10th International Scientific Conference
Technics, Informatics, and Education – TIE 2024
20-22 September 2024

Session: *Computer Sciences and Information Technology*
Professional paper
DOI: 10.46793/TIE24.060F

PsAIchology: An Intelligent Direction in Psychological Sciences

Hojjatollah Farahani[1*] [0000-0002-9799-7008], Peter Watson[2*] [0000-0002-9436-0693], Timea Bezan[3] [0000-0001-6938-6974], Nataša Kovač[4] [0000-0002-6671-2938], Lisa-Christina Winter[5] [0009-0007-1149-9061], Marija Blagojević[6] [0000-0003-4186-0448], Parviz Azadfallah[1] [0000-0003-2810-762X], Abbasali Allahyari[1] [0009-0006-5506-7888], Samira Masoumian[7] [0000-0003-3066-2577] and Paulino Jiménez[5] [0000-0002-8229-1141]

[1] Tarbiat Modares University/Psychology, Tehran, Iran
[2] MRC Cognition and Brain Sciences Unit, University of Cambridge, Cambridge, UK
[3] Singidunum University/Faculty of Informatics and computing, Belgrade, Serbia
[4] University of Donja Gorica, Faculty of Applied Sciences, Podgorica, Montenegro
[5] University of Graz /Department of Psychology, Graz, Austria
[6] University of Kragujevac/ Faculty of Technical Sciences Čačak, Serbia
[7] AJA University of Medical Sciences/Clinical Psychology, Tehran, Iran
[1*] h.farahani@modares.ac.ir,
[2*] Peter.Watson@mrc-cbu.cam.ac.uk

Abstract: *The intersection of machine learning (ML) and cognition is often referred to as 'artificial intelligence', whereas the intersection of psychology and ML is a term we would like to coin as 'Artificial Psychology' or "PsAIchology". The main purpose of this paper is to introduce three commonly used machine learning algorithms for mind research along with their R codes. This paper aims not only to introduce these methods for analyzing data but also tries to provide the answers for questions that may arise for a mind researcher including a) how to choose which algorithm needs to be used for a given dataset, b) How to implement them using R code, c) How to assess model performance to select the best performing algorithm and d) How to interpret the results of the ML algorithms obtained from fitting to a set of data. In this paper, we introduce and illustrate the most commonly used ML algorithms including, AdaBoost, Extreme Gradient Boosting (XGBoost), Random forest and give related R codes with the results obtained from running them. Finally, model performance is interpreted and discussed.*

Keywords: *artificial intelligence; artificial psychology; machine learning; psychology; R language*

1. INTRODUCTION

Psychologists are increasingly interested in adopting powerful computational techniques from the field of machine learning to accurately predict real-world phenomena [1]. PsAIchology or artificial psychology is a highly multidisciplinary field of study in psychology. PsAIchology tries to solve problems that occur when psychologists do research and need a robust analysis method. Conventional statistical approaches have deep-rooted limitations. These approaches are excellent on paper but often fail to model the real world. Mind researchers have been trying to overcome this by simplifying the models being studied. This stance has not received much practical attention recently. Promoting and improving artificial intelligence helps mind researchers to find a holistic model of mental models. This development achieves this goal by using multiple perspectives and multiple data sets together with interactive and realistic models. This comprehensive, holistic, and interactive view may lead to a new research line in the near future. AP can open up a new horizon for mind researchers from clinical to theoretical psychologists to find a more realistic model. This horizon is rooted in a multidisciplinary approach updating our view along with the development of the related sciences leading to the finding of new results even from old datasets and models. AP has some assumptions. Satisfying these assumptions helps find a more precise and deeper way of modeling for artificial psychologists. The assumptions of AP are discussed here. First, we assume that the mind is filled with uncertainty. It is important to note that uncertainty not only occurs in nature but also in almost all man-made systems. Second, we assume that the mind is continuous. In other words, we assume a continuous consciousness in which the brain acts holistically and outputs behaviors discretely [2]; therefore, there is not a sharp dividing line between emotion and cognition. The brain consists of grey matter that constructs mental systems not

