Human Activity Recognition using Machine Learning Techniques

Vasileios Nastos*, Alexandros Arjmand*, Klevis Tsakai*, Dimitrios Dimopoulos*†, Dimitrios Varvarousis†,
Alexandros Tzallas*, Nikolaos Giannakeas *, Avraam Ploumis†, Christos Gogos *
*Department of Informatics and Telecommunications, University of Ioannina, Arta, Greece 47100
Email: {vnastos, k.arjmand, ddimop, tzallas, giannakeas, cgogos}@uoi.gr, tsakai.kevi@gmail.com
†Department of Physical Medicine and Rehabilitation, University of Ioannina, Ioannina, Greece 26500
Email: dimvarvar@gmail.com, aploumis@uoi.gr

2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM) | 979-8-3503-9858-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SEEDA-CECNSM57760.2022.9932971

***Abstract*—Human activity recognition (HAR) and gait analysis are two study topics that are used to identify numerous daily activities, such as walking, running, and stair climbing, and how they are performed. The valid identification of any gait deviation, as an abnormality in the gait cycle, can help in the real-time monitoring of patients with neuromuscular and musculoskeletal causes, and eventually in the restoration of their normal gait function. The current study combines multiple data preprocessing approaches with supervised machine learning algorithms to provide a framework for recognizing diverse gait activities using data samples from the publicly accessible “HuGaDB” human gait database. The automated analysis method takes into account 3-dimensional (3D) signals derived from two types of inertial sensors: accelerometers and gyroscopes, as well as electromyography (EMG) devices placed on the right and left leg of 18 healthy human participants. The proposed tool achieves a classification accuracy of 80% and F1-score of 79% with Random Forest emerging as the optimal gait patterns identification method.**

Keywords—Human Activity Recognition, Gait Pattern, Gait Analysis, Accelerometer, Gyroscope, Electromyography, Data Mining, Machine Learning

I. Introduction

Interpreting human activity is concerned with correctly recognizing common human actions in real-life conditions. To identify each activity, data is acquired through portable sensors, such as multimodal sensing devices, which are placed in various parts of the body, or stationary monitoring devices, including 3D motion tracking camera systems. Such an identification problem is a challenge, as it can be integrated into telemedicine and mobility rehabilitation systems for patients with neuromuscular and musculoskeletal conditions leading to motor impairment [1]. Exporting motion data from wearable sensors, on the other hand, brings various obstacles, including inter-class similarity, intra-class variability, and imbalanced class data [2]. Because of the aforementioned factors, as well as the fact that every kinetic function is unique, a significant amount of research has been conducted using a variety of approaches based on data mining and machine learning techniques for extracting high-level information from raw data and detecting any deviation from normal gait function [3].

Methods targeting diverse characteristics of healthy and patient motor status derived from wearable sensors are available in the literature, which focus on activities involving the lower limbs of the body (hip, knee, leg, foot), such as walking, running, balance, and so on. Badawi et al. [4] examined whether integrating accelerometer and gyroscope signals could improve the recognition of human activities in the “HuGaDB” dataset using various machine learning techniques and whether some sensor positions were better than others at identifying each activity. Kececi et al. [5] utilised also the open source “HuGaDB” database’s gait patterns for user authentication purposes by employing multiple machine learning classifiers. With regard to non-publicly available and custom datasets, Ardestani et al. [6] aimed to train a feed-forward artificial neural network, utilizing gait data gathered from pre-rehabilitation ground reaction forces and electromyograms, to predict the subsequent medial knee contact force based on rehabilitation patterns. Xia et al. [7] developed a convolutional neural network architecture to distinguish freezing of gait (FOG) from normal walking patterns from one-dimensional acceleration input signals, with the purpose of monitoring and aiding FOG patients during their rehabilitation treatment. The Toledo-Pérez et al. methodology [8] compared the accuracy of the intention of movement classification based on the increasing number (1 to 4) of surface electromyography (sEMG) signal channels in the right lower limb of healthy subjects. Di Nardo et al. [9] followed a similar approach to assess the impact of a varying number of sEMG sensors (4 to 1) on the binary classification of gait phases and prediction of heel-strike from swing to stance and toe-off from stance to swing time. Morbidoni et al. [10] employed an artificial neural network to predict the foot-floor-contact signal and classify gait events based on sEMG activity data (deceleration, reversing, acceleration, etc.). The Nacpil et al. study [11] evaluated the ability of transfer functions to predict the nonlinear behavior of sEMG signals and muscle acceleration during walking, with the ultimate goal of detecting gait pathologies and assisting in the design of lower prosthetic limbs that mimic the movement of the healthy counterpart. Wei et al. [12] studied the effectiveness of various feature extraction and preprocessing strategies from sEMG and electroencephalogram (EEG) channels, as well as the performance of different machine learning algorithms in the classification of gait phases. Zeng et al. [13] explored the effectiveness of several classification models to identify any anterior cruciate ligament (ACL) injury and to distinguish between ACL-deficient and ACL-intact knees based on features obtained from knee, hip and ankle gait kinematic and kinetic data. Christou et al. [14] created a method for differentiating

