2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW) | 978-1-6654-0021-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ACIIW52867.2021.9666442

Task-based Classification of Reflective Thinking Using Mixture of Classifiers

Saandeep Aathreya
Dept. Computer Science and Eng.
University of South Florida
Tampa, Florida
saandeepaath@usf.edu

Liza Jivnani
Dept. Computer Science and Eng.
University of South Florida
Tampa, Florida
ljivnani@usf.edu

Shivam Srivastava
Dept. Computer Science and Eng.
University of South Florida
Tampa, Florida
shivam4@usf.edu

Saurabh Hinduja
Dept. Computer Science and Eng.
University of South Florida
Tampa, Florida
saurabhh@usf.edu

Shaun Canavan
Dept. Computer Science and Eng.
University of South Florida
Tampa, Florida
scanavan@usf.edu

***Abstract*—This paper studies the problem of Reflective Thinking in children during mathematics related problem solving activities. We present our approach in solving task 2 of the AffectMove challenge, which is Reflective Thinking Detection (RTD) while solving a mathematical activity. We utilize temporal data consisting of 3D joint positions, to construct a series of classifiers that can predict whether the subject appeared to possess reflective thinking ability during the given instance. We tackle the challenge of highly imbalanced data by incorporating and analyzing several meaningful data augmentation techniques and handcrafted features. We then feed different features through a number of machine learning classifiers and select the best performing model. We evaluate our predictions on multiple metrics including accuracy, F1 score, and MCC to work towards a generalized solution for the real-world dataset.**

***Index Terms*—Handcrafted, Reflective Thinking, UMAP, ML**

I. Introduction

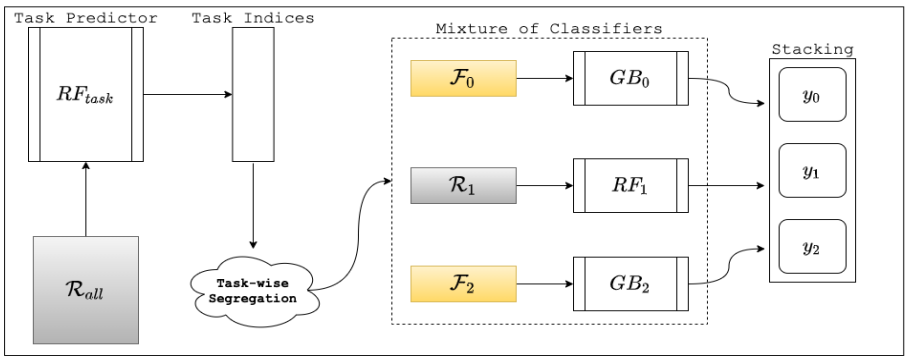
Reflective Thinking is the ability to find insight and information to develop a deeper understanding of problems. In the context of learning, reflective thinking is vital to both teachers and students, and despite its potential in improving learning, there is a scarcity in implementing tools that determine whether a student engages in reflective thinking [1]. According to literature, there is no straightforward definition for reflective thinking in the context of problem solving as most of the work over the years has focused on reflection as an experiential learning [2]. Therefore, the most apt definition can be obtained by Dewey et. al. [3], which takes into consideration the takeaways from past experiences and using this information to find the solution. Having a precise definition is important in real-world problem solving techniques where data annotation is more accurate resulting in less errors.

It has been shown that due to short-term memory and the transient consciousness of humans, it is arduous to focus on a single task for a longer period of time [4]. This is especially highlighted in a classroom environment where children are expected to have attention for the duration of an entire class. This becomes non-conducive to learning and therefore it is not only important to address non-reflective behaviour but also have the means to identify them. Manually assessing reflective thinking in any environment is time consuming and is a demanding task for the educators. With the development of several advanced machine learning algorithms, the challenging problem of being able to identify reflective and non-reflective behaviour can be deciphered more effectively by taking into account various modalities that are relevant to the activity being performed. For example, Liu et. al. [5] used text data to assess the reflective writing of pharmacy students. Additionally, students involved in learning can be functionally evaluated on their reflective thinking using facial expressions and audio signals [2]. Considering this, we use the 3D joint positions of the body and handcrafted features which are extracted out of the raw features for reflective thinking analysis.

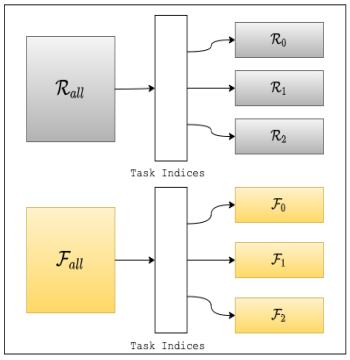
Our work is focused on solving task 2 of the AffectMove challenge [6] where given a set of temporal, 3D joint positions for 17 joints, we aim to estimate the reflective behaviour for each temporal instance. Each instance belongs to solving one of the three tasks, from the WeDraw-1 dataset [2], which are different types of mathematical problems ranging from forming angles, making shapes, to finding symmetry. We design a task specific approach which takes into account the diverse nature of each of the tasks using several features. We employ data preprocessing techniques such as resampling and standardization, and examine various augmentation methods like SMOTE [7] and rotations.

