Norm Violations or Technical Failures, we propose to rely on Eye Gaze, Head Movement, Facial Expressions (Actions Units), and Emotions, as seen by the robot, along with the recent actions of the robot. We propose a two step cascaded decision, where the first step is to detect if an error occurs, followed by the error type classification (SNV vs. TF). We perform an extensive study of the various options on input data and classification algorithms, using a game-based scenario with a humanoid robot. We focus on Vizzy robot and in a dataset where Vizzy individually interacted with 24 participants in a block assembly game, where it had two moods. The "good" mood would help the participants win the game. The "bad" mood would be rude, causing social norm violations, and would clumsily destroy the assembled blocks, causing technical failures, and making the participant lose the game. Regarding the impact of input data, we observe that: (i) emotions improve the error detection step but not the error classification step, and (ii) the actions of the robot improves both error detection and error classification. Regarding the learning algorithms, Random Forest achieves the best performance both in error detection and error classification. The usage of the median filter on the error classification result increased the performance of Random Forest to 79.63% mean accuracy.**

***Index Terms*—Social Signals, Human Robot Interaction, Error Detection, Error Classification**

I. Introduction

Robots are becoming part of our daily life and will interact with us in many tasks. However, interaction failures can happen, either caused by a malfunction, Technical Failure (TF), or by a misunderstanding of the social conduct by the robot, Social Norm Violation (SNV) [1]. Robot mistakes lead to a loss of user trust [2] and can endanger humans. Thus, robots must have the ability to verify and correct their actions. Our goal is to build an automatic error detector and classifier that interprets human reactions, allowing robots to understand if something went wrong during interactions. We focus on signals that a mobile social robot can capture using its onboard sensors. Eye gaze, head movement, and facial expressions (Action Units (AU)) are the most relevant signals supported by the literature (section II). We will also study whether emotions and information of the recent actions of the robot (context) are informative features that improve the proposed error detector and classifier.

Fig. 1: Vizzy interacting during a cognitive board-game

We focus on one-to-one human-robot interactions during zero-acquaintance encounters. We will be using the dataset obtained by Avelino et al. [3] for our experiments.

The remainder of this paper is structured as follows. In section II we briefly overview past works on automatic detection of human-robot interaction failures, as well as works that analyse relevant features for this task. Section III presents our methodology and the proposed pipeline for automatic error detection and classification. Then, section IV sets up the experiments to evaluate the proposed algorithm. It describes the tests to compare our solution against the baseline and to analyse the contribution of distinct features and classifiers. Afterward, we analyse the results in section V, finishing with conclusions in section VI.

II. Related Work

A. Human-Robot Interaction errors

Several studies of how people react when dealing with robots have emerged in the last decade, more specifically, when dealing with erroneous robots [1]–[8]. These studies analysed the different modalities of reactions of people, during experiments where they had to perform a certain task, such as cooking or building something [4]–[6]. They identified that the most relevant signals emitted by the participants during the experiments were gaze shifting, head movements, facial expressions, and speech.

Emotions are part of both human-human and human-robot interactions. Some studies noticed how people changed their mood/emotions throughout the experiments with erroneous robots, [6], [5], however, as far as we know, no work has yet used emotions for error detection. As such, we decided to analyse and use emotion recognition for our error detection problem.

Most emotion studies consider six basic classes: happiness,

[1] Instituto Superior Técnico, Universidade de Lisboa
fernando.a.a.d.r.loureiro@tecnico.ulisboa.pt

Special thanks to the Family Soares dos Santos Scholarships, without whom this work would have not been possible.

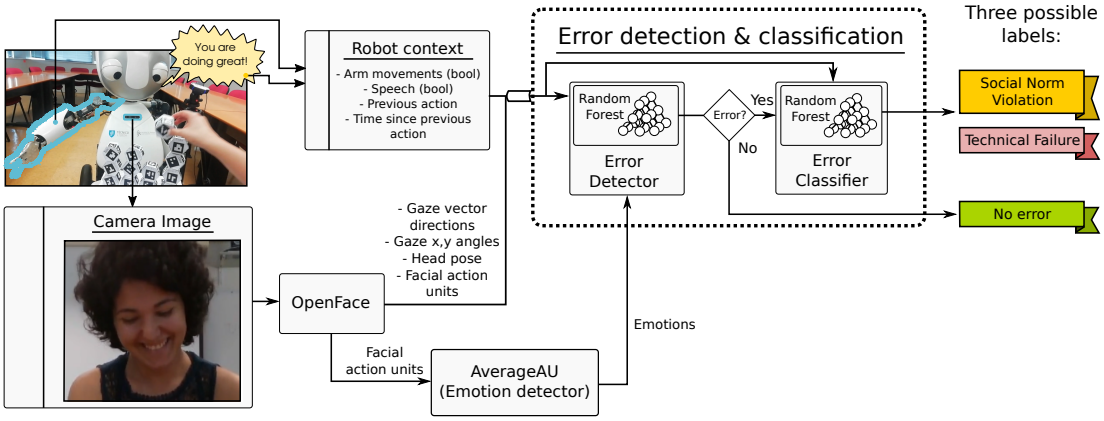


Fig. 2: Proposed System

anger, disgust, fear, sadness, and surprise, [9], and the main features used for the recognition from images are the facial expressions, [10], [11]. Facial expressions can also be translated into action units - AU, actions of group or individual facial muscles, and some studies aim to associate specific action units to a corresponding emotion, more specifically the six basic ones [12], [13], [14], [15]. We intend to explore the usage of AU to obtain emotions, to have a simple and efficient emotion recognition algorithm.

B. Automatic detection of interaction errors

Kontogiorgos et al. [7] used a Random Forest (RF) with gaze, head movement, and speech as features for error detection, and concluded that head movements and gaze were the most relevant features when a user is dealing with a humanoid robot. Furthermore, they highlight the importance of contextualization when assessing the response of the participants to robot failures. Trung et al. [8] used Naive Bayes (NB), and K-Nearest Neighbour (KNN) classifier with head and shoulders 3D position, for error detection and classification. They noticed that the KNN achieve high scores when having already some examples from the user. On the other hand, NB dealt better with new participants. They advise splitting the classification into two steps, first identify if an error had occurred, and then classify it as a SNV or TF.
Building upon these works, we are going to use features that were mentioned as relevant by the state-of-the-art interaction studies, but, as far as we know, have not yet been used for error automatic detection, such as facial expressions, emotions, and context of the interaction in the form of the most recent actions of the robot.

