Paper:

An Incremental Multi-model Learning Architecture for Emotion Recognition

Akihiro Matsufuji[*1], **Erina Kasano**[*1], **Eri Sato-Shimokawara**[*1], **and Toru Yamaguchi**[*1]

Tokyo Metropolitan University, Hino Tokyo Japan[*1]

E-mail: [matsufuji-akihiro, erina-kasano]@ed.tmu.ac.jp, [eri, yamachan]@tmu.ac.jp

[Received 00/00/00; accepted 00/00/00]

Abstract. Emotion strongly influences our work performance, decision making, mental health, and relationship satisfaction in our daily life. To provide the supports and feedbacks that correspond to human emotion appropriately, emotion recognition technologies have been developing. Recent emotion recognition methods are developed with sophisticated machine learning models using a large amount of labeled data. However, the accuracy of these models has been insufficient even of the sophisticated models. According to the field of psychology, the important features of emotion recognition/expression are different among people. We assumed that these models that discover a relationship pattern between features and emotion state using labeled data are suffered from individual differences. Therefore, we integrate the multiple models learned from each person and each feature as knowledge and use appropriate models to infer the emotion of a new user.

In this study, we focus on training a model by incrementally using new data obtained from the new user. In the practice of dealing with various individual differences or domains, our architecture flexibly increments and updates with new data.

We evaluate the performances of the previous integration method and our architecture in the scenario of training a new user model incrementally and stocking multi-models learned from each user and each modal.

Keywords: affective computing, human robot interaction, non-verbal, multi-modal learning, individual difference.

1. Introduction

Analysis and modeling of human behavior have become less difficult nowadays because of the vast amount of data generated by users on the internet such as images, videos, comments, and views that hold rich information about the user themselves. Using multi-modal information can help us to design better human-human and human-machine interaction. According to Vinciarelli and Mohammadi [1], any technology involving understanding and prediction of human behavior is likely to benefit from the personality computing approach. Particularly, human emotion understanding and prediction have been attracting attention in many fields (e.g., human-robot interaction, emotion regulation). Emotion powerfully influences our work performance, decision making, mental health [2]; human-machine interaction considering emotion from multi-modal data has great benefit in our daily life. Unfortunately, the accuracy of emotion understanding and prediction has been insufficient even of the sophisticated machine learning models. According to the field of psychology, the important features of emotion recognition/expression are different among people. We assumed that these models that discover a relationship pattern between features and emotion state using labeled data are suffered from individual differences. Furthermore, in the practice of training the multi-modal system, the dataset including parallel data of all modalities is extremely rare, because the vast amount of labeled emotion data is not on the internet, unlike the above-mentioned human behavior data.

The data of multiple modalities often come from different people in different situations. Thus, it needs to consider the method of organization of the non-parallel dataset to train a multi-modal system for predicting a specific person considering the individual difference.

Furthermore, our model takes advantage of the knowledge about individual differences from other people for predicting a new user, unlike a one-size-fits-all model that calculates general knowledge among people.

Therefore, we proposed the method that integrates the multi-models learned from each person and each feature as knowledge and use appropriate models to infer the emotion of a new user.

Previous work [14] showed this method could have the potential to achieve high performance and have flexibility predicting various domains that include individual differences. However, there is a premise that the architecture has already obtained the trained model which appropriate to predict the new data. In this study, we focus on training a model by incrementally using new data obtained from the new user. In the practice of dealing with various individual differences or domains, our architecture flexibly adds and updates with new data. We evaluate the performances of the previous integration method and our architecture in the scenario of incrementally using a new user model and stocking multi-models learned from each user

and each modal.

2. Related studies

In this section, we described the related studies of multi-model learning. First one, multi-modal learning is a traditional method to combine multi-modal information in one-size-fits-all models or metamodels to predict complex situations. Second, ensemble learning is to combine the multiple machine learning model to achieve high performance. These method separates machine learning models and combines these models using metamodel. Some of these methods covered each other; we referred the architecture using multi-model and metamodel to build our multi-modal and multi-person architecture.

2.1. Multi-modal learning

Techniques for multi-modal learning have long been investigated by the research community [3], [4]. Traditionally, there are two approaches for combining the signals of multiple sensors.