We acknowledge support of this work by the project “MEGATRON” (MIS 5047227) which is implemented under the Action “Reinforcement of the Research and Innovation Infrastructure”, funded by the Operational Programme ”Competitiveness, Entrepreneurship and Innovation” (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

979-8-3503-9858-8/22/$31.00 ©2022 IEEE

healthy from patient subjects with hemiplegia, but also the type of hemiplegia, utilizing three-dimensional data obtained from magnetometer and gyroscope devices, as well as accelerometer sensors from the ”RehaGait” mobile analysis system. In the Li et al. paper [15] a prediction model was presented for estimating human motor intention in the lower limbs from sEMG signals and motion data of the hip and knee during walking, using a hybrid model consisting of a fuzzy wavelet neural network (FWNN) and a zeroing neural network (ZNN) for eliminating the FWNN’s prediction errors.

In this study, a methodological framework is presented for identifying human activities in a publicly available dataset. Data from two types of inertial sensors are used: 1) accelerometers and 2) gyroscopes, as well as data from electromyography (EMG) sensors to identify patterns in the movement of each healthy subject in the database. More technically, the sensor data is used to predict each kinetic activity by applying two dimensionality reduction and five classification algorithms. Subsequently, combinations of these feature extraction and classification techniques are applied to achieve the most optimal kinetic state identification performance. The performance of the applied combinations is evaluated and compared using the 10-fold cross-validation method, with classification accuracy and F1-score as the main criteria.

II. Methodology

The following paragraphs give an overview of the dataset used in the proposed methodological approach, followed by a detailed examination of the data preprocessing and machine learning techniques used to build the automated human motion activities recognition tool. In the future, the proposed method could be integrated into a complete prognostic tool for differentiating healthy from abnormal gait functions and monitoring the disease-affected activity in clinical trials and rehabilitation centers.

A. Dataset Description

The dataset used in this work is entitled “HuGaDB” [16] and is obtained from the Kaggle web-based data science community[1]. It contains data from the motor activity of 18 healthy individuals which were collected using three pairs of inertial sensors, corresponding to a) 3-axis accelerometers and b) 3-axis gyroscopes, as well as one pair of EMG sensors put on the right and left legs. More specifically, the inertial sensor signals were retrieved from a pair of sensors placed on the rectus femoris muscle 5 cm above the knee, a pair of sensors in the center of the shinbone where the calf ends, and a pair on the metatarsal bones of the feet. The EMG signals were instead obtained from sensors in the vastus lateralis.

In total, 38 signals were acquired, 36 from inertial sensors and 2 from EMG sensors, which were assembled into 16 dataframes per participant (*n*=18) in the form of feature samples [16]. In these produced dataframes, each exported sample was annotated with a label $y \in \{$walking, running, going up, going down, sitting, sitting down, standing up, standing, bicycling, up by elevator, down by elevator, sitting in car$\}$ (Table I) that denotes the activity performed at that moment.

Table I: Recorded Gait Activities in HuGaDB

ID	Activity	Time (min)	Percent	Samples
1	Walking	192	32.15	679073
2	Running	20	3.39	71653
3	Going up	37	6.23	131604
4	Going down	33	5.52	116637
5	Sitting	68	11.45	241849
6	Sitting down	6	1.14	24112
7	Standing up	6	1.06	22373
8	Standing	93	15.56	328655
9	Bicycling	44	7.41	156560
10	Up by elevator	25	4.22	89144
11	Down by elevator	19	3.30	69729
12	Sitting in car	51	8.55	180573
	Total	598	100	2111962

B. Machine Learning Workflow

The identification of 12 different gait activities using the HuGaDB data structure (Table I) results in a multi-class classification problem. For this problem to be solved, the proposed methodology employs a machine learning pipeline, as shown in Figure 1. The following texts provide a brief overview of each step performed to identify each human activity, while the classification results are presented in Section III.