Unlike other works which use LSTM [8] for temporal data classification [9]–[11], we show that recurrent networks tend to overfit and perform poorly when the data is highly imbalanced, in the context of the weDraw-1 dataset. We also show that substantial data augmentation on LSTM networks show little to no improvement on the performance of the network. Consequently, we move to a more statistical approach in machine learning and employ various classifiers in our

978-1-6654-0021-3/21/$31.00 ©2021 IEEE



(a) Overall Pipeline



(b) Task-wise Segregation

Fig. 1: (a) Overall Pipeline architecture. The raw features $\mathcal{R}_{all}$ is first fed into the Random Forest classifier RF_{task} for task prediction. (b) The predictions of this classifier is used to segregate the Raw features $\mathcal{R}_{all}$ into its respective tasks $[\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2]$ and Handcrafted features $\mathcal{F}_{all}$ into $[\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2]$. (a) The new task-wise feature set is then fed into the mixture of classifiers for RTD classification, namely, Gradient Boosting (GB_0 and GB_2) for tasks 0 and 2 respectively and Random Forest (RF_1) for task 1. Finally, each output vector y_i for task i is stacked to form a single output vector.

approach. We hypothesize that a single classifier will have a sub-par performance and accordingly select a mixture of classifiers such as Random Forests [12] and Gradient Boosting [13] trained on the aforementioned feature sets to obtain a reasonable performance. The contributions of our work are three-fold and can be summarized as follows.

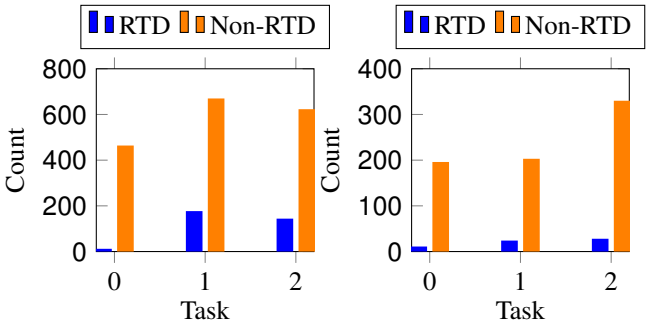
1) A task-specific pipeline architecture (Fig. 1) is designed to classify reflective and non-reflective thinking. The pipeline first predicts the task and then a task-specific classifier is used for Reflective Thinking Detection (RTD). Our architecture consists of a mixture of classifiers such as Random Forest [12] and Gradient Boosting [13] for task prediction and RTD.
2) A set of handcrafted features are proposed, from the given raw features, and substantial evaluation of these features on different classifiers is performed. Moreover, we evaluate our predictions using metrics including Accuracy, F1 score and MCC [14]
3) We perform comprehensive data analysis with UMAP [15] embeddings to show the how different feature contribute in task predictions and RTD classification.

II. Related Work

Reflective thinking is a ambiguous concept [16]. Dewey [17] [18] introduced the term "Reflective thinking", where he theorised the five phases of reflective thinking which are warranted for teaching and learning. Gelter [19] has explained why reflective thinking is uncommon. They propose that spontaneous reflective thinking happens only when something has gone wrong making reflective thinking rare. Reflective thinking is very important for cognitive thinking and learning, and for pedagogical purposes, it enables students to better evaluate themselves by structuring their thoughts in a focused manner. This helps them to better resolve uncertainties and complex situations [20]–[23]. Volta et. al. [24] analysed the cognitive states of visually impaired children solving mathematical tasks by examining the body communication indicative of engagement and confidence levels. Consequently, 2D positional data demonstrates the capability for automatic engagement and confidence level detection. The works on reflective thinking using different modalities have been largely focused on a few specific domains ranging from academics, healthcare and music learning. Johnston et. al [25] outlined a theoretical approaches to encourage creative and reflective approach to music making.

There is limited research on automatic detection of reflective thinking. Toptsis et. al. [26] propose the use of artificial k-lines to solve three different problems, illustrating its application in inter-domain areas. Liu et. al. [27] used online text data of teachers to analyse their reflective thinking. Based on the outcome of inductive content analysis [28], a single-label text classifier was built which was then applied on large-scale unstructured data. Liu et. al [5] used text data to understand pharmacy student's reflective statements about their work placement. They developed a machine learning approach for binary classification on the text data and evaluated them on four different classifiers, on a dataset of 301 statements. Olugbade et. al [2] evaluated the reflective thinking among 26 children in mathematical problem solving settings. They used LSTM networks on 17 joint positions to temporally determine the reflective behaviour. Another branch of approach to any behaviour-based recognition is to discern the tasks or activities involved. This task-based solution has been shown to influence the final behaviour prediction in a favourable manner. Wang et. al. [29] performed continuous protective behaviour detection based on Human Activity Recognition (HAR) using Graph CNNs [30] and LSTMs.

Aside from reflective thinking, a great deal of work has been done identifying affect using multiple modalities. Hinduja et al. [31] showed that hand crafted features can be used to detected empathy with the help of machine learning. They were able to detect valence using multimodal data and different



(a) Training set (b) Validation set

Fig. 2: Task-wise distribution of WeDraw-1 dataset [2].

machine learning algorithms for each modality. Srivastava et al. [32] used multimodal data to detect engagement levels. They showed the affect of unbalanced data on performance metrics and that body landmarks can be used to detect affect. Aathreya et al. [33] used multi level training to boost performance, that is, they trained the network at different levels.

III. Data Analysis

A. WeDraw-1 Dataset

The WeDraw-1 dataset [2] consists of 2090 movement sequences of 13 children as training samples, 792 movement sequences of 5 children as validation samples, and 672 movement sequences of 6 children as testing samples. Each sample contains an $M \times N$ matrix, where $M = 51$ and refers to the 3D location of 17 full-body joints and T denotes the number of frames in that sample. The number of frames varies in range 11-146 among participant due to frame dropout during recording, with the mean length of sequences being empirically determined as 120. Each sample refers to a 5-second segment of a participant when performing one of the three tasks. More specifically, **Task 0:** Forming Angles; **Task 1**: Finding Symmetry & Making Shape Reflections; and **Task 2**: Bodily Angles Sums and Differences, Rotating in Angles.