2.1.1. Early fusion

One approach is called early fusion or feature-level fusion. Feature-level fusion involves how to integrate the multiple modalities into a single feature vector, before being used as input to a machine learning algorithm. For example, the simplest form of early fusion involves concatenation of different multi-modal features, as illustrated in Fig.1.

Most feature-level fusion models make the assumption that there is conditional independence between different modalities, which sometimes may not be true in practice, as multiple modalities tend to be highly correlated. But [5] also argues that different streams contain information that is correlated to another stream only at a high level, which supports the output of each modality to be processed independently of the others. However, feature-level fusion still has mainly two difficulties. First is that feature-fusion could be quite challenging if the data to be fused is raw due to the mismatched sampling rate and forms of representation. Secondly, it is easy to contain redundant information when applying feature level Typically, dimensionality reduction techniques like PCA, autoencoders are applied to remove these redundancies in the input space.

2.1.2. Late fusion

The other is called late fusion or decision-level fusion, which refers to the aggregation of decisions from multiple classifiers, each trained on separate modalities, as illustrated in Fig. 2. Decision-level fusion was popular within the machine-learning community in the early- to mid-2000s, people favor this fusion approach because it doesn't have the shortcoming of asynchronicity like early fusion, therefore, is easier to implement and it doesn't depend on the representations of different modalities. There

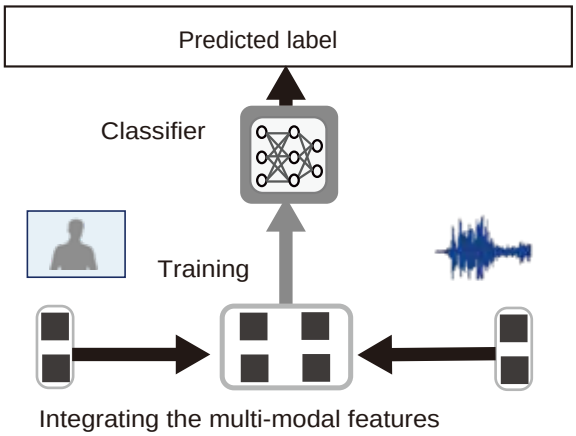

Fig. 1. : An illustration of early fusion for multi-modal learning

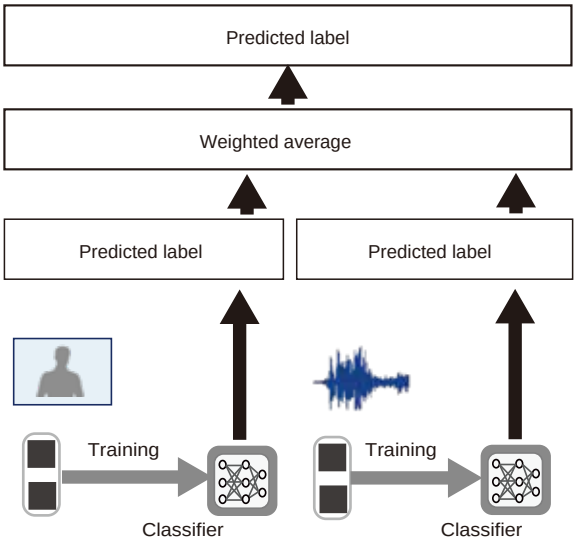

Fig. 2. : An illustration of late fusion for multi-modal learning

are various ways to determine how the decisions from different modalities are combined to a single one, which could be max-fusion, averaged-fusion with referring to the ensemble learning architecture. Although decision-level fusion is easier to implement, it is not necessarily better than feature level fusion. Because the classifiers (or regressors) in decision-level fusion are relatively rigid, it is likely to miss information important for the final decision during the processing of each modality and the performance of feature-level fusion heavily depends on the task.

2.2. Multi-model architecture

In this subsection, we described the detail of combining multiple machine model studies. These studies are

referred to in late fusion methods in multi-modal learning. Our proposed architecture which separates the models learned each person could be categorized in multiple machine learning studies. With regard to utilizing multiple machine learning methods, the architecture of ensemble machine learning methods [8] uses meta-algorithms that combine several machine learning techniques into one predictive model to decrease the variance (bagging) and bias (boosting), or improve the predictions (stacking). Bagging is a term indicating bootstrap aggregation. One way to reduce the variance of an estimate is to average multiple estimates together [9]. Bagging uses bootstrap sampling to obtain the sample data for training the base learners. For aggregating the outputs of the base learners, bagging uses voting for classification and averaging for regression. In a random forest [10], each tree in the ensemble is built from a sample drawn with replacement from the training set. In addition, instead of using all features, a random sample data of features is selected. The bias of the forest increases slightly, but owing to the averaging of less correlated trees, its variance decreases, resulting in an overall improved model. Boosting [11] refers to a family of algorithms that are able to convert weak learners into strong learners.