separated by solid lines. These ambiguous areas are the ones mind researchers are trying to handle by the use of statistical models. The third assumption is that the mind is a complex system; human mentality is made up of complicated systems. Even the simplest systems are complex. This complexity can be captured and interpreted by a dynamic model. The fourth assumption is that there is always a link between mind and data. It is not possible to study mental activities directly. Brain data needs to connect to some psychological constructs and behaviors. Therefore, we need to use multiple sources of data in a single model at the same time. Conventional statistical techniques use rigorous mathematical models. These models require large amounts of data for analysis and prediction. In the real world, we are facing big, imperfectly measured data as well as nonlinear relationships in complex systems. The fifth assumption is that brain data is highly dimensional data. This implies that the dataset has many features even in small sample sizes. This problem commonly occurs in psychological research, especially in clinical, cognitive psychology, and neuroscience, where we need to deal with P > n.

1.1. What Is PsAIchology?

In summary, psychologists need new analysis models to help them model complex mental systems. PsAIchology uses intelligent models that satisfy these assumptions. One technique used in applied computing is to emulate the strategies involved in the intelligent systems or models for problem-solving. Intelligent models are related to the human way of thinking and interpretation. These models use fuzzy logic, artificial intelligence, and genetic algorithms both individually or together. Artificial psychology was first proposed by Dan Curtis in 1963 as a theoretical discipline. PsAIchology is a combination of psychology and artificial intelligence. APsychology can be called the science of studying an individual's mental processes and behaviors. Artificial intelligence also has a wide variety of definitions; however, it is a science that deals with the design of intelligent machines and systems; systems that can perform tasks requiring human intelligence [3]. PsAIchology uses artificial intelligence to design, train, test, and ultimately deploy methodological models in psychology using. features borrowed from psychology and artificial intelligence. PsAIchology in this paper is used to derive robust, interpretable, and explainable models for prediction.

Machine learning, especially supervised machine learning, is one of the most important sections of PsAIchology. Machine learning is a subfield of artificial intelligence that specializes in using data to make predictions or support decision making [4]. In psychology and other social or behavioral sciences, machine learning (ML) has started to emerge as an essential set of tools in predictive modeling to potentially increase the generalizability of findings [1, 5]. ML can. Over the past decade, supervised machine learning (ML) has appeared with increasing frequency in psychology and other social sciences. In psychology, ML has been used to tackle such diverse topics as predicting psychological traits from digital traces of online and offline behavior [6, 7, 8], modeling consistency in human behavior [9], and investigating the empirical structure of self-regulation [10]. Traditional psychological research aims to establish causal effects of predictor variables on outcome variables, whereas machine learning projects aim to achieve maximal (unbiased) accuracy when predicting outcome variables. The most commonly used supervised algorithms will be reviewed along with R code.

1.2. How to choose the best machine learning algorithm?

1-Past experience

A PsAIchology researcher can apply his/her experience from the past in dealing with similar problems or from reviewing the literature.

2-Trial and Error

Various algorithms are fitted to find which one is most appropriate for the given dataset. These algorithms can be improved using optimization with a tuning process or by hybridizing some related algorithms (known as spot-checking). In the context of machine learning, spot-checking refers to the process of testing a suite of standard algorithms on your dataset to establish a performance baseline. This helps identify a few top-performing models that can then be tuned and optimized further. For an effective spot-checking process, here are some recommended practices:

- Splitting the data into training and test data (hold-out method): if the data is huge, then a massive amount of data is needed to make the model accurate.
- Cross-validation divides the data into 5-10 subsets(folds) of almost equal size. Out of these folds is used as a validation set, and the others are involved in training the model. -We generate overall prediction error by taking the total of prediction errors across the folds
- For small amounts of data, we can repeat the K-fold approach multiple times. R code for repeating 10-fold validation 3 times.
- Leave One Out Cross-Validation (LOOCV), this method also splits the dataset into 2 parts but it overcomes the drawbacks of the Validation set approach. LOOCV carries out the cross-validation in the following way: a)firstly we train the model on N-1 data points, b) then test the model against the one data point which was left out in the previous step , c) we calculate prediction error for that one point, d)we repeat the above 3 steps until the model is trained

and tested on all data points, d)we generate overall prediction error by taking the total of the prediction errors. We now present some of the most commonly used supervised machine learning algorithms.