The data is divided into 2 classes - reflective thinking present (RTD) and absent (Non-RTD). The distribution of samples among these two classes is highly imbalanced, with 333 RTD sample over 1757 Non-RTD samples in the training sample. See Fig. 2 for the task-wise distribution.

B. Handcrafted and Raw Features

1) Raw Features: The WeDraw-1 movement dataset [2] consists of raw features which have been used in our analysis. The dataset is comprised of sequences where each sequence $\mathcal{S}$ belongs to a task $t \in [0, 1, 2]$. $\mathcal{S}$ consists of 3D positions for 17 different joints. These features are denoted by $\mathcal{R}_t \in \mathbb{R}^{51 \times f}$, where f represents the number of frames in $\mathcal{S}$ and 51 is the flattened dimension for (17×3) joint positions. See Fig. 3 for an example sequence of the 3d joint plots.

As part of preprocessing, we first normalize the data to make it invariant to any feature scaling within the dataset. We perform z-score normalization [34] to bring the data to a zero mean and unit variance. Next, to make the raw features compatible with the various classifiers and deep learning models, we re-sample the given features to a fixed number of frames. Therefore, $\mathcal{R}_t$ is now resampled to $\mathcal{R}_t \in \mathbb{R}^{51 \times k}$, where k is the fixed value.

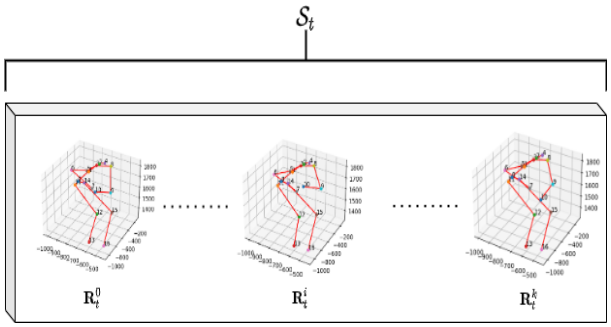


Fig. 3: An example sequence for a task t. $\mathcal{R}^i$ is the 17 joint position for the frame i.

2) Handcrafted Features: The handcrafted features were divided into $m = 5$ distinct segments $H_{seg} = \{$*Left hand to Right Hand, Head to Left hand, Head to Right hand, Head to Right knee, Head to Left knee*$\}$ (Fig. 4). We empirically found these features to be well suited for recognizing reflective thinking. To extract the handcrafted features, we first handled the NaN values values by keeping them invariable. Hence, while calculating the mean and standard deviation, the NaN values were removed. Secondly, person dependent z-score normalization was used by evaluating the participant ids. Consequently, the range of the normalized data was $[-3, 3]$. Next, the data was iteratively resampled to size k, to make it compatible with each of the classifiers. Finally, we extracted features based on the joint locations, where every feature was comprised of two joints (H_{seg}). The Euclidean distance, D, between the two joints was calculated, and then concatenated into our new feature vector $\mathcal{F} = \{D_1, \ldots, D_m\}$, where D_i is the i^{th} distance calculated from H_{seg}.

3) Handcrafted Features - Extended: To investigate the contribution of different body landmarks, we also extracted $h = 29$ hand-crafted features ("FEATS" [2]). As the samples are of varying length, the raw signals are first resampled to k frames such that $\mathcal{R}_t^i \in \mathbb{R}^{51 \times k}$. Extracted features include head and trunk orientation, positional and angular energy, range and amount of movement, and hand-to-head distance. The features were then concatenated into a new feature vector $\mathcal{F}_{ext} = \{f_1, \ldots, f_h\}$, where f_i is the i^{th} extracted feature. We refer the reader to the work from Olugbade et al [2], for specific details on the 29 extracted features.

4) Temporal Distances: Similar to Section III-B2, where we calculate the euclidean distances between the joints within a single frame of the sequence $\mathcal{S}$, we also calculate the euclidean distances along the temporal dimensions between two consecutive frames of the same joint. Given a sequence with f frames, we calculate the distance as shown below

$$\mathcal{D}_j^i = \sqrt{(\mathcal{R}_j^{i+1} - \mathcal{R}_j^i)^2}, \tag{1}$$

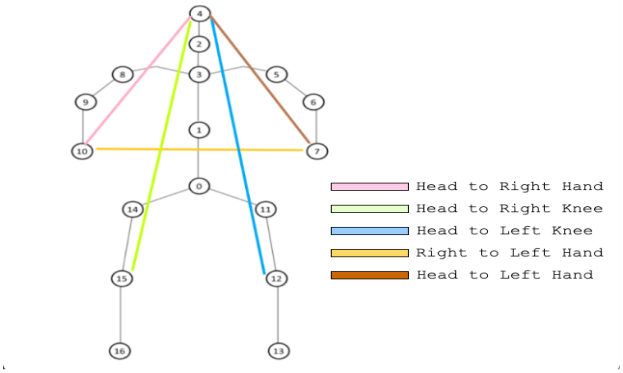


Fig. 4: Handcrafted features (H_{seg}).

where i and $i+1$ denotes the joint positions between two consecutive frames in the sequence for the joint j. These distances are then concatenated into a new feature vector $\mathcal{F}_{temp} = \{D_1, \ldots, D_j^{f-1}\}$, where D_i is the i^{th} distance, and resampled to $\mathcal{D}_j \in \mathbb{R}^{17\times k}$.