III. Methodology

A. Proposed algorithm

Our main goal is to build an algorithm to detect error situations and classify them as SNV or TF. We propose the pipeline in Figure 2. Our algorithm detects and classifies errors frame by frame and following Trung et. al. [8], does it in two steps. First, a Random Forest error detector uses robot context features, gaze features, head pose, facial action units, and emotions. If an error is detected, an error classifier, which is also a Random Forest model, uses all the previous signals except emotions to classify it as SNV or TF. Otherwise, the algorithm outputs the ”No error” label. A median filter is also used on the error detector. With the filter we can make sure that miss-classified errors or no error frames can be corrected. Our shortest reaction has the duration of around 2 seconds, as such we decided that the window of the filter is about 30 frames, which is equivalent to one second.

B. Feature extraction

To obtain eye gaze, head, and facial information from the participants we used OpenFace [16], an open-source facial behaviour analysis tool capable of facial landmark detection, head pose estimation, facial Action Units recognition.

To detect emotions, we propose a method that uses facial action units [15]. This way, we can use OpenFace to compute all head and face signals, with reduced computational requirements, since there is no need for an additional machine learning algorithm to obtain emotions. More specifically, we average the specific action units for each emotion, defined by Ekman et al [15], and select the emotion with the highest value. The neutral emotion is selected if the highest value is not above a previously defined threshold. We call this method AverageAU or avgAU.

The actions of the robot consist of the current action, which consists of movement (Boolean) and speech (Boolean), last performed action, which can be move, speech, or move&speech. And time since the last action, which is measured in seconds.

IV. Experiment Setup

In this section, we describe a set of experiments to evaluate the proposed solution. To do so, we compare its performance against distinct combinations of classifiers and input features, where we use the accuracy and F1 score and check for statistically significant differences.
To compare the proposed algorithm with other algorithms, we use the Wilcoxon test [17], a non-parametric test that does not assume any properties regarding the distribution of the

variables in analysis, and also the Student's t-test [18]. The Wilcoxon test is used, over the t-test, when the distribution of the difference between the means of two samples cannot be assumed to be normally distributed. To test for normality, we used the Shapiro test [19]. These hypothesis tests will tell us if there is a statistically significant difference between the algorithms: if p-value is smaller than 0.05, then there is a statistically significant difference between the algorithms. Students t-test is said to be more reliable than Wilcoxon test when the assumption that the data has a normal distribution is assured [20].
Size effect is also used in some experiments when a statistical significant difference is achieved to evaluate the magnitude of the difference. We use the Cohen's d size effect [21]. If the d is below 0.2, then the size effect is small, if it is between 0.2 and 0.8 is considered medium, above 0.8 is considered large.

These experiments also allow us to study the impact of the input features in the overall performance of the algorithm, allowing us to test the following hypotheses:

Hypothesis 1 (H1): *Adding Facial Action Units to the literature base feature vector (head, gaze) will significantly improve **a)** error detection and **b)** error classification.*

Hypothesis 2 (H2): *Information of the current action of the robot significantly improves **a)** error detection and **b)** error classification.*

Hypothesis 3 (H3): *Temporal information of the actions of the robot significantly improves **a)**error detection and **b)** error classification.*

Hypothesis 4 (H4): *Emotion information of the user significantly improves **a)** error detection and **b)** error classification.*

In addition, we also perform a set of experiments to evaluate the proposed emotion detector, averageAU, and validate its use over alternative methods. Due to the widespread use of facial half masks due to the COVID-19 pandemic, we also test the emotion detector algorithm in masked faces.

A. Dataset

We use the dataset of Avelino et al. [3], obtained in human-robot interaction experiments with the social robot Vizzy [22]. The dataset consists of an experiment where 24 participants individually interacted with Vizzy in a block assembly game. For 11 of those participants, the robot was programmed to generate both SNV (arrogant and grumpy behaviour) and TF (destroy the block assembly). For everyone, a video was captured from a camera on the robot.
To annotate the error and error type of the dataset, we analysed the reactions of the users, and labelled the beginning of an error as soon as we detected a reaction and the end of the error when the reaction started to fade. Errors were classified as SNV or TF depending on the action of the robot. In the dataset of [3], the majority of speak actions were SNV, while most move actions were TF. There were also some speak actions where the voice of Vizzy failed a bit and confused the participants, those were considered as TF. On figure 3 we show the annotation of error and error type of a video from Avelino dataset, where we note that the dataset is unbalanced, since the number of no error frames is significantly higher than the number of error frames (SNV and TF).

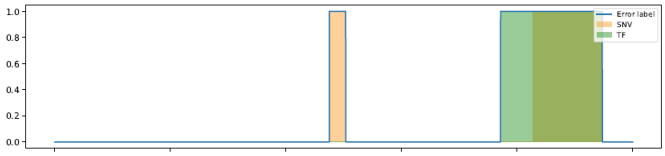


Fig. 3: Error and error type annotation, blue line is the error (bool), orange shade is Social Norm Violation, green shade is Technical Failure

B. Proposed error detector and classifier

We start by showing the results of our proposed solution, which is using Random Forest with head, gaze, AU, emotions and actions of the robot, fig. 2, and compare it to the features used in previous automatic error detector works, [7] [8], which used head and gaze features. First, we detect if an error has occurred with the error detector, if so, then the error type classifier is used. Furthermore, we compare the usage of the proposed algorithm with and without a median filter on the error detector.
This is a multi-label problem with Error, SNV, and TF labels, and tested in an imbalanced dataset, so balanced accuracy and hamming loss are used for the experiments.

C. Error detection

Based on the previous automatic error detector works [7] [8], we compare RF with NB, and KNN in our problem. As planned, we start with the detection of the error, and then the classification of the error in SNV or TF.
An initial experiment was performed to understand which method to use for the balancing and splitting of the data. Regarding the balancing method, we tested under-sampling, and over-sampling, where we over-sampled the true error frames, by horizontally flipping the video recordings of the dataset.
For the splitting of the data, we used the train_test_split method from sklearn[1], and a random selector of 25% of the videos of the dataset of Avelino et al. for the test set.

1) Naive Bayes Vs Random Forest Vs KNN: Random Forest is compared with Naive Bayes and K-nearest Neighbour, used by Trung et al. [8], all tuned with the proposed combination of features.