2. ADABOOST AND XGBOOST ALGORITHMS

Boosting is a powerful machine learning algorithm that helps improve the accuracy of models in data science. The AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned, with higher weights assigned to incorrectly classified instances. This algorithm builds a model and gives equal weights to all the data points. It then assigns higher weights to points that are wrongly classified. It will keep training models until the lowest error rate is achieved. The AdaBoost algorithm, introduced by Freund and Schapire in 1997, revolutionized ensemble modeling. Since its inception, AdaBoost has become a widely adopted technique for addressing binary classification challenges. This powerful algorithm enhances prediction accuracy by transforming a multitude of weak learners into robust, strong learners. Extreme gradient boosting (xgboost) is similar to the gradient boosting framework but is more efficient. It has both a linear model solver and tree learning algorithms. XGBoost is used both in regression and classification as a go-to algorithm. As the name suggests, it utilizes the gradient boosting technique by adding more and more weak learners until no further improvement can be made. Its parameters include the number of boosting iterations, referred to as nrounds, the learning rate (also known as eta). Eta is a hyperparameter that scales the contribution of each tree in the ensemble, the maximum depth of a tree, which is a pruning parameter designed to control the overall tree depth. Gamma is the minimum loss reduction required to make a further partition on a leaf node of the tree. Subsample is the fraction of all observations to be randomly sampled for each tree. Lambda is the L2 regularization term and alpha is the L1 regularization term. The R code for XGBoost is it is shown at Fig.1

```r
library(gbm)
library(caret)
parts = createDataPartition(P$Painintensity, p = 0.7, list = F)
train = P[parts, ]
test = P[-parts, ]
# train a model using our training data
model_gbm = gbm(Painintensity ~.,
                data = train,
                distribution = "multinomial",
                cv.folds = 10,
                shrinkage = .01,
                n.minobsinnode = 10,
                n.trees = 500)       # 500 tress to be built

summary(model_gbm)
#use model to make predictions on test data
pred_test = predict.gbm(object = model_gbm,
                        newdata = test,
                        n.trees = 500,           # 500 tress to be built
                        type = "response")
pred_test
# Give class names to the highest prediction value.
class_names = colnames(pred_test)[apply(pred_test, 1, which.max)]
result = data.frame(test$Painintensity, class_names)

print(result)
conf_mat = confusionMatrix(test$Painintensity, as.factor(class_names))
print(conf_mat)
```

Figure 1. *R code for XGBoost*

3. RANDOM FOREST

A Random Forest Algorithm is a supervised machine learning algorithm that is extremely popular and is used for Classification and Regression problems in Machine Learning. The greater the number of trees in a Random Forest Algorithm, the higher its accuracy and problem-solving ability. Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model. The following steps explain the working Random Forest Algorithm, Step 1: Select random samples from a given data or training set, Step 2: This algorithm will construct a decision tree for every training data, step 3: Voting will take place by averaging the decision tree, step 4: Finally, select the most voted prediction result as the final prediction result. This combination of multiple models is called Ensemble. Ensemble uses two methods: 1) Bagging creates a different training subset of data from sample training data with replacement. The final output is based on majority voting.2) Boosting combines weak learners with strong learners by creating sequential models to give higher accuracy in the final model.

ADA BOOST, XG BOOST are boosting algorithms, which were discussed in the previous section.

The R codes for Random forest is presented at Fig. 2.

```r
library('Metrics')
library('randomForest')
library('ggplot2')
library('ggthemes')
library('dplyr')
library('DALEX')
library("caret")
P <- read_sav("P.sav")
P$Painintensity<-as.factor(P$Painintensity)
P$Time<-NULL
#dividing the dataset into train and test
index <- createDataPartition(P$Painintensity, p = 0.7, list = FALSE)
train <- P[index, ]
test  <- P[-index, ]
control = trainControl(method="repeatedcv", number = 5, savePredictions=TRUE)
# Random Forest
mod_rf = train(Painintensity ~ .,data = train, method='rf', trControl = control)
library("DALEX")
# Create a custom predict function
p_fun <- function(object, newdata){
  predict(object, newdata=newdata, type="prob")[,2]}
# Convert the outcome variable to a numeric binary vector
yTest <- as.numeric(as.character(test$Painintensity))
# Create explainer objects for each machine learning model
explainer_rf  <- explain(mod_rf, label = "RF",
                         data = test, y = yTest,
                         predict_function = p_fun,
                         verbose = FALSE)
mp_rf  <- model_performance(explainer_rf)
mp_rf
vi_rf <- variable_importance(explainer_rf, loss_function = loss_root_mean_square)
vi_rf
plot(vi_rf)
```