C. Data Augmentation

To handle the imbalance in the given dataset, we investigate multiple data augmentation techniques to assess their performance. In some cases, there is a 1:40 ratio between the majority and minority class (Fig 2). It is worth noting that by simply oversampling the minority class with repetition resulted in overfitting. We detail the investigated techniques in the following subsections.

1) SMOTE: Synthetic Minority Oversampling Technique (SMOTE) [7] is a widely used oversampling technique where synthetic examples can be created for the minority class. It works on the principle of nearest neighbors where it generates new examples by combining the target feature and the feature of it's neighbor. Motivated by applications of SMOTE and its variants [35], and it's applicability to continuous data [36], we seek to resolve the data imbalance using SMOTE. We detail the effect of SMOTE in Section V, where we demonstrate the improvement in training, as well as prediction on unseen data which suggests SMOTE works well in this context.

2) Reflection and Rotation: There has been a significant research on image classification problems that shows the effectiveness of image rotation, flipping and geometric transformations [37], [38] as a means of handling imbalanced data. Similarly, olugbade et. al. [2] apply reflection on the joints by inverting each of the 3 axes to create $4x$ more data. In this paper, we aim to implement a similar augmentation technique for 3D plots by applying a rotation matrix along the z-axis on the raw features (III-B1). This is performed by multiplying each feature $\mathcal{R} \in \mathbb{R}^{17\times 3}$ by a rotation matrix λ as

$$Rot(\mathcal{R}) = \lambda_{3\times 3} \cdot \mathcal{R}^{T}_{3\times 17}, \tag{2}$$

which rotates the original 3D plot along the z- axis by 90°. Fig. 6 shows the original joint positions for a frame vs the joint positions rotated along the z-axis by 90°.

D. Uniform Manifold Approximation and Projection (UMAP)

Considering the bag of features and augmentation techniques available to us ($\mathcal{R}, \mathcal{F}, \mathcal{F}_{ext}$ and $\mathcal{F}_{temp}$), it is important to find the right combination which achieves the optimum performance on the classifiers. To perform this task, we isolate the feature(s) that are unlikely to contribute in RTD classification via dimensionality reduction [39] and empirically choose the best augmentation techniques.

Since our pipeline is task-specific, we first plot the UMAP embeddings [15] of raw features for the tasks as shown in Fig. 7. UMAP is a dimensionality reduction technique that works on the principle of topological data analysis. The plot shows a clear segregation between different tasks, especially for task 2. This can be explained, in part, by the nature of task 2 where the participants are mostly seated with movements limited to the arm and shoulder. Considering this, we used the raw features for task prediction, which is the first step in our proposed pipeline (Fig. 1).

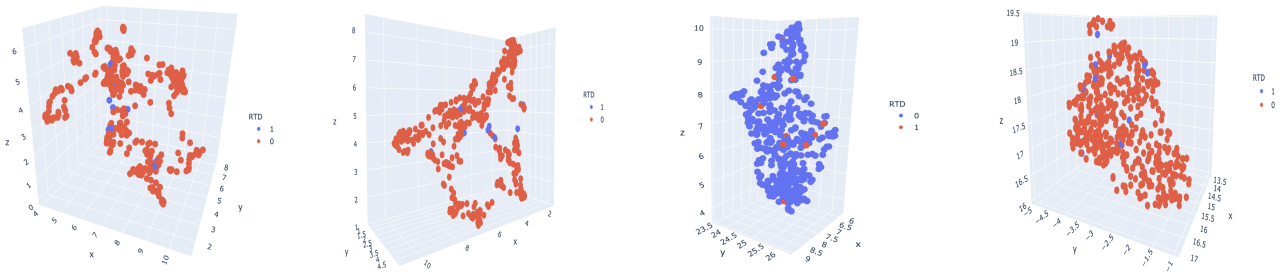
To assess and pick the feature for RTD classification, we plot the task-wise UMAP for all the features mentioned in section III-B. Generally, the UMAP embeddings do not show a clear distinctions for RTD segregation. Consequently, we use the embeddings to rule out the features that don't add to the classification. Figs. 5, 8 and 9 show the plots for task 0, 1 and 2 respectively for each feature type (e.g. raw). Although none of the plots show a clear distinction between the RTD labels, subplots (c) and (d), for all 3 figures, suggest the least amount of separation for handcrafted-extended and temporal. Considering this, for our selected feature types, we use raw and handcrafted features. We also justify this finding in the experiments in Section V.

IV. Methodology

Fig. 1 gives an overview of the proposed architecture. Based on the comprehensive study on different features (Section III), the proposed pipeline ultimately utilizes the raw and handcrafted features with SMOTE for augmentation. To clarify, we only apply SMOTE to the training set and use random forest [12] and gradient boost classifiers [13] for task and RTD predictions.

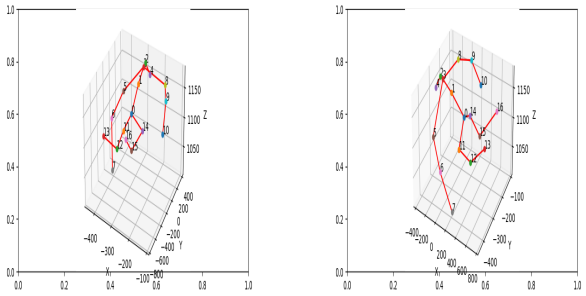
A. Task prediction

As the proposed approach is task-specific, accurate classification of RTD per class is contingent upon the task classification itself. Any error in task classification propagates to the RTD classification thus affecting the overall performance. As a result, the task predictor must be as precise as possible. As shown in Fig. 7, the raw features quite distinctively categorize the tasks and consequently, are best suited for task classification. This is the first step in Fig. 1a, where a random forest classifier, RF_{task}, is used to predict the tasks. Subsequently, the output of the RF_{task} is captured by a submodule for task-wise segregation of features (Fig 1b).