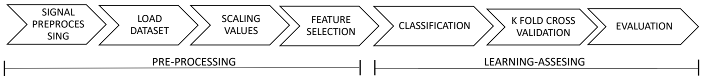

Figure 1: Flowchart of the proposed human activity recognition method

1) Missing values: All human activity data are first loaded into the classification system based on the CSV extension files shared by the HuGaDB authors, and then a search for missing values in each of the participants’ dataframes is conducted. During the search, no missing values were discovered.

2) Scaling: To address the issue of varied scaling in the employed data, their standardization with Z-score normalization is preferred, since the dataframes comprise feature values from three-dimensional sensors and electromyography signals expressed in different units of measurement: a) accelerometer: m/s^2, b) gyroscope: $^o/s$ and c) EMG: *Volts*. Here, the values are centered around the mean, which is assumed to have a value of 0, whereas the resulting distribution has a unit of standard deviation. As an outcome, each feature in isolation is scaled to the closed range [-1,1]. As an alternative, ”Min-Max” normalization [0,1] is applied. This is a crucial step before training and testing machine learning algorithms since feature values with a larger range tend to outweigh those with a smaller range, and may cause a decrease in activity recognition performance. Additionally, by scaling the feature set values, we assure that the information will remain constant by limiting the number of outliers to the minimum possible.

3) Dimensionality reduction: To minimize the dimensionality of the HuGaDB data and maintain the features most relevant to each motor activity, the principal components analysis (PCA) is used. PCA is a linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and their projection. Using the PCA technique, the data are projected onto a space of $M < D$ dimensions, which maximizes the variance of the projected data [17].

[1] HuGaDB (Human Gait Database)

4) Feature ranking: Another method utilized in feature reduction approaches and based on the probabilistic dependence of each independent feature *x* on the dependent target variable *y*. The observed count is close to the expected count when two features are independent, hence the Chi-square value will be lower. Therefore, a high Chi-square score suggests that the independence hypothesis is false. Simply put, features that are more reliant on the response and have larger Chi-square values might be chosen for model training. To determine the number of most informative features, a wrapping feature selection method[2] is applied (Estimator: Random Forest with balanced distribution [18]). Afterward, followed by a Grid-search procedure that tunes the parameter 'k', which corresponds to the number of features, the value of *k*=36 is obtained.

5) Classification: Given the No Free Lunch theorem [19], it is reasonable to assume that no individual supervised method can provide the optimum generalization capability for each classification problem's input data. In response to this problem, a pool of well-known supervised models is formed within the scope of the current methodology, with the previously preprocessed gait data as input. Among the models are Decision Trees, k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), as well as the Random Forests and AdaBoost ensemble algorithms. The following is a quick summary of the preferred classifiers:

- Decision Tree: It is a hierarchical supervised learning model, in which an input sample is identified through a recurring branching process [20], [21]. Finding the optimal number of layers in the tree to maximize the prediction accuracy while minimizing the risk of overfitting is often a challenge. A key advantage of this algorithm is its interpretation, as it relies on a logical set of decisions based on "if-then" conditions.
- Random Forest: It consists of a large number of individual decision trees [22] that are typically trained using the bagging method [23]. Specifically, decision trees with low correlations among each other operate as an ensemble guided by the "wisdom of crowds" concept, contributing to the formation of a better learning agent with lower error rates.
- k-NN: A lazy-learner algorithm which assigns an unknown sample to the class that the majority of its k-nearest neighbors belong. The term of "lazy-learner" refers to the fact that it doesn't implement a function that subsequently performs the discrimination, instead the feature samples are called into the computer system memory during runtime, defining k-NN as a memory-based classifier.
- SVM: At its core is a two-class classifier. However, when applied to a multi-class ($K > 2$) problem, then a combination of multiple two-class SVMs is created. The method uses the concept of margin for the classification of input samples, which is defined as the shortest distance between the decision boundary and the data points both the positive and negative sides of a hyperplane [17], [24].
- AdaBoost: It is a meta-algorithm that refers to an ensemble of base classifiers called "weak learners", that are trained sequentially [17], [25]. Each base classifier is trained using a weighted form of the dataset, with the weighting coefficient associated with each data sample being determined by the performance of the previous classifiers. In particular, data samples that are incorrectly labeled by one of the base classifiers are given greater weight when training the next classifier in the sequence. After all of the classifiers are trained, their predictions are combined through a weighted majority voting system.