Figure 2. *R code for Random forest*

4. ARTIFICIAL NEURAL NETWORKS

Artificial neural network often known as a neural network or simply a neural net, is a machine learning model that takes its cues from the structure and operation of the human brain. It is a key element in machine learning's branch known as deep learning. Interconnected nodes, also referred to as artificial neurons or perceptrons, are arranged in layers to form neural networks. An input layer, one or more hidden layers, and an output layer are examples of these layers. A neural network's individual neurons each execute a weighted sum of their inputs, apply an activation function to the sum, and then generate an output. The architecture of the network, including the number of layers and neurons in each layer, might vary significantly depending on the particular task at hand. Several machine learning tasks, such as classification, regression, image recognition, natural language processing, and others, can be performed using neural networks because of their great degree of versatility.

In order to reduce the discrepancy between expected and actual outputs, a neural network must be trained by changing the weights of its connections. Optimization techniques like gradient descent are used to do this. In particular, deep neural networks have made significant advances in fields like computer vision, speech recognition, and autonomous driving. Neural networks have demonstrated an exceptional ability to resolve complicated issues. They play a key role in modern AI and machine learning due to their capacity to automatically learn and extract features from data.

A supervised neural network model is a type of machine learning model used for tasks where you have labelled data, meaning you know both the input and the corresponding correct output. In this model, you feed input data into layers of interconnected artificial neurons, which process the information and produce an output. During training, the model learns to adjust its internal parameters (weights and biases) to minimize the difference between its predictions and the actual labels in the training data. This process continues until the model can make accurate predictions on new, unseen data. Supervised neural networks are commonly used for tasks like image classification, speech recognition, and natural language processing, where the goal is to map inputs to specific categories or values.

4.1 Multi-Layer Perceptron Architecture

MLP (Multi-Layer Perceptron) is a type of neural network with an architecture consisting of input, hidden, and output layers of interconnected neurons. It is capable of learning complex patterns and performing tasks such as classification and regression by adjusting its parameters through training. The following is a detailed exploration of the MLP architecture:

- Input Layer: The input layer is where the MLP and dataset first engage with one another. A feature in the incoming data is matched to each neuron in this layer. For instance, each neuron

might represent the intensity value of a pixel in picture categorization. These unprocessed input values are to be distributed to the neurons in the next hidden layers by the input layer.

- Hidden Layers: MLPs have a hidden layer or layers that are present between the input and output layers. The main computations happen at these layers. Every neuron in a hidden layer analyzes the data that comes from the neurons in the layer above it. In the same buried layer, neurons do not interact directly with one another but rather indirectly via weighted connections. The hidden layer transformation allows the network to learn intricate links and representations in the data. The intricacy of the task might affect the depth (number of hidden layers) and width (number of neurons in each layer).
- Output Layer: The MLP's neurons in the output layer, the last layer, generate the model's predictions. The structure of this layer is determined by the particular task at hand. The probability score for binary classification may be generated by a single neuron with a sigmoid activation function. Multiple neurons, often with softmax activation, can give probabilities to each class in a multi-class classification system. When doing regression tasks, the output layer frequently just has a single neuron that can forecast a continuous value.

Each neuron applies an activation function to the weighted total of its inputs, whether it is in the input, hidden, or output layer. The sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU) are often used activation functions. The MLP modifies connection (synapse) weights during training using backpropagation and optimization methods like gradient descent. In order to reduce the discrepancy between projected and actual outputs, this method aids the network in learning and fine-tuning its parameters. MLPs are appropriate for a variety of machine learning and deep learning problems, from straightforward to extremely complicated, due to their flexibility in terms of the number of hidden layers, neurons per layer, and choice of activation functions.