(a) Raw (b) Handcrafted (c) Handcrafted - Extended (d) Temporal

Fig. 5: UMAP embeddings, for different feature types, for Task 0 on RTD labels.



(a) Original plot (b) Rotated by 90°

Fig. 6: 3D rotation along z-axis.

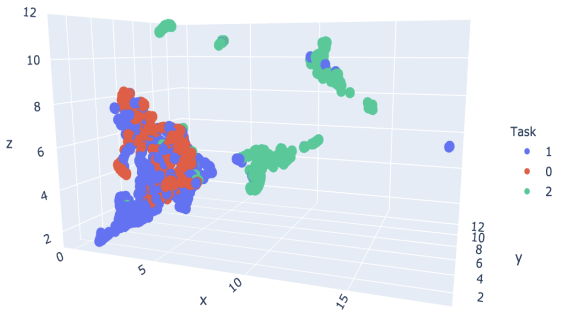


Fig. 7: UMAP embeddings of Raw Features based on tasks.

B. Task-wise segregation of features

Since the following steps involves classifying the labels, the raw and handcrafted features are used with multiple classifiers. Each of the features must be segregated accordingly based on the predicted task (see Fig. 1b). As a result, the raw features $\mathcal{R}_{all}$ are split into $[\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2]$ and handcrafted features, $\mathcal{F}$ are split into $[\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2]$ where each sub-feature belongs to its respective task predicted.

C. Mixture of Classifiers

The mixture of classifiers are shown in Fig. 1a which predicts the RTD classification of different tasks and combines them into a single output vector y. The UMAP embeddings helped in recognizing the features unsuitable for classification, omitting them from the pipeline. Through a grid-search, we aim to find to the best set of classifiers and hyper-parameters for the raw and handcrafted features. Since the dataset is highly imbalanced, we focus on F1 minority and MCC scores to select the best combination of classifier and features. Other metrics such as accuracy and F1 majority remained fairly consistent throughout our experiments and have been shown as imprecise metrics for imbalanced data evaluation [40].

For task 0, Gradient Boosting (GB_0) was found to provide the best MCC and F1 minority using the handcrafted features $\mathcal{F}_0$. Gradient boosting has been shown to improve the performance of the model with imbalanced datasets [41], [42]. It is a classifier that uses a set of weak learners to build a strong learner (decision trees). It's a method to deal with imbalanced data wherein the training set is successively constructed based on the misclassified examples. The classifier GB_0 was configured with 200 trees with a learning rate of 0.5, and the max depth, of individual estimators, was kept to one.

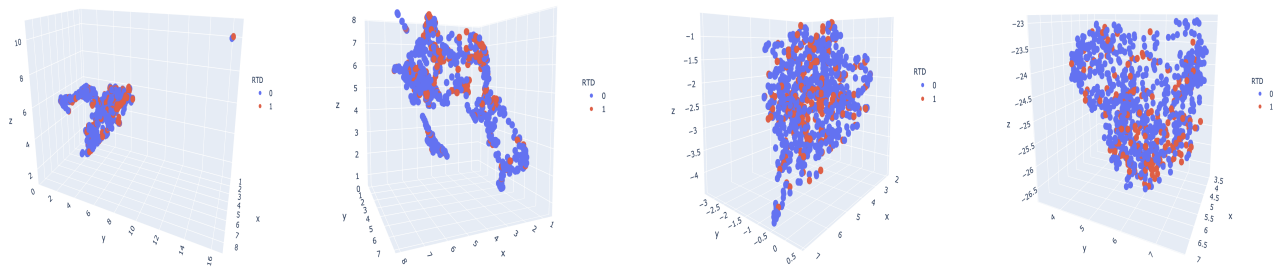
For task 1, Random Forest classifier (RF_1) provided the best results in terms of F1 minority and MCC with the raw features $\mathcal{R}_1$. Notwithstanding the augmentation technique (SMOTE) applied during the pre-processing stage, we nonetheless modify the classifier to accommodate for the data imbalance. Random forest with its default parameters can be unsuitable for imbalanced datasets [43], and therefore we opt for a class weighted random forest where weights are assigned to each class while calculating the impurity score of the chosen split point. RF_1 was configured with 100 estimators with a minimum of 1 leaf at the leaf node.

Finally, for task 2, the best combination of features and classifiers was found to be the handcrafted features $\mathcal{F}_2$ with Gradient Boosting, GB_2. From our 3D plots, task 2 was distinctively different from the other 2 where the participant was seated throughout the experiments and therefore achieved the lowest performance compared to other two tasks (Fig. 3). As per the description, this particular task implied to have minimal movements of the body making it challenging. GB_2 was configured with 150 estimators and a learning rate of 0.8 with a maximum depth of individual estimators at four.

