The Effectiveness of PCA in Decision Tree and Random Forest for Raisin Dataset

Agnieszka Polowczyk[1], Alicja Polowczyk[1]

[1]Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland

Abstract

In machine learning we can emphasize models based on the such as supervised learning and unsupervised learning. Mainly algorithms based on learning without teacher are used to clustering process. This algorithms are used to split ours data to smaller groups, clusters with similar and comparable attributes. Guided learning is utilized to create many classifiers. On the basis of previously prepared training data, the classifier learns certain relations and dependencies so that it can correctly predict target values later. In our paper we will look at two rule-based models that use decision rules to classify data samples. Examples of models are Decision Tree and Random Forest which are created for different hyperparameters. We will also show how the reduction of dimensionality affects to effectiveness and efficiency our models by using PCA technique and correlation analysis to select the most relevant features.

Keywords

Machine learning, pca, decision tree, random forest, raisin, classifiers

1. Introduction

Artificial intelligence is used in many areas. In image processing for example is applied in feature correction and encryption [1]. In the financial and economic sector AI is used to predict housing prices or even prices of products on the food market. We can also find applications in recommendation systems, ie. [2] proposed crop recommender for agriculture by the use of XAI-driven model. There are many types of models in machine learning, they are for example: Linear Regression, Gaussian Naive Bayes Classifier, Decision Tree, Random Forest, Support Vector Machine or model based on neural networks [3, 4, 5]. Every classifier has another method to determine the predicted values which means that not every model will have high effectiveness for each dataset. Better accuracy for image classification is achieved by models such as CNN [6], but for simple and low-dimensional data, where distance between points is important in classification using KNN is a good idea. We should always choose a model after the initial analysis of the data. Every model is equipped with many hyperparameters that we can adjust and self-change.

In the case of KNN model we can establish number of nearest neighbors. However, often the problem is to determine the optimal value k-nearest neighbors, in [7] described the K-Tree method that solves this problem. In a Random Forest, we specify the number of Decision Trees during training. Additionally, an important aspect before training the algorithm is preparing data. This preparing is based on standardization or normalization our dataset. In the case of high dimensionality of the data, various dimensionality reduction techniques are often used [8, 9, 10] to reduce computational complexity and speed up the model training process. We can also find various applications to data classification and recommendation systems by using models of machine learning. In [11] was proposed model of neural network for imbalanced data collection on the input of classifier. Very often computation models are used for positioning, ie. power electric systems [12, 13], or for human behavior understanding [14, 15]

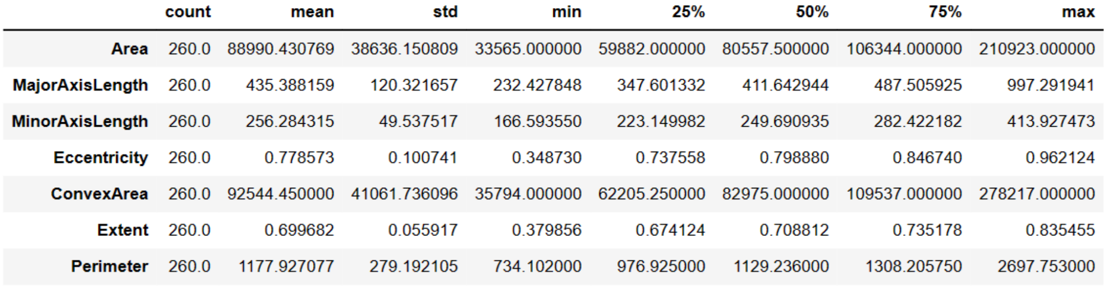
In this paper, we will compare two rule-based models: Decision Tree and Random Forest, which were built for three different dataset:

- model uses PCA to reduce the dimensionality of the data
- model uses two features selected after data analysis
- model uses all the features

We will also check the effectiveness of above, our models. In the case of Decision Tree for different measure: entropy and gini, and for various depths. For the Random Forest, we will test the performance of the algorithm for a different number of decision trees. At the end, we will make a summary, whether the reduction of dimensions contributed to the high accuracy of our models.