4.2 MLP Classifier with its Parameters

The MLP Classifier, short for Multi-Layer Perceptron Classifier, is a neural network-based classification algorithm provided by the Scikit-Learn library. It's a type of feedforward neural network, where information moves in only one direction: forward through the layers. Below is a detailed explanation of the MLP Classifier and its parameters, which in return collectively define the architecture and behavior of the MLP Classifier:

- Hidden Layer Sizes [Parameter: hidden_layer_sizes]: An MLP neural network's hidden_layer_sizes parameter is a crucial structural element. It describes how the network's hidden layers are structured. Each element of the tuple that this parameter accepts represents the number of neurons in a certain hidden layer. The network contains two hidden layers, with the first having 64 neurons and the second having 32 neurons, for instance, if hidden_layer_sizes is set to (64, 32). The network's ability to recognize complex patterns and correlations in the data is greatly influenced by the choice of the number of neurons and hidden layers. In order to model complex data, deeper networks with more neurons may be more susceptible to overfitting.
- Activation Function [Parameter: activation]: Each neuron in the MLP's hidden layers is activated using a different activation function, which is determined by the activation parameter. The network can model intricate input-to-output mappings thanks to the non-linearity introduced by activation functions. There are numerous activation functions, each with their own special qualities. One well-liked option is "relu" (Rectified Linear Unit), which is both computationally effective and successful in reducing the vanishing gradient problem. Both "tanh" (Hyperbolic Tangent) and "logistic" (Logistic Sigmoid) are frequently used and have various uses.
- Solver for Weight Optimization [Parameter: solver]: The neural network's weights are updated during training using an optimization technique, which is determined by the solver parameter. To reduce the loss function of the network, several solvers use various methods. The "adam" algorithm, which combines ideas from RMSprop and Momentum, works well with large datasets and intricate models. The limited-memory Broyden-Fletcher-Goldfarb-Shanno optimization algorithm is used by "lbfgs," which is best for smaller datasets. The stochastic gradient descent algorithm, known as "sgd," adjusts weights based on random selections (mini-batches) of the training data at each iteration.
- Learning Rate [Parameter: learning_rate]: Each training iteration's weight updates are controlled by the learning_rate parameter. It is essential in establishing the training process's stability and rate of convergence. The learning rate might be "constant," "invscaling," or "adaptive," which can all have an impact on how it changes over time. It is crucial to choose the right learning rate since an extremely high rate might cause divergence or slow convergence, while a rate that is too low can cause very slow convergence.
- Maximum Iterations [Parameter: max_iter]: The max_iter parameter restricts

how many iterations can be made before the solver converges during training. Convergence is the condition at which further iterations of the network's weights do not appreciably lower the loss. The solution quits and returns the current findings, which might not be ideal, if it cannot converge within the predetermined limit. When max_iter is chosen appropriately, the training process is neither abruptly stopped nor overly prolonged, allowing the model to converge to the desired level. The R codes is presented at Fig. 3.

```
library("neuralnet")
library("NeuralNetTools")
library("ggplot2")
library("GGally")
library("caret")
library("gmodels")
library("dplyr")
NE <- read_sav("A1s.sav")
NE=data.frame(NE, stringsAsFactors = TRUE)
NE=data.frame(NE, stringsAsFactors = TRUE)
colname=c("CurrentMood","DepressionEpisodes","TraitDepression",
          "TraitRumination","BroodingRumination","ReflectionRumination",
          "ChronicAnxiety","ChronicStress",
          "ExtendedCategoricProportionbynonmissingresponses","SpecGroup")
colnames(NE)=colname
summary(NE)
sapply(NE, class)
NE$CurrentMood=as.numeric (NE$CurrentMood)
NE$DepressionEpisodes=as.numeric (NE$DepressionEpisodes)
NE$TraitDepression=as.numeric (NE$TraitDepression)
NE$TraitRumination=as.numeric (NE$TraitRumination)
NE$BroodingRumination=as.numeric (NE$BroodingRumination)
NE$ReflectionRumination=as.numeric (NE$ReflectionRumination)
NE$ChronicAnxiety=as.numeric (NE$ChronicAnxiety)
NE$ChronicStress=as.numeric (NE$ChronicStress)
NE$ExtendedCategoricProportionbynonmissingresponses=as.numeric (NE$ExtendedCategoricProportionbynonmissingresponses)
NE$SpecGroup<-as.factor(ifelse(NE$SpecGroup==1, "a", "b"))
levels(NE$SpecGroup)
index = createDataPartition(NE$SpecGroup, p = .6, list = FALSE, times = 1)
train =NE[index,]
test= NE[-index,]
CrossTable(NE$SpecGroup)
CrossTable(train$SpecGroup)
CrossTable(test$SpecGroup)
```