The predictions are then combined through a weighted majority vote (classification) or a weighted sum (regression) to produce the final prediction. As the principal difference between boosting and committee methods, such as bagging, base learners are trained in a sequence on weighted data. The advantage of ensemble methods is typically an out-performance of any machine learning technique. However, ensemble learning has two limitations for satisfying our research aim.

3. Motivation

We aimed to estimate whether participants will answer a question with confidence or not as one of the emotional states during a communication. For example, when educational tutoring robots ask students to give a correct answer, the robot can assess whether the students genuinely understood the question if it can estimate that they answered it with confidence [12, 13].

According to Merabian [6] based on the field of psychology, non-verbal information accounts for 93% of human-human communication. Thus, recent related technologies consider non-verbal behaviors to estimate the emotional states of humans. Automated analysis of human behavioral cues relies on machine learning models from sensory inputs (e.g., cameras, and microphones) capturing various modalities (face, body, and voice) of human behaviors. In this study, we used acoustic and motion modalities as the non-verbal features to estimate that they answered it with confidence.

In our previous studies, we proposed a prototype multi-model architecture and mentioned the importance of considering the individual differences in humans. Specifically, the previous study [14] mentioned that the performance of the model was dependent on the training and test samples, and also mentioned the various types of trained samples prepared to infer the state of the test sample. This prototype architecture only outputs the highest prediction parameter from each model output. These studies simply prepared the several machine learning models and using the max voting method for predicting the internal human state. The study [15] described the adaptive weighting of these prepared models to use appropriate models with new sample data. However, this study required a few labeled data of the obtained new sample data from the human-robot interaction. Furthermore, there is a premise that the architecture has already obtained the basic model that trained similar individual differences information with the new sample data; previous work focuses on finding appropriate models by calculation of the similarity between data used training and data obtained as a new sample data. Thus, the performance is depending on stock models, and it would not predict effectively new sample data which is not similar to data that used to train models. In addition, the adaptation is difficult to work in the situation of short term interaction that robots or agents could not obtain labeled data from new sample data. Considering the above-described real situation of dealing with various individual differences or domains, in this study, we focus on the method to add a model that trains new sample data obtained from the new user.

4. Proposed method

In regard to our multi-model architecture, we first generated the machine learning models for data about each modality information of user A. Dataset of person A is denoted as $D_A = (X_{1,1}, X_{1,2}, ..., X_{1,m}, y_1), ..., (X_{i,1}, X_{i,2}, ..., X_{i,m}, y_i)$, where $(X_{i,m}, y_i)$ is one data instance, $X_{i,m}$ is the data attributes, y_i is the label data, and m represents the number of multi-modal information. Classifier training user A data also represents as $C^t_{A,1}, C^t_{A,2}, ..., C^t_{A,m}$ referring the late fusion that is one of multi-modal learning architecture. In our architecture, we also stocked multi-person models, where t is the times of updating. Following the training of user A, the remaining users' dataset are used to training each classifier. Given dataset of users which is from user A to user J denoted as $A, B, ..., J \in P$ and a number of modal is $1, 2 \in M$, a set of classifiers in our architecture denoted as $E^t = (C^t_{A,1}, C^t_{A,2}, C^t_{B,1}, C^t_{B,2}, ..., C^t_{J,1}, C^t_{J,2})$, where E^t is our architecture and t is times of updating and incrementally add the base classifiers.