V. Experiments and Results

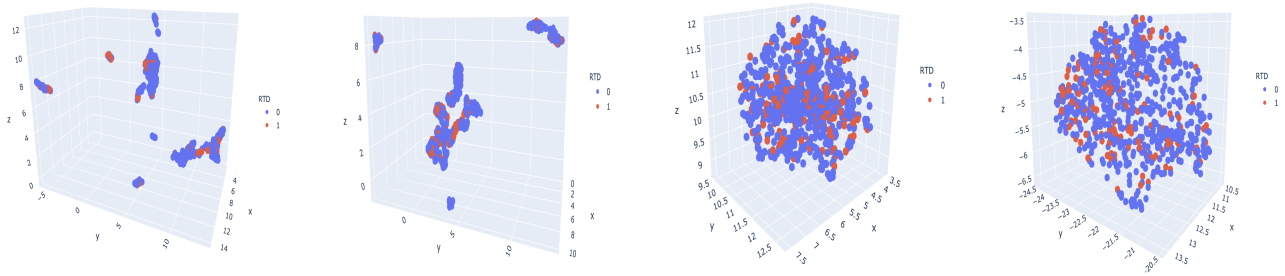
A. Task and RTD prediction

As can be seen in Table I, RF_{task} is able to achieve a high accuracy of 91% with an average F1 score of 0.90 and MCC



(a) Raw (b) Handcrafted (c) Handcrafted - Extended (d) Temporal

Fig. 8: UMAP embeddings, for different feature types, for Task 1 on RTD labels



(a) Raw (b) Handcrafted (c) Handcrafted - Extended (d) Temporal

Fig. 9: UMAP embeddings, for different feature types, for Task 2 on RTD labels

TABLE I: Task prediction across training and validation sets, for different features and different classifiers.

Dataset	Method	Feature	Accuracy	F1	MCC
Train	Random Forest	Raw	1	[1.0, 1.0, 1.0]	1
Train	LSTM	Handcrafted - Extended	0.81	[0.75, 0.81, 0.86]	0.81
Val	**Random Forest**	**Raw**	**0.91**	**[0.89, 0.87, 0.95]**	**0.86**
Val	LSTM	Handcrafted - Extended	0.75	[0.77, 0.65, 0.81]	0.62

TABLE II: Grid-search results of the pipeline. *HF* refers to Handcrafted Features ($\mathcal{F}$).

Dataset	Task	Method	Feature	Accuracy	F1	MCC
Train	0	Gradient Boosting	HF	1.0	[1.0, 1.0]	1
	1	Random Forest	Raw	1.0	[1.0, 1.0]	1
	2	Gradient Boosting	HF	1.0	[1.0, 1.0]	1
	ALL	[RF, GB]	[HF, Raw]	1.0	[1.0, 1.0]	1
Validation	0	Gradient Boosting	HF	0.93	[0.96, 0.4]	0.37
	1	Random Forest	Raw	0.9	[0.94, 0.38]	0.35
	2	Gradient Boosting	HF	0.82	[0.90, 0.22]	0.14
	ALL	**[RF, GB]**	**[HF, Raw]**	**0.85**	**[0.91, 0.28]**	**0.2**
	0 + 1	[RF, GB]	[HF, Raw]	0.92	[0.96, 0.39]	0.36
Test	**ALL**	**[RF, GB]**	**[HF, Raw]**	**0.81**	**[0.89, 0.23]**	**0.13**

TABLE III: RTD classification with *SMOTE* as Data Augmentation technique using LSTM networks

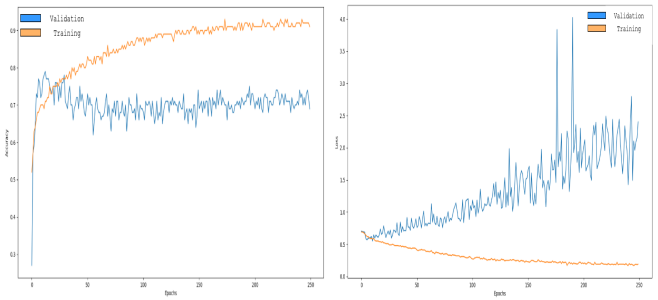
Dataset	Task	Accuracy	F1	MCC
Train	0	0.86	[0.87, 0.85]	0.73
	1	0.72	[0.74, 0.7]	0.45
	2	0.73	[0.75, 0.71]	0.47
	ALL	0.91	[0.92, 0.91]	0.83
Validation	0	0.9	[0.95, 0.09]	0.07
	1	0.68	[0.84, 0.34]	0.26
	2	0.73	[0.84, 0.06]	0.05
	ALL	0.75	[0.85, 0.18]	0.05

of 0.86. We believe this performance is justifiable for further steps down the pipeline.

For RTD classification (Table II) the results of the validation set are consistent with the test set. We achieve a superior performance for task 0 and 1 over 2, which is reflected in the overall result. This can be further seen in the rows for tasks 0 and 1 in Table II where we see a large margin in the F1 and MCC scores between *ALL* and the rows for those tasks.

B. Training LSTM models

1) Extended Handcrafted Features: For resampling, we empirically chose the value of $T = 120$. The feature set $\mathcal{F}_{ext} \in \mathbb{R}^{n \times 120 \times 29}$, where n is the number of training samples is fed to a stacked LSTM network. This network consists of three stacked LSTM layers with 32 cells each. The output of



(a) Accuracy (b) Loss

Fig. 10: Accuracy/loss plots for training vs validation (LSTM).

the stacked LSTM network is then passed to a Dense layer with softmax activation and N neurons. To study the impact of handcrafted features, we trained different models for task-prediction ($LSTM_{task}$) and RTD-prediction($LSTM_{RTD}$). Since each sequences $\mathcal{S}$ belongs to a task $t \in [0, 1, 2]$, N=3 for model $LSTM_{task}$, whereas N=2 for model $LSTM_{RTD}$, which is responsible for detecting reflective thinking. Both models were compiled with Adam [44] as the optimizer with a learning rate of 0.001 and are trained for 50 epochs. These models are trained with the original data distribution.