Figure 3. *R code for Artificial neural network*

5. DISCUSSION

In this paper we have illustrated applications of ML models. These represent the supervised learning branch of machine learning. This means that there is a training data set that learns patterns from labelled data e.g. group classification by experts. The algorithm learns about relationships in the data and then applies them to new data. Using training data removes bias in prediction from using the same data set to both train and test an algorithm. The training-test set approach is a particularly attractive and viable approach in the twenty-first century given the ever increasing numbers of large amounts of complete data that are collected on-line and by using electronic devices. With increasing computer power procedures to fit these models are becoming more commonly available in software such as R.

It should be noted unsupervised forms of machine learning are used in variable selection and in the clustering of unlabeled data sets [11, 12]. These algorithms discover hidden patterns or data groupings without the need for human intervention and also include variable association models such as unsupervised neural networks.

Farahani et al, 2023 [13] provides further applications of the machine learning models described in this paper using R.

As it is presented in [14] there are a lot of examples of using artificial intelligence for supporting mental health. There is huge potential in using Artificial intelligence in psychological practice. According to [14] "the advancement of AI technologies and their application in psychological practice have important implications that can be expected to transform the mental health care field". However, the main role is on professionals and the main challenge is related to an ethics in these research areas.

6. CONCLUSION

AI has been traditionally used in cognitive research. AI, for example, has been used to reproduce activities of the human brain [15] and improve the level of social intelligence [16].

The most important feature, however, that distinguishes humans from machines is that humans process external input from the world to stimulate different subjective emotional orientations such as satisfaction, dissatisfaction, love and dislike. An important advancement in recent years, therefore, has been the enablement of the computer to understand and express

emotions. Huang [17] has used computer modelling on 5,500 face frontal photographs to predict face attractiveness and found this model performed well. Lebedeva et al. [18] further showed that algorithms can be trained to identify beauty preferences using a limited number of images.

Jensen et al. [19] have shown an association between modern biomarkers which has led to a belief that AI can be used to diagnose autism and help autistic people to improve social, communication, and emotional skills [20].

The above show the range of areas AI is being used and its importance to the further understanding of the working of the human brain.

Artificial Intelligence is an area of increasing importance and is now being used in almost every field of science. It has the potential to accelerate the pace of scientific discovery by being trained to discover patterns in data that have not been apparent using traditional methods. The training of AI algorithms is facilitated by the internet which allows the collection of richer sources of data to allow more powerful and detailed predictions and revolutionise our understanding of the world around us.

ACKNOWLEDGEMENTS

This study was supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia, and these results are parts of the Grant No. 451-03-66 / 2024-03 / 200132 with University of Kragujevac – Faculty of Technical Sciences Čačak.

REFERENCES

[1] Yarkoni, T., & Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science*, 12, 1100–1122.

[2] Huette, S., Winter, B., Matlock, T., & Spivey, M. (2012). Processing motion implied in language: Eye-movement differences during aspect comprehension. *Cognitive Processing*, *13*(1), 193–197.

[3] Crowder, J., & Friess, S. (2010). Artificial neural emotions and emotional memory*. Ic-Ai*, 373–378.

[4] Raschka, S., & Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2 (3rd ed.). Birmingham, UK: Packt Publishing.