If the dataset of user K denoted as $D_K = (X_{i,1}, ..., X_{i,m}, y_i)$ newly arrived, all classifiers predict the labels using newly arrived dataset of user K. Given the number of instances in the newly arrived dataset denoted as $i \in I$, and the number of user who trained in our architecture represented as $p \in P$ and the number of modals is $m \in M$ respectively. The calculation of our

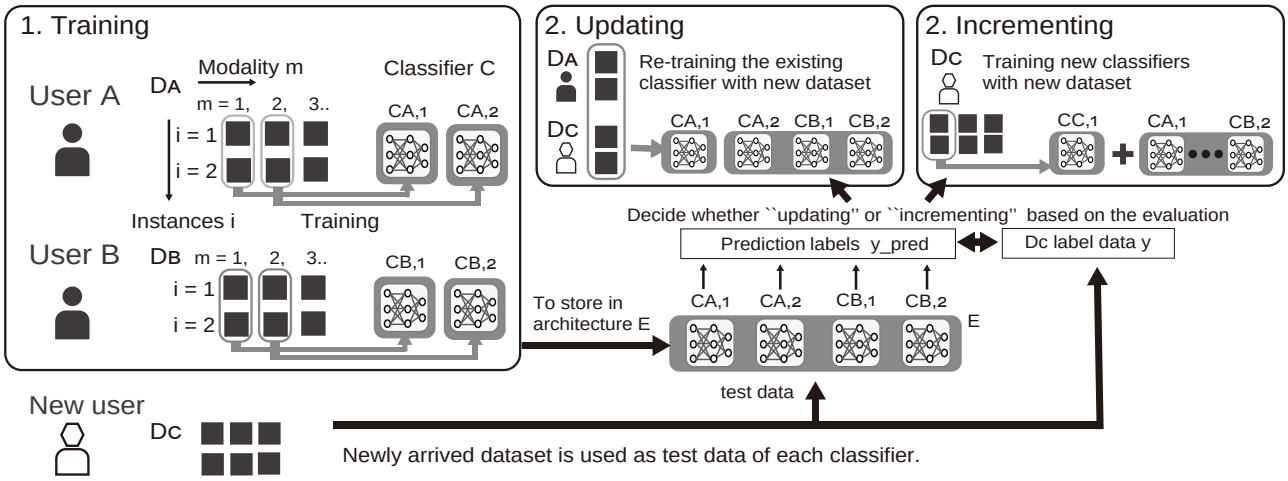

Fig. 3. : The procedure of our proposed method

architecture is illustrated below.

$$y_{pred,i} = \max_{p \in P, m \in M} C^t_{p,m}(X_{i,m})$$

The prediction label $y_{pred,i}$ is obtained by the max voting of all base classifiers of our architecture E using the instance $d_i = (X_{i,M}, y_i)$ in the newly arrived dataset.

We assume that the class label y_i of the instance d_i can be accessible after the prediction label $y_{pred,i}$ is made. We calculate the all $y_{pred,i}, i \in I$ using the newly arrived dataset of user K denoted as D_K . Therefore, the prediction labels $y_{pred,I}$ and the class label y_I is are used to calculate the performances of each base classifier in our architecture, $E^t = (C^t_{A,1}, C^t_{A,2}, C^t_{B,1}, C^t_{B,2}, ..., C^t_{J,1}, C^t_{J,2})$. In this study, the base classifier's classification accuracy (Acc.) is used as the performance of C^t_A, which is the rate of misclassified or correct classified instance $d_i \in D_k$ made by C^t_A. The confidence class is treated as a positive class, and the unconfidence class is treated as a negative class in our experiment settings. $Acc = \frac{TP+TN}{TP+FP+FN+TN}$; and the term of TP represents the number of instances correctly labeled as a positive class. FP represents the number of instances incorrectly labeled as a positive class. TN represents the number of instances correctly labeled as a negative class. FN represents the number of instances incorrectly labeled as a negative class.

The performance of the base classifier C^t_p that predicts the labels of newly arrived data will be compared with a threshold that manually defined performance. If the performance of one of the base classifiers achieved the threshold, the base classifier that achieved the highest performance of predicting labels of the newly arrived dataset incrementally update the base classifier denoted as C^{t+1}_p and the base classifier add to existing multi-model architecture E^t to construct new multi-model architecture E^{t+1}. Since it is assumed that it has already obtained knowledge of the user's individual differences.

If the performances of each classifier that are not achieved the threshold, the newly arrived data is assumed that it has new individual differences. the architecture train a new base classifier $C^0_{K,m}, m \in M$ based on dataset D_K, and adding $C^0_{K,m}, m \in M$ to existing multi-model architecture E^t to construct new multi-model architecture E^{t+1}.