III. Evaluation

The workflow described above is carried out in two phases. The first phase is data preprocessing and dimensionality reduction, where the number of features is reduced to 36 (from the original 38). Table II displays the p-values for the most significant feature values that are retrieved using the Chi-square method, whereas PCA reshaped the feature set to 36 principal components. The table suggests that gyroscopes and EMG devices are the most essential wearable sensors and features that are extracted from them have the highest probability of being selected for model training.

Table II: Chi-square and PCA Feature Importance Values

Feature	**Chi-square**	**PCA**
EMG_right	0.9983	1.0000
EMG_left	0.9442	1.0000
gyroscope_right_thigh_x	0.8751	0.9977
gyroscope_left_shin_x	0.3153	0.9953
gyroscope_left_thigh_x	0.1031	0.9895
gyroscope_left_shin_z	0.0949	0.9834
gyroscope_right_foot_z	0.0061	0.9770
gyroscope_left_foot_z	0.0049	0.9702
gyroscope_left_thigh_z	0.0012	0.9624
gyroscope_right_shin_x	0.0003	0.9543
gyroscope_right_thigh_z	0.0001	0.9460
gyroscope_right_shin_z	0.0000	0.9369
gyroscope_left_foot_y	0.0000	0.9272
gyroscope_right_foot_y	0.0000	0.9173
accelerometer_left_shin_x	0.0000	0.9067
gyroscope_left_shin_y	0.0000	0.8957
gyroscope_right_foot_x	0.0000	0.8841
accelerometer_left_thigh_y	0.0000	0.8719
accelerometer_left_shin_y	0.0000	0.8589
gyroscope_right_shin_y	0.0000	0.8454
gyroscope_left_foot_x	0.0000	0.8299
accelerometer_right_shin_y	0.0000	0.8137
gyroscope_right_thigh_y	0.0000	0.7969
gyroscope_left_thigh_y	0.0000	0.7789
accelerometer_right_thigh_y	0.0000	0.7606
accelerometer_right_foot_z	0.0000	0.7407
accelerometer_left_foot_y	0.0000	0.7203
accelerometer_left_foot_z	0.0000	0.6964
accelerometer_right_foot_y	0.0000	0.6716
accelerometer_right_shin_x	0.0000	0.6439
accelerometer_left_thigh_z	0.0000	0.6149
accelerometer_left_thigh_x	0.0000	0.5497
accelerometer_left_shin_z	0.0000	0.5093
accelerometer_right_foot_x	0.0000	0.4654
accelerometer_right_thigh_z	0.0000	0.4156
accelerometer_right_thigh_x	0.0000	0.3493
accelerometer_right_shin_z	0.0000	0.2595
accelerometer_left_foot_x	0.0000	0.1511

The second phase involves measuring the motion activity prediction performance of the classification models. As mentioned in Section II-B, five classification methods are applied. The confusion matrix, as well as accuracy, recall, precision,

[2] FFS (**F**orward **F**eature **S**election)

and F1-score, are common metrics used to evaluate the effectiveness of supervised algorithms. In classification problems, an algorithm might excel in one metric but might underperform in another. The user should thus be well-informed about the trade-offs involved. The confusion matrix is a numerical tableau depiction of how each model fits the data. It consists of n rows and n columns, where *n* is the number of target classes (in our case, *n*=10). Within the confusion matrix, let TP (True Positive) be the number of positive samples that are also classified as positive and TN (True Negative) be the number of negative samples that are also classified as negative. Let FP (False Positive) be the number of negative samples mistakenly classified as positive, and FN (False Negative) be the number of positive samples mistakenly classified as negative. Through these measures, the classification accuracy is derived as the percentage of correctly classified samples, given in Equation 1, precision as the false negative rate (FNR), given in Equation 2, and recall as the true positive rate (TPR), given in Equation 3. Furthermore, F1-score is calculated providing a value that shows how favorite is the balance between precision and recall, and is given by Equation 4.

As previously indicated, the target classes have been reduced from 12 to 10. This is because the activities ”Down by elevator” and ”Up by elevator” have been merged since the samples from the two categories have a high correlation rate. Additionally, the activity ”Sitting down” is merged with the category ”Sitting” due to the former’s lack of information, which leads to high confusion. This is inferred according to information obtained from the confusion matrix, indicating a reduced variability between them. In the Experiments section, a comparison is made between each supervised model’s predictions and the ground truth labels.