2) Outlier detection: The response from the participants can be considered as a deviation from the regular behaviour. In figure 4, we present a graph showing the distribution of the percentage of each emotion for error and no error cases. We can see that there is a spike of happiness. With this in mind, we decided to try to use outlier detection algorithms to

[1]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

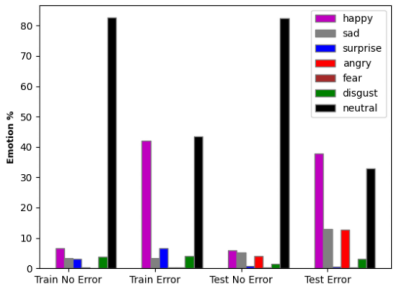


Fig. 4: Distribution of emotions on No error and error situations during training and testing set

identify these mistakes and compare them to Random Forest. We focused our attention to outlier detector methods that are frequently used by the community: Isolation Forest [23], Local Outlier Factor (LOF) [24] and Minimum Covariance determinant (Elliptic Envelope) [25].

For these experiments, we used all the dataset, which is imbalanced, and as such used balanced accuracy and f1 score. Domingues et al. [26] performed a comparison study on various outlier detection algorithms, where they concluded that Isolation Forest achieves better results in efficiently identifying outliers while showing excellent scalability on large datasets. Local Outlier Factor (LOF) reached the lowest performance.

3) Different combination of input features: In this section we show the experiments where we compare the different combination of input features to evaluate the impact they have in the system.

We compare the addition of the action units, to the head and gaze features, that were used in previous automatic error works. These three features together form what we usually call the base features, since these are the features that were classified as the most relevant for error detection in the state-of-the-art studies. Then, we compare the addition of emotions and actions individually, and finally add both.

Finally, we reach our proposed combination where we add temporal information to the actions of the robot. More specifically, besides having the current action, we added the last action performed and the time since the last action. The idea is that some reactions have a delay, so they happen when the action of the robot is over, by having the information of the last action as well as the time since it happened, we provide the algorithm with a more detailed situation.

An additional experiment was performed to verify if the remaining features provided by openFace would significantly improve our algorithm. For the facial expression, we used the action units provided by openFace, however, it does also output[2] landmark information regarding the eye region, face, and parameters of the rigid face and non-rigid face.

We also test how the error detector behaves only with features that can be captured from a user that is using a half-face mask.

4) Usage of Emotions on Error detector: A further experiment to compare features was done, where we verify if the addition of emotions and/or actions helps or not the algorithm. Since what we use is a random selector of the videos that goes for the test set, in each run that we did, different videos were used for training. For this experiment, we decided to perform 10 sets of experiments, wherein each we would perform 25 runs. In each set, we obtained the number of runs that the combination that achieved better results, which was chosen as the one with the highest score when the McNemar test was below 0.05. The McNemar hypothesis test tells us that one algorithm makes more mistakes than the other, if the p-value is below 0.05, and if it is above then the algorithms fail similarly. For this experiment, we used the students t-test, when the data follows a normal distribution, and the Wilcoxon test to evaluate each set, and finally the entire 10 sets of experiments. We perform these two hypothesis tests because it is said that the t-test is more reliable than the Wilcoxon test, when the assumption that the data has a normal distribution is assured [20], which we verify by using the Shapiro test, so we decided to explore this in this experiment.

[2]https://github.com/TadasBaltrusaitis/OpenFace/wiki/Output-Format

D. Error Type classifier

The classification of the error as a TF or/and SNV is a multi-label classification problem, since at a certain time both types of errors could have happened, and as such, an instance can be assigned with both. For these experiments, since we are dealing with a multi-label problem, we use accuracy and hamming loss.

We start by comparing Random Forest with Naive Bayes, and K-Nearest Neighbour. Then, we test different combinations of features. A similar experiment performed in the error detector is also conducted to verify if the addition of emotions and/or actions improves the algorithm.

E. Emotion recognition

To evaluate the proposed emotion recognition method, averageAU, we perform three experiments. First, we compare the combination of AU proposed by Ekman et al. [15] with other combinations proposed by Ghayoumi et al. [12], Lucey et al. [14] and Karthick et al. [13], as well we test several thresholds for the neutral emotion. In the second experiment, we compare AverageAU with DeepFace [27] and Efficient CNN [28]. In the third, we test the applicability of our algorithm when dealing with people using half-face masks.

1) AU combinations: First, we try several combinations of AU for each emotion, proposed by various works (see table I), as well as various thresholds for the neutral emotion. A new combination is also proposed, which we called mixedBest, which usages the best combination for each emotion achieved by the methods of the other works, according to the experiments performed. The experiment uses the CK+ dataset [14].

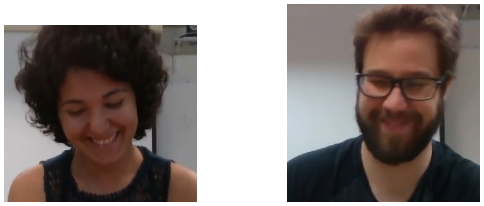
2) Emotion recognition evaluation: We compare our proposed method, AverageAU, with two already built emotion recognition algorithms, DeepFace [27] and Efficient CNN [28]. To compare the three emotion algorithms, we used two representative cases of the dataset of Avelino et al. [3]. A girl

TABLE I: Correspondence of AUs to emotions

Method	Anger	Disgust	Fear	Happy	Sad	Surprise
Tautkute et al. [29]	4, 5, 7, 23	9, 15, 16	1, 2, 4, 5, 7, 20, 26	6, 12	1, 4 15	1, 2, 5, 26
Ghayoumi et al. [12]	2, 4, 7, 9, 10, 20, 26	2, 4, 9, 15, 17	1, 2, 4, 5, 15, 20, 26	1, 6, 12, 14	1, 4, 15, 23	1, 2, 5, 15, 16, 20, 26
Ekman et al. [15]	4, 5, 7, 23	9, 15, 16	1, 2, 4, 5, 20, 26	6, 12	1, 4, 15	1, 2, 5, 26
Lucey et al. [14]	4, 5, 15, 17	1, 4, 15, 17	1, 4, 7, 20	6, 12, 25	1, 2, 4, 15, 17	1, 2, 5, 25, 27
Karthick et al. [13]	4, 5, 7, 23, 24	9, 17	1, 4, 5, 7	6, 12, 25	1, 4, 15, 17	1, 2, 5, 26, 27

(Data1) whose facial expressions are visible, and a boy with a beard (Data7), (see fig. 5). These cases were chosen because they are representative of unambiguous (Data1) and harder to notice emotions (Data7).