Fig. 3 shows the illustration of the procedure of our proposed method. The classifiers $C_{A,m}, C_{B,m}, m \in M$ was trained with the existing dataset $D_A, D_B = (X_{i,m}, y_i), i \in I, m \in M$ that are from user A and user B in the illustration. The newly arrived dataset D_C of the new user were used as test data to evaluate the classification accuracy of existing classifiers $C_{A,m}, C_{B,m}, m \in M$. Each classifier's classification accuracy decided the incremental method that is whether updating the existing classifiers or incrementally add the new classifiers.

5. Experimental setup

5.1. Confident situation

We collected non-verbal information of the internal state of humans at previous study [16]. We set up situation in which the participants could feel confident or not as internal human states; an agent system was used to make the participants feel confidence or not. This section describes the detail of collected dataset in previous study and used in this study. The experiment involved 11 participants who are 21-26 years in age. The experimenter asked each participant to answer 50 questions in several fields. Along with difficult questions, easy versions were also asked to allow for quick answers, and the participants were asked to provide an answer even if the questions were difficult. If the participants answer awkwardly because of the difficulty of the given questions, we define this situation as an unconfident situation. Conversely, if the participants felt easy to answer the question, we defined the situation as a state of confident. In the experiment, for data collection, we gave both easy

and difficult questions of which the participants felt confident and unconfident, respectively. We aimed to apply a condition in which participants could talk with a robot and/or an agent, not a human. We utilized MMD Agent [17], which is a toolkit for building voice interaction systems. To avoid an overlap of answer of the participants and the MMD Agent's utterances, the participants were instructed to answer each question after the MMD Agent had terminated the quety. We also asked the participants to not answer with "I don't know". Because we created a condition in which the participants could feel confident or not based on the difficulty of questions; if the participants answered "I don't know" with confidence, we would be unable to create a condition in which participants did not feel confident. Afterward, they filled out a questionnaire with scores of 1 ("I did not have the confidence to answer this question") points to 5 ("I had the confidence to answer this question") points to grade their confidence about the answer they gave for each question. This questionnaire was created on a 5-point Likert scale. All participants agreed to the use of the data collected during this experiment for research purposes.

5.2. Dataset

The data on the participants' behaviors were collected in this study by using a motion tracking camera (Kinect V2, Microsoft) and a microphone (ICD-SX1000, Sony). The data were collected when the participants answered a question. The motion tracking camera captured the skeleton data, which can be obtained by locating the joints of the tracked participants and tracking their motion. Our system aimed to capture the motion of a participant seated in a chair. The illumination used in the experiment is similar to that of general homes. To prevent an erroneous recognition of another person, the experiment settings allow only one participant within the field of view of the motion tracking camera. The frame rate is 15 fps.

5.2.1. Acoustic modality

To analyze the recorded audio, we used Praat [18], a voice analysis software. We manually segmented the audio according to the utterance of each answer. The audio information is composed of eight pieces of numerical data on effective sound pressure i.e., the maximum sound pressure, minimum sound pressure, sound pressure range, average pitch, maximum pitch, minimum pitch, and pitch range were obtained for the data cut using Praat software. The eight acoustic feature values obtained from the questionnaire were tagged for two classes, confident and unconfident.

5.2.2. Motion modality

We analyzed the recorded motion as captured three-dimensional skeleton information. In previous studies on motion information, head motion is one of the most prominent social signals, in either human-human [19] or human-robot [20, 21] interactions. The results of relative studies have proven that data on the head motions shows the clearest results [22]. Thus, we used the movement of the coordinates of the head and presented an inter-frame difference. Head motion information is composed of six pieces of numerical data of three-dimensional head rotations and three-dimensional head translations.

5.2.3. Binary classification labels

We used only 1 ("I did not have the confidence to answer this question") and 5 ("I had the confidence to answer this question") points from the questionnaire as binary classification labels. Because the binary classification labels used in this research for confident and unconfident situations are ambiguous information even for the participants themselves, we aimed to collect data on the situations in which the participants recognized an internal state clearly on their own by using questionnaires made on a 5-point Likert scale. Specifically, this was applied to remove ambiguous data that occurred when the participants assessed 2, 3, or 4 points in the 5-point scale questionnaire and self-reported data. The dataset was split into training samples and test samples according to the participants. In this experiment, we used 10 of the 11 participants for the training data. The remaining participant was used as the source of the test data. To test the performance of our prediction model, we change the training data and test data for each participant. Specifically, each participant will be used as the test data and we trained the other participants in all patterns.