IV. Experiments

Our experiments are conducted using the Python programming language, the Scikit-learn [26] package, and a Windows 10 workstation equipped with an AMD Ryzen 5700G(8C/16T) processor and 32GB of RAM. A significant number of runs are executed since we try all combinations of data preprocessing steps and classification algorithms that are described in Section II-B. In particular, for 10 folds, we combine the data scaling approach with two feature reduction and five classifier alternatives, totaling 400 individual runs that kept our workstation busy for about three hours. The metrics that we have obtained for visualization are the accuracy and F1-score, for each classifier, which is presented in the boxplots of Figure 2.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{2}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{3}$$

$$F1\_score_i = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{4}$$

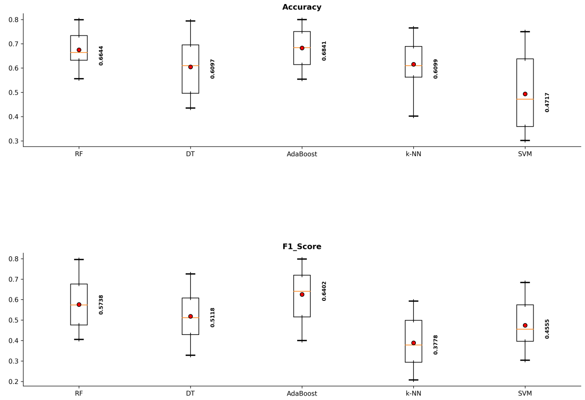

Figure 2: Comparison of classifiers in metric terms

To identify the best-performing classifiers, the Pareto-optimal objects [27] of all runs are computed based on two performance metrics (accuracy and F1-score). For this step, the Pareto front (Pareto frontier) was computed using the Python OApackage[3] library, with the retrieved non-dominant solutions presented in Table III. It is observed that Random Forest emerges as the best classifier, occupying two of the four entries in the table, with AdaBoost occupying the other two.

Table III: Pareto Optimal Solutions for Scoring Metrics

Scaling	Feature Selection	ML Algorthm	Accuracy	F1-score
Z-score	Chi-square	Random Forest	0.799341	0.668602
Min-Max	PCA	Random Forest	0.780483	0.676891
Z-score	Chi-square	AdaBoost	0.624712	0.789582
Z-score	PCA	AdaBoost	0.749181	0.785339

V. Conclusion

In this paper, we examine a particular case of the problem of motion prediction using wearable sensors. HugaDB is used as the dataset and a machine learning workflow is set up utilizing various data preprocessing steps and five classification approaches. Using 10-fold cross-validation, Random Forest emerges as the most effective algorithm for our problem in terms of attaining high accuracy (80%) results and occupies two of four non-dominant Pareto objects. In future work, we plan to expand the workflow to more datasets using different devices, such as force plates and 3D cameras, to acquire new gait data and apply other state-of-the-art methods to the motion prediction problem, such as shallow and deep neural networks, but also voting classification systems.

[3] OApackage (**O**rthogonal **A**rray package)

REFERENCES

[1] C. Jobanputraa, J. Bavishib and N. Doshic, “Human activity recognition: A survey”, Procedia Computer Science, vol. 155, pp. 698–703, January 2019.

[2] S. R. Ramamurthy and N. Roy, “Recent trends in machine learning for human activity recognition - A survey”, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, March 2018.

[3] Md. M. Islam, S. Nooruddin, F. Karray and G. Muhammad, “Human activity recognition using tools of convolutional neural networks: A state of the art review, data sets, challenges and future prospects”, arXiv:2202.03274, February 2022.

[4] A. A. Badawi, A. Al-Kabbany and H. Shaban, “Multimodal human activity recognition from wearable inertial sensors using machine learning”, In proceedings of the 2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), Sarawak, Malaysia, December 3–6, 2018.

[5] A. Kececi, A. Yildirak, K. Ozyazici, G. Ayluctarhan, O. Agbulut and I. Zincir, “Implementation of machine learning algorithms for gait recognition”, Engineering Science and Technology an International Journal, vol. 23, pp. 931–937, February 2020.

[6] M. M.Ardestani, Z. Chen, L. Wang, Q. Lian, Y. Liu, J. He, D. Li and Z. Jin, “Feed forward artificial neural network to predict contact force at medial knee joint: Application to gait modification”, Neurocomputing, vol. 139, pp. 114–129, September 2014.