An additional experiment is performed where we test the



(a) Data1 sample (b) Data7 sample

Fig. 5: Avelino et al [3] dataset images

algorithms using the CK+ dataset.

3) Emotion Algorithm on faces with half masks: To understand if it is possible to acquire emotions when people are using half-face masks, using the CK+ dataset, we remove the AU that are covered by the mask, and use the action units of the upper part of the face.

V. Results

In this section, we present and discuss the results obtained throughout our work.

A. Proposed pipeline evaluation

We show the results of the proposed error detector and classifier method and compare it the results of Random Forest using head and gaze features, Fig. 6. The statistically significant difference is represented with: * == p<0.05, ** == p<0.01, *** == p<0.001, **** == p<0.0001.

As we can see, the proposed error detector and classifier algorithm achieves a higher accuracy score of 72.77% while the method that only uses head and gaze features achieved 57.21% with a statistically significant difference. Our method also achieves a lower hamming loss than the other algorithm. Cohen's d size effect was also used, achieving a large size effect of 5.24, meaning that the difference achieved has a large magnitude.

As such, our solution achieves better results in detecting and classifying an error than the one using only head and gaze features, features used in previous error detector works.

On figure 7 we show the results of our proposed method with and without a median filter on the error detector. The addition of the median filter increased the performance of the algorithm from 72.26% to 79.63% average accuracy, with a large size effect of 2.27.

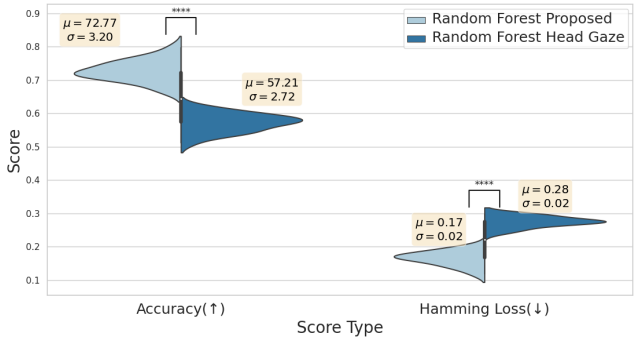


Fig. 6: Comparing the proposed algorithm with the features used in previous works. ↑ - higher scores are better; ↓ - lower scores are better

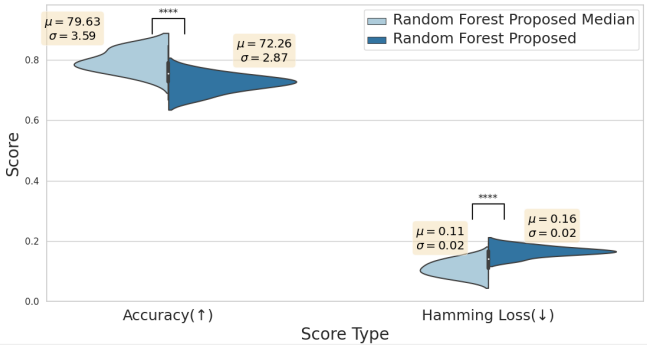


Fig. 7: Comparing the proposed algorithm with and without median filter. ↑ - higher scores are better; ↓ - lower scores are better

B. Error detector

In this section, we show the results from the experiments related to the error detector.

Regarding the balancing methods, both under-sampling and over-sampling achieved similar results in terms of accuracy and F1-score, so the following experiments were done using the data augmentation method. As for the splitting methods, the one from sklearn achieved results of around 98% mean accuracy and F1-score, and the random selector achieved results between 72% and 75%. The reason that the method from sklearn achieves such high results in our case, is probably since distinct samples of the same video may appear during training and testing, while with the random selector we make sure that the participants in the testing set were never seen before.

With this experiment, we may hypothesize that the algorithm can detect error situations correctly if it has dealt with the person before. Nonetheless, the following experiments are

done using the random selector.

1) Naive Bayes Vs Random Forest Vs KNN: On figure 8, we show a violin plot with the results of the comparison between the three classifiers.

Regarding KNN, it proved to be computationally costly for our algorithm. Random Forest took around 10 seconds to fit and predict the entire dataset, while KNN took around 100 seconds. Nevertheless, Random Forest achieved the best results, being statistically significantly different according to the Wilcoxon test from Naive Bayes and K-Nearest Neighbour.

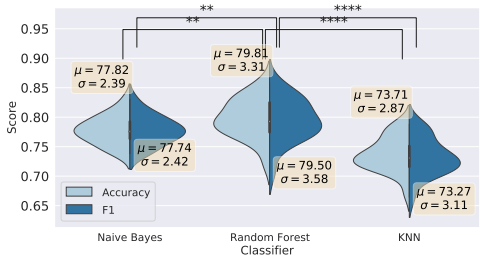


Fig. 8: Random Forest Vs Naive Bayes Vs KNN

We can then conclude that for our problem Random Forest performs the best, out of all three methods.

2) Outlier Detection: In table II we show the results when comparing the different outlier detectors. Isolation Forest was the outlier detector with the highest score, with a statistically significant difference from the other two methods, as well as the most computationally efficient, being the fastest on our dataset.

TABLE II: Comparing different outlier detectors

Outlier Detector	Balanced Accuracy		F1 score	
	Mean	SD	Mean	SD
Isolation Forest(1)	74.51	2.64	50.28	4.54
Local Outlier Factor(2)	53.31	3.68	23.70	3.40
Elliptic Envelope(3)	69.43	5.65	41.54	7.05

Features	Hypothesis
	p-value
(1) Vs (2)	5.96e-8
(1) Vs (3)	1.49e-6
(2) Vs (3)	5.96e-8

In the next experiment, we compare Random Forest with Isolation Forest. Random Forest performed the best, table III, achieving 79.75% balanced accuracy with a statistically significant difference from Isolation Forest with a balanced accuracy of 74.74%. Cohen's d size effect achieved a value above 0.8, meaning that Random Forest achieves a statistically significant different result from Isolation Forest, with a large size effect.