6. Experimental evaluation

In this section, we evaluate our proposed method on the experiment that is a multi-modal or person dataset. Each user-related dataset is used as training data and test data to evaluate our proposed method and baseline models.

6.1. Settings of proposed method

At the initial stage of evaluation, we randomly selected the seven users' denoted as A to G as an initial training dataset. This initial training dataset is assumed as existing components in multi-model architecture before incrementally adding and updating models in practical situations. An initial training dataset is used to create the first base classifiers $C^0_{A,1}, C^0_{A,2}, C^0_{B,1}, C^0_{B,2}, ..., C^0_{G,1}, C^0_{G,2}$ to the initial multi-model architecture E^0. Then each instances of the user H dataset are used as newly arrived data one by one. The prediction labels $y_{pred,i}, i \in I$ are obtained by the max voting of all classifiers. The prediction labels $y_{pred,i}$ and the class labels y_i are compared to calculate each base classifier's classification accuracy(Acc.). In the case of the performance of any base classifiers in our architecture E^0 exceeds the threshold, the base classifier C^0_p, where p is the user number and updated by using newly arrived dataset of user K. The updated base classifier is denoted as C^1_p and to our architecture E^1. If any base classifier could

not achieve the threshold, the new base classifier C_K^0 is add to our proposed architecture E^1. We set the threshold as 0.80 in this evaluation. Subsequently, the dataset of following users, user I to K, are used as test data and evaluate the classification accuracy for evaluating a situation that basic models are incrementally added and our model.

6.2. Base classifiers in multi-model architecture

The base classifier model of all algorithms was set as a multilayer perceptron. Regarding the parameters of the multilayer perceptron, the momentum is 0.2, the learning rate is 0.3, and the epoch is 500. It was implemented based on the scikit-learn [24], which is a popular open source framework for machine learning. To evaluate the prediction accuracy of each model, we conducted a cross-validation [23]. In this experiment, we prepared the base classifiers of three types including audio, motion, and mixed modality. The mixed modality is set as an early fusion method in multi-modal learning. Our proposed architecture employed the max voting method in ensemble learning to select the appropriate model to predict the labels. Thus, mixed modality is also important to deal with the individual differences that appears features in both modality (audio and motion).

6.3. Baselines for comparison and measurement for evaluation

To evaluate our proposed algorithm, we compare it with a learning algorithm that handle the multi-modal features that is late fusion methods.

For evaluation the performance, we used the one of late fusion method, stacking ensemble learning model which integrates the multiple models by weighted voting [9]. These weights are originally defined in the training phase, and it is also difficult to deal with individual differences by the original ensemble learning model. To make a fair comparison, we set multi-modal and person base classifiers in the conventional stacking ensemble method. This method only adds the newly arrived data as new base classifiers and calculates each weight, unlike our proposed method that decides whether to upgrade an existing base classifier or add new base classifier. This is assumed as a simple method incrementally adds newly arrived data by simple weight definition without our proposed method.

6.4. Results and consideration

Fig. 4 shows the classification accuracy of the different learning algorithms evaluated in the human confidence dataset. The classification accuracy at each point that new user data arrived is listed based on prediction performance of each user. The deciding "update the exist base classifier" and "incrementally add the new base classifier" in our architecture depending on the accuracy of existing classifiers of new data instance d_i. The "incrementally add the new base classifier" occurs at $4th, 5th, 7th$ user-related dataset arrived. The classification accuracies are gradually added with fluctuation.

The classification accuracy of all learning algorithms dropped after $7th$ and $9th$ dataset that has new individual differences arrived. The result shows the $7th$ newly arrived dataset has brand new individual differences in this evaluation scenario. The accuracy of our proposed method could recover by adaptively adding the newly arrived data. This result also reconfirmed us there are individual differences in the human internal state.