Table I details the performance of our $LSTM_{task}$ when trained with the 29 handcrafted features. We can see that this model achieved an accuracy of 75% on the validation set with an average F1 score of 0.74 and MCC score of 0.62. In the case of our model $LSTM_{RTD}$, due to the high imbalance in dataset, the model was overfit to the majority class (Non-RTD) and performs poorly when detecting reflective thinking.

2) Raw Features: In our experiments, we train the LSTM [8] models on raw features, $\mathcal{R}$, using the two data augmentation techniques as detailed in Section III-C. This network consists of 4 stacked LSTM layers with 32 hidden cells in each. A learning rate of 1e-3 was set with Adam optimizer [45] was used for 250 epochs. Table III shows the result of training with SMOTE as data augmentation. The table draws two observations - The lstm network clearly overfits on the data. This can be seen in the Fig. 10 which shows the plots for accuracy and loss for training and validation set. The validation loss appears to be fluctuating throughout which is another indication of the model not learning. Additionally, it is unable to handle the class imbalance as can be noted in Table III.

Table IV shows the result of LSTM trained on raw features using rotation as data augmentation. It can be seen that rotation does not contribute to the performance of the network. LSTM have been a well known deep learning model to handle temporal instances and in this case, due to the high class imbalance, Most of the outputs are the majority class.

C. No Augmentation

A final experiment that we performed was using the original pipeline architecture (Fig. 1) but without any augmentation techniques. This was to investigate if the finely tuned hyper-parameters of the classifiers and the data augmentation are positively contributing to our final results. Table V shows the result of this approach on the validation set. Once again, the results follow a pattern akin to previous experiments which resulted in majority class prediction throughout where the model is unable to learn the minority class data. Although the accuracies for both gradient boosting and random forest are reasonably high, it can be seen that the minority class F1 and MCC are significantly lower compared to the augmented results (Table II). As can be seen in the lat column of Table V, the MCC scores show a slight negative correlation with Gradient Boosting, and close to 0 correlation for Random Forest. This suggests that SMOTE is synthetic good synthetic features, that are similar to the real data, and helping mitigate the class imbalance found in the WeDraw-1 dataset.

TABLE IV: RTD classification with *Reflection* as Data Augmentation technique using LSTM networks.

Dataset	Task	Accuracy	F1	MCC
Train	0	0.77	[0.87, 0.25]	0.22
	1	0.92	[0.92, 0.92]	0.85
	2	0.92	[0.91, 0.89]	0.87
	ALL	0.92	[0.88, 0.85]	0.85
Validation	0	0.91	[0.95, 0.0]	0.08
	1	0.87	[0.93, 0.12]	0.08
	2	0.87	[0.91, 0.02]	0.04
	ALL	0.87	[0.87, 0.06]	0.04

TABLE V: Results on validation set without augmentation.

Dataset	Task	Method	Feature	Accuracy	F1	MCC
Validation	0	Gradient Boosting	HF	0.94	[0.97, 0.0]	-0.02
	1	Random Forest	Raw	0.88	[0.94, 0.07]	0.05
	2	Gradient Boosting	HF	0.7	[0.82, 0.05]	-0.09
	ALL	[RF, GB]	[HF, Raw]	0.85	[0.90, 0.03]	0.0

VI. Conclusion, Limitations, and Future Work

In this paper, we propose a task-specific architecture where we first predict the task and then the corresponding RTD classification. The proposed architecture achieved competitive results on tasks 0 and 1, given the high imbalance in the dataset and produced decreased accuracies for task 2. This can explained, in part, by the nature of task 2 which further reduced the overall performance of the model. Additionally, we provided an analysis of different features and augmentation techniques, as well as UMAP embeddings.

Although these results are encouraging, there are some limitations. First, is it's ability to predict the minority class (reflective thinking). Considering this, we will further investigate new feature types, including the fusion of hand-crafted and deep features. Secondly, only the WeDraw-1 challenge dataset [2] was evaluated. While this is a challenging, real-world dataset, to accurately assess the utility of the proposed approach, larger more varied datasets need to be analyzed. Finally, while random forests have shown to work well for 3D gesture data [46], they used balanced data. Using unbalanced data, as done here, new more advanced classifiers are needed to accurately recognize reflective thinking.

REFERENCES

[1] D. Kember, D. Y. Leung, A. Jones, A. Y. Loke, J. McKay, K. Sinclair, H. Tse, C. Webb, F. K. Yuet Wong, M. Wong *et al.*, "Development of a questionnaire to measure the level of reflective thinking," *Assessment & evaluation in higher education*, vol. 25, no. 4, pp. 381–395, 2000.

[2] T. Olugbade, J. Newbold, R. Johnson, E. Volta, P. Alborno, R. Niewiadomski, M. Dillon, G. Volpe, and N. Bianchi-Berthouze, "Automatic detection of reflective thinking in mathematical problem solving based on unconstrained bodily exploration," *IEEE Transactions on Affective Computing*, 2020.

[3] T. YÜRÜK, "John dewey, how we think, a restatement of the relation of reflective thinking to the educative process, dc heath and company, 1933, 10+ 301 s," *Ankara Üniversitesi İlahiyat Fakültesi Dergisi*, vol. 48, no. 1, pp. 185–188.

[4] H. Gelter, "Why is reflective thinking uncommon," *Reflective practice*, vol. 4, no. 3, pp. 337–344, 2003.

[5] M. Liu, S. B. Shum, E. Mantzourani, and C. Lucas, "Evaluating machine learning approaches to classify pharmacy students' reflective statements," in *International Conference on Artificial Intelligence in Education*. Springer, 2019, pp. 220–230.