[5] Yarkoni, T. (2022). The generalizability crisis. Behavioral and Brain Sciences, 45, e1. https://doi.org/10.1017/S0140525X20001685

[6] Kosinski, M., Bachrach, Y., Kohli, P., Stillwell, D., & Graepel, T. (2014). Manifestations of user personality in website choice and behaviour on online social networks. *Machine Learning*, 95, 357–380

[7] Stachl, C., Pargent, F., Hilbert, S., Harari, G. M., Schoedel, R., Vaid, S., Gosling, S. D., & Bühner, M. (2020). Personality Research and Assessment in the Era of Machine Learning. *European Journal of Personality*, *34* (5), 613–631. https://doi.org/10.1002/per.2257

[8] Youyou, W., Kosinski, M., & Stillwell, D. (2015). Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences*, *112* (4), 1036–1040.

[9] Shaw, H., Taylor, P. J., Ellis, D. A., & Conchie, S. M. (2022). Behavioral consistency in the digital age. *Psychological Science*, 095679762110404. doi: 10.1177/09567976211040491

[10] Eisenberg, I. W., Bissett, P. G., Enkavi, A. Z., Li, J., MacKinnon, D. P., Marsch, L. A., & Poldrack, R. A. (2019). Uncovering the structure of self-regulation through data-driven ontology discovery. *Nature Communications*, 10, 2319. doi: 10.1038/s41467-019-10301-1

[11] Farahani, H., Azadfallah, P., Watson, P., Qaderi, K.,. Pasha, A., Dirmina, F., Esrafilian, B., Koulaie, N., Fayazi, N., Sepehrnia, N., Esfandiary, A., Abbasi, F. N., and Rashidi, K. (2023). Predicting the Social Emotional Competence Based on Childhood Trauma, Internalized Shame, Disability/Shame Scheme, Cognitive Flexibility, Distress Tolerance and Alexithymia in An Iranian Sample Using Bayesian Regression. *Journal of Child & Adolescent Trauma*, *16*(2), 351-363.

[12] Naeem, S., Ali, A., Anam, S., Ahmed, M. (2023). An Unsupervised Machine Learning Algorithms: Comprehensive Review. *International Journal of Computing and Digital Systems 13*(1), 911-921.

[13] Farahani, H., Blagojević, M., Azadfallah, P., Watson, P., Esrafilian, F. and Saljoughi, S. (2023). An Introduction to Artificial Psychology: Application Fuzzy Set Theory and Deep Machine Learning in Psychological Research using R. Springer Publishing

[14] Uses of Artificial Intelligence in Psychology. Available from: https://www.researchgate.net/publication/364454285_Uses_of_Artificial_Intelligence_in_Psychology [accessed Jul 07 2024].

[15] Nadji-Tehrani, M. and Eslami, A. (2020). A brain-inspired framework for evolutionary artificial general intelligence. *IEEE Trans. Neural Netw. Learn. Syst.* 31 5257–5271. 10.1109/TNNLS.2020.2965567

[16] Yang, G. Z., Dario, P. and Kragic, D. (2018). Social robotics—trust, learning, and social interaction. *Sci. Rob*. 3:eaau8839. 10.1126/scirobotics.aau8839

[17] Huang C. (2017). Combining convolutional neural networks for emotion recognition, in Proceedings of the 2017 IEEE MIT undergraduate research technology conference (URTC) (Cambridge, MA: IEEE;), 1–4. 10.1109/URTC.2017.8284175

[18] Lebedeva, I., Ying, F. and Guo, Y. (2022). Personalized facial beauty assessment: A meta-learning approach. *Vis. Comput.* 1–13. 10.1007/s00371-021-02387-w

[19]Jensen, A.R., Lane, A.L., Werner, B.A., McLees, S.E., Fletcher, T.S. and Frye, R.E. (2022) Modern Biomarkers for Autism Spectrum Disorder: Future Directions Mol Diagn Ther. *26*(5): 483–495. doi: 10.1007/s40291-022-00600-7

[20]Chaddad, A., Li, J., Lu, Q., Li, Y., Okuwobi, I.P., Tanougast, C., Desrosiers, C., and Niazi, T. (2021) Can Autism Be Diagnosed with Artificial Intelligence? A Narrative Review Diagnostics (Basel) *11*(11): 2032.