[7] Y. Xia, J. Zhang, Q. Ye, N. Cheng, Y. Lu and D. Zhang, “Evaluation of deep convolutional neural networks for detection of freezing of gait in Parkinson’s disease patients”, Biomedical Signal Processing and Control, vol. 46, pp. 221–230, September 2018.

[8] D. C. Toledo-Pérez, M. A. Martínez-Prado, R. A. Gómez-Loenzo, W. J. Paredes-García and J. Rodríguez-Reséndiz, “A study of movement classification of the lower limb based on up to 4-EMG channels”, Electronics, vol. 8, February 2019.

[9] F. Di Nardo, C. Morbidoni, A. Cucchiarelli and S. Fioretti, “Influence of EMG-signal processing and experimental set-up on prediction of gait events by neural network”, Biomedical Signal Processing and Control, vol. 63, January 2021.

[10] C. Morbidoni, A. Cucchiarelli, S. Fioretti and F. Di Nardo, “A deep learning approach to EMG-based classification of gait phases during level ground walking”, Electronics, vol. 8, August 2019.

[11] E. J. C. Nacpil, S. Nacy and G. Youssef, “Feasibility assessment of transfer functions describing biomechanics of the human lower limb during the gait cycle”, Biomedical Signal Processing and Control, vol. 69, August 2021.

[12] P. Wei, J. Zhang, F. Tian and J. Hong, “A comparison of neural networks algorithms for EEG and sEMG features based gait phases recognition”, Biomedical Signal Processing and Control, vol. 68, April 2021.

[13] W. Zeng, S. A. Ismail and E. Pappas, “The impact of feature extraction and selection for the classification of gait patterns between ACL deficient and intact knees based on different classification models”, EURASIP Journal on Advances in Signal Processing, October 2021.

[14] V. Christou, A. Arjmand, D. Dimopoulos, D. Varvarousis, I. Tsoulos, A. T. Tzallas, et al., “Automatic hemiplegia type detection (right or left) using the Levenberg-Marquardt backpropagation method”, Information, vol. 13, February 2022.

[15] W. Li, K. Liu, Z. Sun, C. Li, Y. Chai and J. Gu, “A neural network-based model for lower limb continuous estimation against the disturbance of uncertainty”, Biomedical Signal Processing and Control, vol. 71, January 2022.

[16] R. Chereshnev and A. Kertesz-Farkas, “HuGaDB: Human gait database for activity recognition from wearable inertial sensor networks”, arXiv:1705.08506, July 2017.

[17] C. M. Bishop, “Pattern recognition and machine learning”, Springer-Verlag New York, February 2006.

[18] Z. Agusta and K. Adiwijaya, “Modified balanced random forest for improving imbalanced data prediction”, International Journal of Advances in Intelligent Informatics, vol. 5, pp. 58–65, 2019.

[19] D. Gómez and A. Rojas. “An empirical overview of the no free lunch theorem and its effect on real-world machine learning classification”, Neural Computation, vol. 28, pp. 216-–228, January 2016.

[20] H. Patel and P. Prajapati. “Study and analysis of decision tree based classification algorithms”, International Journal of Computer Sciences and Engineering, vol. 6, pp.74-–78, October 2018.

[21] T. Dietterich, C. Bishop, D. Heckerman, M. Jordan and M. Kearns, “Introduction to machine learning”, 2nd ed. Ethem Alpaydin, The MIT Press, 2010.

[22] N. M. Abdulkareem and A. M. Abdulazeez, “Machine learning classification based on random forest algorithm: A review”, International Journal of Science and Business, vol. 5, pp. 128-–142, January 2021.

[23] C. D Sutton, “Classification and regression trees, bagging, and boosting”, Handbook of Statistics, vol. 24, pp. 303-–329, December 2005.

[24] M. Mohri, A. Rostamizadeh and A. Talwalkar, “Foundations of machine learning”, The MIT Press, August 2012.

[25] R. E. Schapire, “Explaining AdaBoost”, Empirical Inference, Springer, pp. 37-–52, October 2013.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., “Scikit-learn: Machine Learning in Python”, Journal of Machine Learning Research, vol. 12, pp. 2825–2830, January 2011.

[27] H. Y. Alhammadi and J. A. Romagnoli, ”The Integration of process design and control”, Computer Aided Chemical Engineering, vol. 17, pp. 264–305, March 2004.