[6] T. Olugbade, R. Sagoleo, S. Ghisio, N. Gold, A. Williams, B. de Gelder, A. Camurri, G. Volpe, and N. Berthouze, "The affectmove 2021 challenge - affectrecognition from naturalistic movement data," Sep 2021.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[10] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1647–1656.

[11] C. Si *et al.*, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *CVPR*, 2019.

[12] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[13] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[14] "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *BAA - Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[15] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[16] M. Clarà, "What Is Reflection? Looking for Clarity in an Ambiguous Notion," *Journal of Teacher Education*, vol. 66, no. 3, pp. 261–271, May 2015. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0022487114552028

[17] J. Dewey, *How we think.* Lexington: D C Heath, 1910. [Online]. Available: http://content.apa.org/books/10903-000

[18] ——, *How we think: a restatement of the relation of reflective thinking to the educative process*. Boston: Houghton Mifflin, 1998.

[19] H. Gelter, "Why is reflective thinking uncommon," *Reflective Practice*, vol. 4, no. 3, pp. 337–344, Oct. 2003. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/1462394032000112237

[20] C. Chan and K. Lee, "Reflection literacy: A multilevel perspective on the challenges of using reflections in higher education through a comprehensive literature review," *Educational Research Review*, vol. 32, p. 100376, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1747938X20308368

[21] C. Rodgers, "Defining Reflection: Another Look at John Dewey and Reflective Thinking," *Teachers College Record*, vol. 104, no. 4, pp. 842–866, Jun. 2002. [Online]. Available: http://www.tcrecord.org/

[22] D. A. Kolb, *Experiential learning: experience as the source of learning and development*. Englewood Cliffs, N.J: Prentice-Hall, 1984.

[23] J. Mezirow, *Transformative dimensions of adult learning*, 1st ed., ser. The Jossey-Bass higher and adult education series. San Francisco: Jossey-Bass, 1991.

[24] E. Volta, R. Niewiadomski, T. Olugbade, C. Gilio, E. Cocchi, N. Berthouze, M. Gori, and G. Volpe, "Analysis of cognitive states during bodily exploration of mathematical concepts in visually impaired children," in *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2019, pp. 1–7.

[25] A. Johnston, S. Amitani, and E. Edmonds, "Amplifying reflective thinking in musical performance," in *Proceedings of the 5th conference on Creativity & cognition*, 2005, pp. 166–175.

[26] A. A. Toptsis, "Reflective thinking, machine learning, and user authentication via artificial k-lines," *Int. J. Adv. Sci. Technol*, vol. 5, pp. 51–73, 2009.

[27] Q. Liu, S. Zhang, Q. Wang, and W. Chen, "Mining online discussion data for understanding teachers reflective thinking," *IEEE Transactions on Learning Technologies*, vol. 11, no. 2, pp. 243–254, 2017.

[28] H. Kyngäs, "Inductive content analysis," in *The application of content analysis in nursing science research*. Springer, 2020, pp. 13–21.

[29] C. Wang, Y. Gao, A. Mathur, A. C. De C. Williams, N. D. Lane, and N. Bianchi-Berthouze, "Leveraging activity recognition to enable protective behavior detection in continuous data," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, pp. 1–27, 2021.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[31] S. Hinduja, M. T. Uddin, S. R. Jannat, A. Sharma, and S. Canavan, "Fusion of Hand-crafted and Deep Features for Empathy Prediction," in *FG*, Lille, France, May 2019, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/8756522/

[32] S. Srivastava *et al.*, "Recognizing Emotion in the Wild using Multimodal Data," in *Proceedings of the 2020 International Conference on Multimodal Interaction*. Virtual Event Netherlands: ACM, Oct. 2020, pp. 849–857. [Online]. Available: https://dl.acm.org/doi/10.1145/3382507.3417970

[33] S. A. S. Lakshminarayan, S. Hinduja, and S. Canavan, "Three-level Training of Multi-Head Architecture for Pain Detection," in *FG*, Buenos Aires, Argentina, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9320171/

[34] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

[35] A. Fernández *et al.*, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[36] A. Bernardo, H. M. Gomes, J. Montiel, B. Pfahringer, A. Bifet, and E. Della Valle, "C-smote: Continuous synthetic minority oversampling for evolving data streams," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 483–492.

[37] C. Lei, B. Hu, D. Wang, S. Zhang, and Z. Chen, "A preliminary study on data augmentation of deep learning for image classification," in *Proceedings of the 11th Asia-Pacific Symposium on Internetware*, 2019, pp. 1–6.

[38] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[39] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54 776–54 788, 2020.

[40] R. Malhotra and M. Khanna, "An empirical study for software change prediction using imbalanced data," *Empirical Software Engineering*, vol. 22, no. 6, pp. 2806–2851, 2017.

[41] N. Cahyana, S. Khomsah, and A. S. Aribowo, "Improving imbalanced dataset classification using oversampling and gradient boosting," in *2019 5th International Conference on Science in Information Technology (ICSITech)*. IEEE, 2019, pp. 217–222.

[42] R. Low, L. Cheah, and L. You, "Commercial vehicle activity prediction with imbalanced class distribution using a hybrid sampling and gradient boosting approach," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[43] C. Jose and G. Gopakumar, "An improved random forest algorithm for classification in an imbalanced dataset." in *2019 URSI Asia-Pacific Radio Science Conference (AP-RASC)*. IEEE, 2019, pp. 1–4.

[44] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–2.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[46] J. Schioppo, Z. Meyer, D. Fabiano, and S. Canavan, "Learning sign language in a virtual environment," *CHI Late Breaking Work*, 2019.