TABLE III: Random Forest Vs Isolation Forest

Algorithm	Balanced Accuracy		F1		Hypothesis test	Size Effect
	mean	SD	mean	SD	p-value	Cohen's d
Random Forest	79.75	3.49	61.57	3.88	6.10e-5	1.47
Isolation Forest	74.74	3.36	51.84	4.35		

Even though Isolation Forest achieves lower scores over Random Forest, as an outlier detection method it has an advantage over Random Forest, that it should be capable of detecting new types of reactions to mistakes, that Random Forest has not seen/trained yet. In figure 9 we show such example with a violin plot, where we removed the TF frames from the training set of Random Forest, and see that in this case, Isolation Forest outperforms the previous method.

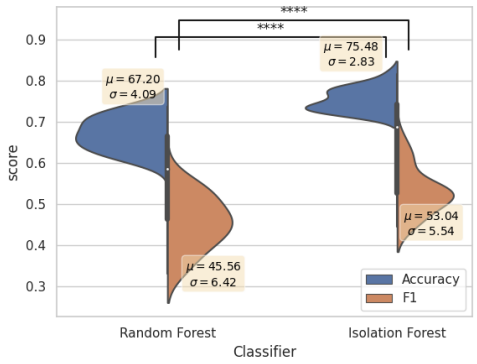


Fig. 9: Remove TF error, * == p<0.05, ** == p<0.01, *** == p<0.001, **** == p<0.0001

However, this advantage of the outlier detector when dealing with new errors can also be a disadvantage. During the experiments, we noticed that the Isolation Forest does not deal as well as Random Forest with interactions where no error occurs. From figure 10 we can see that Random Forest achieves a statistically significantly higher score than Isolation Forest when the test set only has no error situations.

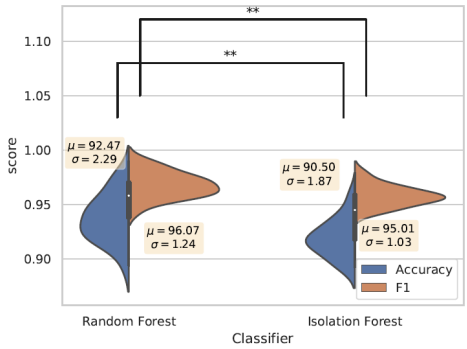


Fig. 10: No errors, * == p<0.05, ** == p<0.01, *** == p<0.001, **** == p<0.0001

So, for the error detector, the results confirm that Random Forest, as proposed, achieved better results.

3) Different combination of input features: We proceed to check how the AU influences the algorithm, table IV. From the table, we conclude that the addition of AU improved the algorithm in detecting an error, with a statistically significant difference. With this we verify the hypothesis **H1a)**.

TABLE IV: Addition of action units

Features	Accuracy		Wilcoxon		F1		Wilcoxon	
	Mean	SD	p	stat	Mean	SD	p	stat
Head, Gaze	58.96	3.64	1.73e-6	0.0	53.80	6.01	1.92e-6	1.0
Head, Gaze, AU	70.63	6.00			68.40	7.89		

In table V we can see that, while the addition of emotions did not achieve a statistically significant different performance,

p-value above 0.05, the addition of action as well the addition of both the previous features, caused an improvement of the algorithm, when comparing with the base features, with a statistically significant difference.
From this experiment, we can conclude then that the addition of actions of the robot help Random Forest in detecting error situations, which confirms hypothesis **H2a)**.

TABLE V: Comparing different combination of features

Features	Accuracy		Wilcoxon		F1		Wilcoxon	
	Mean	SD	p	stat	Mean	SD	p	stat
Base	72.67	5.34			70.96	6.65		
+Emotion	72.62	5.34	0.7	214	70.90	6.66	0.7	214
+Action	72.94	5.28	0.017	166	71.27	6.57	0.018	168
+Action +Emotion	72.79	5.25	0.035	187	71.11	6.54	0.030	182

The experiments performed so far were done using data obtained from the camera in Vizzy. However, during the experiments of Avelino et al. [3] a laptop was also present. This laptop was used to show the game score to the participant, and was also an interaction point of the experiment, even so, that some participants reacted to the mistakes of Vizzy facing the laptop. As such, we decided to perform the previous experiments but now adding the data obtained from the laptop point of view. In table VI we present the results, and we can see that with the addition of the laptop view, the addition of emotions and the addition of both new features, achieve a statistical significance difference, improving the algorithm.

TABLE VI: Comparing different combination of features, laptop and Vizzy view

Features	Accuracy		Wilcoxon		F1		Wilcoxon	
	Mean	SD	p	stat	Mean	SD	p	stat
Base	70.01	3.02			67.95	4.01		
+ Actions	70.19	3.05	0.098	152	68.14	4.04	0.15	163
+ Emotions	70.56	2.88	0.003	90	68.63	3.78	0.0028	87
+ Actions + Emotions	70.47	2.84	0.0023	84	68.50	3.73	0.0024	85

The main goal of this work is to analyse data of people interacting with social robots, in our case Vizzy. As such, the following experiments will focus on the usage of the Vizzy dataset angle alone, and the usage of the Vizzy dataset with the Laptop angle as an addition.

In table VII we show the results of the experiment to add temporal information. We first compared the addition of the last action to the previous features, achieving better performance with a p-value below 0.05. Then, we added the time since the last action which also provided a statistically significant different performance. In the last row, we test if the time since the last action achieved a significantly different result from adding only the last action, which it did.
As such, we conclude that the addition of temporal features improves the algorithm, verifying hypothesis **H3a)**.

The algorithm using all features (head, gaze, AU, emotions, actions) plus the landmarks achieved a mean accuracy of 69.34%, table VIII, lower than the usage of the base plus actions and base plus actions and emotions, and is also statistically significantly different, a p-value below 0.05. Besides not being as efficient as the other algorithms, it is also more

TABLE VII: Addition of temporal actions

Features	Accuracy		Wilcoxon		F1		Wilcoxon	
	Mean	SD	p	stat	Mean	SD	p	stat
+ Actions + Emotions(1)	73.43	6.61			71.84	8.04		
(1) + lastAction (2)	74.71	6.04	2.15e-5	21	73.38	7.24	2.16e-5	26
(2) + t_lastAction (3)	75.85	5.80	1.24e-5	20	74.62	6.93	1.97e-5	25
(2) Vs (3)			0.00066	67			0.0014	77

computationally expensive since it increased the number of features from around 60 to 700, making it impractical to use in real-time.