The pros of the stacking ensemble learning model are to deal with a brand new user-related dataset that has a new individual difference without dropping the performance immediately. The weighted average could work to smooth out any inequalities of prediction results of each base classifier in this scenario that is dividing the dataset based on users. However, the classification accuracy of the stacking ensemble learning model could not achieve 0.85 points although the classification accuracy of our proposed method could achieve 0.85 points when $6th$, $8th$, $10th$ dataset arrived. It seems that the weights of ensemble learning could not deal with our evaluation scenario that is to deal with the data that has a new domain or individual differences and it works to smooth out any inequalities to avoid dropping the performance. Since original stacking ensemble learning divides the dataset into sub dataset randomly to prepare the different variance and bias as described in related studies. Our concept is to cover with original ensemble learning concepts but separating the dataset based on individual differences is not completely represented by using the difference of variance and bias in the dataset.

We conducted further analysis of the pros of incrementally adding and updating models in our proposed method. We compared it with the same architecture that does not update the existing models, only add the new base models based on newly arrived data. In other words, the originally proposed settings made $C_{p,m}^1$ by updating from $C_{p,m}^0$ with the newly arrived dataset, and the second settings stacked new base classifiers $C_{p,m}^0$ when new dataset arrived. In this detailed comparison, not only the performance of the multi-model architectures E^t but also the performance of each classifier $C_{p,m}^0$ or $C_{p,m}^1$. In the phase of the user $9th$ dataset arrived, the performance of $C_{user6,m}^0$ achieved the threshold of updating and $C_{(user6,user9),m}^1$ was made by updating the $C_{user6,m}^0$ with user 9 dataset. In the second set, the $C_{user9,m}^0$ is simply added in the architecture E^9. In this evaluation, we denoted the $p \in P$ as the number of users because we shuffled the user A to K. In the next phase, each classifiers process the test data of newly arrived $10th$ dataset. In the result, the classification accuracy (Acc.) of $C_{(user6,user9),m}^1$ achieved the highest score that is 0.934. It was shown in Fig. 4 as the output of our proposed method. On the other hand, the classification accuracy of $C_{user6,m}^0$ and $C_{user9,m}^0$ are 0.891 and 0.869, respectively. This result shows that our incrementally added and updating method worked to make the flexible multi-model architecture to deal with individual differences. Furthermore, the output of classifiers $C_{(user6,user9),m}^1$, $C_{user6,m}^0$ and $C_{user9,m}^0$ revealed user 10 data has the same feature of individual difference

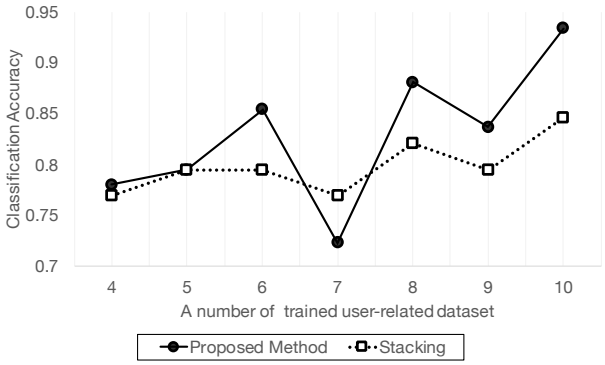

Fig. 4. : Classification Accuracy over sequential person data streams.

with user 6 and 9 data.

7. Conclusion

In this study, based on the advantage of different types of multi-model and modal learning algorithms, we proposed an incrementally multiple-model method in multimodal and person architecture to deal with data involving individual differences. The individual differences in each person could be called as small domains which covered partly with each other. The novelty of our architecture is its integrated multi-model that has the bias referring using late fusion architecture and each model is constructed based on individual differences of person. The decision method of two approaches which are updating the existing base classifier in our architecture or incrementally adding a new base classifier helps our architecture to accommodate to stock a variety of individual differences with multi-modal knowledge and reduce duplicate knowledge and the computation costs. The experiment indicated that the classification accuracies are gradually increased with fluctuation and our proposed architecture could the accuracy of our proposed method increase by incrementally model the newly arrived data. Furthermore, the analysis of the base classifiers that are updated and simple adding models situation shows our method could work in the scenario to deal with individual differences.

In our future research, we will improve our algorithm by equipping it with the ability of dynamic weighted voting from ensemble learning. Furthermore, it will enable it to take advantage of existing base classifiers' knowledge about individual differences not only the classifier that predicts the highest score.

References:

[1] A. Vinciarelli, G. Mohammadi, "A survey of personality computing," In IEEE Transactions on Affective Computing Vol.5, No.3, pp. 273–291, 2014.