TABLE VIII: Random Forest error detector, with eye and facial landmarks

Features	Accuracy		F1 score	
	Mean	SD	Mean	SD
All + Landmarks (1)	69.34	3.88	67.88	4.82
+Actions(2)	75.78	3.16	75.20	3.66
+Action +Emotions(3)	75.96	3.23	75.40	3.72

Features	Wilcoxon	
	p	stat
(1) Vs (2)	1.73e-6	0.0
(1) Vs (3)	1.73e-6	0.0
(2) Vs (3)	0.086	138.0

4) Error detector for people with mask: In table IX we show the results from the experiment, where the use of Gaze, Head, and Actions achieved a mean accuracy of 72.18%, a lower score when comparing with the use of the total features used by the algorithm, (2) and (3) from the table, but still an efficient algorithm.

TABLE IX: Error Detector with Features ready to deal with people with masks

Features	Accuracy		F1	
	Mean	SD	Mean	SD
Gaze, Head, Actions (1)	72.18	4.25	71.39	4.73
Base +Actions(2)	79.86	3.06	79.57	3.32
Base + Actions +Emotions(3)	80.13	2.88	79.86	3.10

Features	Wilcoxon Accuracy	
	p	stat
(1) Vs (2)	1.73e-6	0.0
(1) Vs (3)	1.73e-6	0.0
(2) Vs (3)	0.026	124.0

5) Usage of Emotions on Error Detector: Regarding the usage of features, with the previous experiments, we could conclude that the usage of action helps the algorithm. However, when comparing the addition of emotions to the base plus actions, generally a statistically significant difference is not achieved. But in certain runs, a statistically significant difference was obtained.
The results are shown in table X and from it, we can see that in each run, base plus actions plus emotions generally has the highest number of runs where it is best. Looking at the 10 sets, in four of those both Wilcoxon and t-test had values below 0.05, with the usage of emotions and actions achieving the highest value, the rest 6 sets no statistically significant difference was achieved. When evaluating all the mean values from the 10 sets, both hypothesis tests achieved p-values below 0.05, with the overall highest mean achieved by base plus actions plus emotions, this verifies the hypothesis **H4a)**.

So, we can conclude that using Random Forest with action units, head movement, and position, gaze shifting, actions of the robot and emotions achieves the best performance in detecting an error in our dataset, which confirms all the hypothesis written before **H1 - H4**.

TABLE X: Comparison between combination of features, with Vizzy dataset

Set of 25 runs	Actions + Emotions (1) Score			Actions (2) score			Wilcoxon	t-test	hypothesis test for all 250 runs
	mean	SD	Runs	mean	SD	Runs	p-value	p-value	
1	79.99	3.21	13	79.77	3.41	3	0.11	0.30	
2	79.28	3.36	9	79.10	3.64	3	0.12	0.16	
3	**81.00**	3.58	10	80.61	3.73	4	**0.0094**	**0.0091**	wilcoxon: 0.027
4	79.63	4.01	9	79.58	4.26	9	0.69	0.78	t-test: 0.016
5	**79.74**	3.09	9	79.27	3.35	4	**0.0027**	**0.0045**	(1): 79.99
6	**79.35**	3.33	11	79.11	3.45	4	**0.03**	**0.045**	(2): 79.80
7	80.15	3.11	13	80.06	3.48	9	0.38	0.66	
8	**78.79**	3.19	17	78.44	3.51	3	**0.022**	**0.043**	
9	81.95	3.19	8	82.18	3.40	8	0.34	0.17	
10	79.99	3.07	16	79.89	3.41	7	0.31	0.54	

C. Error Type Classifier

In this section we present the results for the error type classifier

1) Random Forest Vs Naive Bayes Vs KNN: We compare Naive Bayes, and K-Nearest Neighbour with Random Forest, using the base plus actions features, figure 11.
Random Forest achieved the highest scores on accuracy mean score, and the lowest on hamming loss, with a statistically significant different result from the other two algorithms.

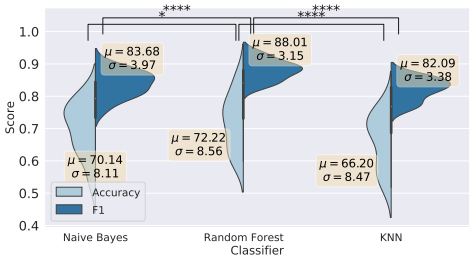


Fig. 11: Random Forest Vs Naive Bayes Vs KNN

We can reach the conclusion that Random Forest achieves the best performance for our error type classifier.

2) Different combination of input features: In table XI we verify that the addition of the action units improved the algorithm when comparing with the features used by the previous automatic error detector studies ([8], [7]). This confirms the hypothesis **H1b)**.

TABLE XI: Compare the addition of Action Units, Vizzy angle

Features	Accuracy		Hamming Loss	
	Mean	SD	Mean	SD
Head and Gaze (1)	59.90	9.80	0.27	0.073
Head, Gaze and AU(2)	64.11	10.0	0.24	0.073

Features	Wilcoxon	
	p	stat
(1) Vs (2)	0.0001	5.0

In table XII, all the additions to the base features achieved a statistically significant difference with higher accuracy and lower hamming loss. The addition of actions increased the most performance of the algorithm.
We can conclude then, that the actions of the robot help the algorithm in classifying the type of error, confirming the hypothesis **H2b)** and **H3b)**.

3) Usage of Emotions for Error Classifier: Throughout the error classifier experiments, there was not a clear observation if the addition of emotion to the actions improved the algorithm. So, in this section, we present an experiment like the one before, where we perform 10 set experiments, each constituted

TABLE XII: Random Forest classify error type, combination of features

Features	Accuracy		Hamming Loss	
	Mean	SD	Mean	SD
Base (1)	62.23	9.13	0.244	0.068
+Actions(2)	74.56	8.47	0.136	0.045
+Emotions(3)	63.09	9.35	0.239	0.068
+Action +Emotions(4)	74.63	8.73	0.136	0.046

Features	Wilcoxon	
	p	stat
(1) Vs (2)	8.86e-5	0.0
(1) Vs (3)	0.033	48.0
(1) Vs (4)	8.86e-5	0.0
(2) Vs (3)	8.86e-5	0.0
(2) Vs (4)	0.68	94.0
(3) Vs (4)	8.86e-5	0.0

by 25 runs. As we can see from table XIII, the usage of base plus actions more frequently achieves the highest accuracy scores with a statistically significant difference. The overall hypothesis test also achieved a p-value below 0.05, with the best performance achieved by the base plus actions of 75.48% accuracy. This negates the hypothesis **H4b)**.

TABLE XIII: base + actions Vs base + actions + emotions on Vizzy angle

Set of 25 runs	Actions + Emotions(1) score	Actions (2) score	Wilcoxon	t-test	hypothesis test for all 250 runs
	Accuracy	Accuracy			
	mean (SD)	mean (SD)	p-value	p-value	
1	74.58 (7.98)	**75.43 (7.90)**	**0.00055**	**0.00058**	
2	73.79 (8.62)	**74.20 (8.54)**	**0.027**	**0.016**	wilcoxon: 0.0019
3	75.52 (7.60)	**76.15 (7.58)**	**0.01**	**0.008**	t-test: 0.014
4	74.12 (10.99)	74.47 (10.96)	0.28	0.22	(1): 74.65
5	72.18 (9.55)	**72.93 (9.69)**	**0.0009**	**0.0002**	(2): 75.48
6	76.05 (8.28)	**76.86 (7.96)**	**0.0002**	**0.0003**	
7	73.28 (8.33)	73.51 (9.07)	0.24	0.44	
8	77.02 (9.11)	**77.80 (9.19)**	**0.0037**	**0.0079**	
9	75.82 (6.93)	**79.03 (6.94)**	**0.011**	**0.009**	
10	74.20 (9.04)	74.46 (9.43)	0.29	0.30	

In conclusion, to classify the error into TF and SNV, the usage of head, gaze, action units and actions performs the best in our dataset. This confirms all the hypothesis written before, except for the **H4b)** since emotions does not help in classifying an error.

D. Emotion Algorithm

In this section we show the results for the emotion algorithm.

1) AU combination: For all combination sets, having a threshold below 0.5 demonstrated poor performance in detecting neutral faces. With thresholds above 1, the methods considered most emotions as neutral. As such we decided to compare the methods with a threshold of 0.8.
From table XIV we notice that the combinations of Ghayoumi et al. and Lucey et al. have an overall worst performance. Tautkute et al. and Karthick et al. achieved the worst in detecting fear. MixedBest has a poor performance in detecting anger. Ekman et al., the proposed combination, achieved high accuracy with the lowest being on fear, nonetheless is the combination that offers the most balanced scores.

TABLE XIV: Comparing averageAU methods with threshold 0.8

Method(%)	Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral	Overall
[12]	20.0	16.61	33.6	98.55	38.57	89.16	74.44	59.14
[14]	6.67	0.0	44.0	99.7	7.14	90.84	68.89	52.78
[29]	37.33	82.71	8.0	100.0	63.57	87.95	74.44	73.64
[15]	37.33	82.71	26.40	100.0	63.57	87.95	72.22	74.92
[13]	24.89	85.08	11.2	100.0	68.57	80.72	75.56	71.25
mixedBest	15.11	83.05	52.0	100.0	68.57	92.53	67.78	75.23

Ekman et al. [15] performed the best from the various combinations, as such, as it was proposed, in the following experiments we use this combination for the averageAU method.

2) Emotion recognition evaluation: With the AverageAU method established, we proceed to compare the three available methods with the experiments using the dataset from Avelino et al [3].

On table XV averageAU performed the best in all conditions. For data 1, all methods achieved over 80% accuracy. AverageAU performed best, with an accuracy of 94.3%. For data 7, Efficient CNN [28] performed the worst while AverageAU performed the best, with 13.74% and 75.42% accuracy, respectively.

Since Efficient CNN was unable to deal with the bearded example (figure 5b) and was outperformed by all the other algorithms, we argue that it is not suitable for our application. Regarding deepFace and averageAU, they both have similar performances.

TABLE XV: Experiments on Avelino dataset sample, for emotion algorithm

Data	SNV	TF	Efficient CNN	DeepFace	avg AU
1	Happy, Surprise		83.5%	88.35%	**94.3%**
7	Happy, Surprise		13.74%	54.7%	**75.42%**
1	Happy, Surprise	Happy, Surprise, Neutral	86.19%	88.04%	**88.5%**
1,7	Happy, Surprise	Happy, Surprise, Neutral	66.85%	81.06%	**84.34%**

To have a further comparison between deepFace and averageAU we use the CK+ dataset [14]. The results are presented in table XVI. AverageAU performed the best in all emotions with an overall accuracy score of 74.92%, while deepFace had an overall accuracy score of 31.0%. A statistical significance test was performed using McNemar test, having a p-value below 0.01.

TABLE XVI: Experiments on CK+ dataset

Method(%)	Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral	Overall
DeepFace	23.11	6.4	8.8	83.48	21.43	12.29	62.22	31.0
avgAU	37.33	82.71	26.4	100.0	63.57	87.95	72.22	74.92

Since averageAU performed the best both on our experiment in the dataset of Avelino and on CK+, we use this algorithm to classify emotions on our dataset.

3) AverageAU on faces with half masks: We show the results of the averageAU when capturing the emotions of people using masks. In table XVII we present the action units from the Ekman correspondence that can be used.

TABLE XVII: Ekman AU from superior part of the face (covid Ekman)

Method	Anger	Disgust	Fear	Happy	Sad	Surprise
Ekman	4, 5, 7	9	1, 2, 4, 5	6	1, 4	1, 2, 5

In table XVIII we show the results. It is hard to identify fear with only the upper part of the face, however, the algorithm was able to identify the rest of the emotions. This shows that with the AU visible when a person is using a half-face mask, it is possible to identify emotions, except for fear.

TABLE XVIII: Results of covid Ekman on CK+ dataset

Method(%)	Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral	overall
covidEkman	36.0	94.58	0.0	89.28	52.86	83.13	88.88	71.38

However, the action units used previously were obtained with openFace with images of people not using masks. For the averageAU we need the AU to be reliable, which is not the case when openFace deals with people with masks.

We performed an experiment where we tried to acquire the same action units but now adding a face mask to the images. The resulting action units were different, as well the results of the averageAU, table XIX.

TABLE XIX: CK+ masked, avgAU result

Method(%)	Anger	Disgust	Fear	Happy	Sad	Surprise	Neutral	overall
Ekman Covid	1.5	100.0	0.0	0.0	0.0	5.07	0.0	20.6

A reason for the difference of the action units is that OpenFace uses specific facial landmarks to align the face and to normalize it to compare to a neutral position, [30]. The masks hide some of those landmarks. The facial landmarks are also used to obtain geometry and appearance face features, to then classify action units.