[2] S. L. Koole, K. Rothermund, "i feel better but i don't know why," The psychology of implicit emotion regulation. Cognition and Emotion, Vol.25, No.3, pp.389–399, 2011.

[3] P. K. Atrey, M. A. Hossain, A. El. Saddik, M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," Multimedia systems, Vol.16, No.6, pp.345–379, 2010.

[4] B. Khaleghi, A. Khamisa, F. O. KarrayaSaiedeh, N. Razavib, "Multisensor data fusion: A review of the state-of-the-art," Information fusion, Vol.14, No.1, pp.28–44, 2013.

[5] N. Sebe, I. Cohen, A. Garg, Th. S. Huang, "Machine learning in computer vision," Springer Science and Business Media, Vol. 29. 2005.

[6] A. Merabian, "Silent Communication," Wadsworth, Belmont, California, 1971.

[7] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A. Y. Ng, "Multimodal deep learning", In Proceedings of the 28th international conference on machine learning (ICML-11), pp. 689–696, 2011.

[8] TG. Dietter, G. Thomas, "Ensemble methods in machine learning," In Multiple classifier systems, Vol. 1857, pp.1–15, 2000.

[9] S. Quan P. Bernhard, "Bagging ensemble selection," AI 2011: Ad vances in Artificial Intelligence, pp.251–260, 2011.

[10] L.Breiman, "Random forests," In Machine learning, Vo. 45, No. 1, pp.5-32, 2001.

[11] D. G. Thomas, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization", In Machine learning, No. 40, Vol. 2, pp.139–157, 2000.

[12] E. Marinoiu, M. Zanfir, V. Olaru, G. Sminchisescu, "3d human sensing, action and emotion recognition in robot assisted therapy of children with autism," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition ,pp. 2158–2167, 2018.

[13] E. Lakomkin, M. A. Zamani, C. Weber, S. Magg, S. Wermter, "On the Robustness of Speech Emotion Recognition for Human-Robot Interaction with Deep Neural Networks," In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 854–860, 2018.

[14] A. Matsufuji, T. Shuyu, E. Kasano, W. F. Hsieh, Y. Ho, E. Sato-Shimokawara, L. H. Chen, T. Yamaguchi, "Multi Characteristic Model Architecture for Estimating Human Internal State", The 6th International Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII 2019), SAT2-B5, 2019.

[15] A. Matsufuji, T. Shuyu, E. Kasano, W. F. Hsieh, E. Sato-Shimokawara, L. H. Chen, T. Yamaguchi, "Adaptive Multi Model Architecture by Using Similarity Between Trained User and New User," International Conference on Technologies and Applications of Artificial Intelligence(TAAI).IEEE, 2019.

[16] E. Kasano, S. Muramatsu, A. Matsufuji, E. Sato-Shimokawara, T. Yamaguchi, "Estimation of Speakers Confidence in Conversation Using Speech Information and Head Motion," the 16th international conference on ubiquitous robots, 2019.

[17] A. Lee, K. Oura, K. Tokuda, "MMDAgent - A fully open-source toolkit for voice interaction systems," Proceedings of the ICASSP 2013, pp. 8382-8385, 2013.

[18] P. Boersma, "A System for Doing Phonetics by Computer," Glot International, Vol. 34, No. 5, pp. 9-10, 2001.

[19] A. Vinciarelli, M. Pantic, B. Hereve, "Social signal processing: survey of an emerging domain, Image and Vision Computing", Vol. 27, No. 12, pp. 1743-1759, 2009.

[20] M. Giuliani, N. Mirning, G. Stollnberger, S. Stadler, R. Buchner, 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, pp. 931, 2015.

[21] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel, A. Waibel, "Natural human-robot interaction using speech, head pose and gestures," In Intelligent Robots and Systems.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Confidence Vol. 3, pp. 2422–2427, 2004.

[22] L. P. Morency, C. Sidner, C. Lee, T. Darrel, "Head gestures for perceptual interfaces: The role of context in improving recognition", Artificial Intelligence, Vol. 171, No. 8–9, pp. 568–585, 2007.

[23] D.M. Allen, "The relationship between variable selection and data augmentation and a method for prediction," in technometrics, Vol. 16, pp-125–127, 1974.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, J. Vanderplas, "Scikit-learn: Machine learning in Python," Journal of machine learning research, Vol. 12, 2825-2830. 2011.