VI. Conclusions and Future Work

In this work, we proposed an algorithm that detects and classifies error situations during one-to-one human-robot interactions in a controlled environment. The proposed pipeline uses facial and head features extracted from image frames of a robot onboard camera and information of robot actions. The proposed pipeline achieved significantly higher results when using the proposed set of features, which includes head, gaze, AU, emotions, and actions of the robot, than with features used in past works, that used head and gaze features [7], [8]. With an average accuracy of 72.77%, our algorithm showed promising results in the evaluation dataset. The usage of a median filter showed an improvement in the performance of the algorithm, with an average accuracy of 79.63%. Further tests validated the use of Random Forest models to detect errors and classify them with the proposed set of features. These results are obtained from an exhaustive study of the combination of several input features and classification algorithms. We want to stress the following results from the components of the pipeline:

- Random Forest classifiers work better on both error detection and error classification;
- Action units and robot context improve in a significant manner the performance of both error detection and error classification;
- Emotion features improve the performance of error detection but not error classification;
- The emotion recognition algorithm proposed in this work outperforms state-of-the-art methods in the case of our dataset. In addition, our method is computationally efficient when compared to deep learning-based methods.

We obtained promising results using the Isolation Forest algorithm, which is able to cope with mislabeled data while having similar performance to the conventional Random Forest.

Future works should study these findings in detail. In future works, we intend to perform actual human-robot interaction studies to test our algorithm in real-time, making the robot react to error information. Moreover, we will explore more contextual information, for instance age or culture, as well as temporal image and action features. Finally, we also intend to evaluate the proposed emotion recognition algorithm in more challenging scenarios, such as dealing with multiple participants simultaneously.

REFERENCES

[1] M. Giuliani, N. Mirnig, G. Stollnberger, S. Stadler, R. Buchner, and M. Tscheligi, "Systematic analysis of video data from different human–robot interaction studies: a categorization of social signals during error situations," *Frontiers in Psychology*, vol. 6, jul 2015.

[2] M. Salem, G. Lakatos, F. Amirabdollahian, and K. Dautenhahn, "Would you trust a (faulty) robot?" in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction - HRI '15*. ACM Press, 2015.

[3] J. Avelino, A. Gonçalves, R. Ventura, L. Garcia-Marques, and A. Bernardino, "Collecting social signals in constructive and destructive events during human-robot collaborative tasks," in *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 107–109.

[4] N. Mirnig, G. Stollnberger, M. Miksch, S. Stadler, M. Giuliani, and M. Tscheligi, "To err is robot: How humans assess and act toward an erroneous social robot," *Frontiers in Robotics and AI*, vol. 4, may 2017.

[5] C. J. Hayes, M. Moosaei, and L. D. Riek, "Exploring implicit human responses to robot mistakes in a learning from demonstration task," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, aug 2016.

[6] D. E. Cahya, R. Ramakrishnan, and M. Giuliani, "Static and temporal differences in social signals between error-free and erroneous situations in human-robot collaboration," in *Social Robotics*. Springer International Publishing, 2019, pp. 189–199.

[7] D. Kontogiorgos, A. Pereira, B. Sahindal, S. van Waveren, and J. Gustafson, "Behavioural responses to robot conversational failures," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, mar 2020.

[8] P. Trung, M. Giuliani, M. Miksch, G. Stollnberger, S. Stadler, N. Mirnig, and M. Tscheligi, "Head and shoulders: automatic error detection in human-robot interaction," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction - ICMI 2017*. ACM Press, 2017.

[9] P. Ekman, "An argument for basic emotions," *Cognition & emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[10] S. Li and W. Deng, "Deep facial expression recognition: A survey," *IEEE Transactions on Affective Computing*, pp. 1–1, 2020.

[11] C. A. Corneanu, M. O. Simon, J. F. Cohn, and S. E. Guerrero, "Survey on RGB, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1548–1568, aug 2016.

[12] M. Ghayoumi and A. K. Bansal, "Unifying geometric features and facial action units for improved performance of facial expression analysis," *CoRR*, vol. abs/1606.00822, 2016. [Online]. Available: http://arxiv.org/abs/1606.00822

[13] K. Karthick and J. Jasmine, "Survey of advanced facial feature tracking and facial expression recognition," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 10, pp. 2278–1021, 2013.

[14] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *2010 ieee computer society conference on computer vision and pattern recognition-workshops*. IEEE, 2010, pp. 94–101.

[15] P. Ekman and W. V. Friesen, "Measuring facial movement," *Environmental psychology and nonverbal behavior*, vol. 1, no. 1, pp. 56–75, 1976.

[16] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 59–66.

[17] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

[18] Student, "The probable error of a mean," *Biometrika*, pp. 1–25, 1908.

[19] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[20] P. D. Bridge and S. S. Sawilowsky, "Increasing physicians' awareness of the impact of statistics on research outcomes: comparative power of the t-test and wilcoxon rank-sum test in small samples applied research," *Journal of clinical epidemiology*, vol. 52, no. 3, pp. 229–235, 1999.

[21] J. Cohen, *Statistical power analysis for the behavioral sciences*. Academic press, 2013.

[22] P. Moreno, R. Nunes, R. Figueiredo, R. Ferreira, A. Bernardino, J. Santos-Victor, R. Beira, L. Vargas, D. Aragão, and M. Aragão, "Vizzy: A humanoid on wheels for assistive robotics," in *Advances in Intelligent Systems and Computing*. Springer International Publishing, dec 2015, pp. 17–28.

[23] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[24] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[25] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999.

[26] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406–421, 2018.

[27] S. I. Serengil and A. Ozpinar, "Lightface: A hybrid deep face recognition framework," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2020, pp. 23–27.

[28] H. Siqueira, S. Magg, and S. Wermter, "Efficient facial feature learning with wide ensemble-based convolutional neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 5800–5809.

[29] I. Tautkutė and T. Trzciński, "Classifying and visualizing emotions with emotional dan," *Fundamenta Informaticae*, vol. 168, no. 2-4, pp. 269–285, 2019.

[30] A. Zadeh, Y. Chong Lim, T. Baltrusaitis, and L.-P. Morency, "Convolutional experts constrained local model for 3d facial landmark detection," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2519–2528.