ICYRIME 2023: 8th International Conference of Yearly Reports on Informatics, Mathematics, and Engineering. Naples, July 28-31, 2023

ap307985@polsl.pl (A. Polowczyk); ap307986@polsl.pl (A. Polowczyk)



	count	mean	std	min	25%	50%	75%	max
Area	260.0	88990.430769	38636.150809	33565.000000	59882.000000	80557.500000	106344.000000	210923.000000
MajorAxisLength	260.0	435.388159	120.321657	232.427848	347.601332	411.642944	487.505925	997.291941
MinorAxisLength	260.0	256.284315	49.537517	166.593550	223.149982	249.690935	282.422182	413.927473
Eccentricity	260.0	0.778573	0.100741	0.348730	0.737558	0.798880	0.846740	0.962124
ConvexArea	260.0	92544.450000	41061.736096	35794.000000	62205.250000	82975.000000	109537.000000	278217.000000
Extent	260.0	0.699682	0.055917	0.379856	0.674124	0.708812	0.735178	0.835455
Perimeter	260.0	1177.927077	279.192105	734.102000	976.925000	1129.236000	1308.205750	2697.753000

Figure 1: Raisin dataset information

2. Raisin database

The database that we used to build various classifiers contains samples that were described by 7 morphological features. These features were obtained after previously processing the photos.Values are continuous and we can see that each feature has value from different ranges. There are also high values of standard deviations for example, for Area and ConvexArea features, indicating that the values for these features are highly dispersed from their mean. The Fig. 1 shows a table containing the statistics of our attributes.

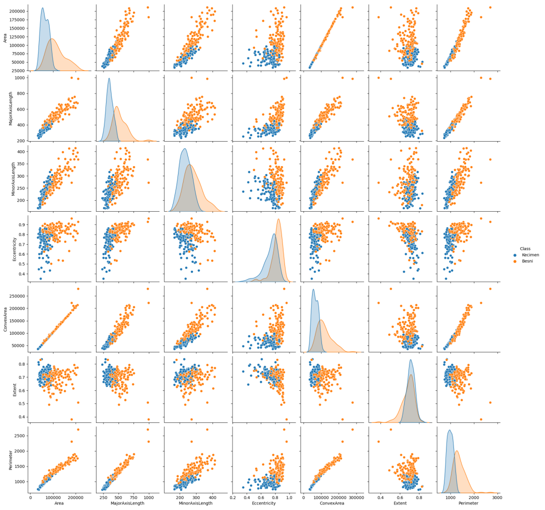


Figure 2: Raisin dataset graphs

2.1. Standardization

To improve the effectiveness and efficiency of the model, data normalization or standardization is used. However rule-based models don't require transformations to a single scale, because this classifiers make predictions based on specific rules. Nevertheless, in our case, we have standardized for:

- models that were built from lower dimensionality data using the PCA technique. When using this technique, it is recommended to before standardize the data.
- models that were based on two features that we chose. Standardization data contributed to changes in values to a similar range which helped in the creation of decision boundary charts.

In our classifiers, we used standardization that transforms the data in such a way that its mean is equal to 0 and the standard deviation is equal to 1. First for every attribute we calculated its mean and standard deviation. Later, we used the obtained results to compute the new values using the below formula:

$$x_{new} = \frac{x - \mu}{\sigma} \tag{1}$$

2.2. Model based on PCA

One of the popular dimensionality reduction techniques is PCA. The task of PCA is to return n-features that we can create a model with high accuracy. PCA model can be improved for sophisticated data on the input, [16] presented denoising of the input for improved processing. In our models were used PCA, which returns to us new training and test data reduced from seven to two dimensions.

2.3. Model based on two features

Another way to prepare data for the model is to reduce dimensionality based on correlation analysis. Correlation defines the relation between two variables. Correlation value close to 1 or -1 mean a strong correlation, but value close to 0 mean weak correlation. The Extent feature was removed from our training and testing data, because its correlation value with our target feature was only

0.28. Additionally, the following features were eliminated: ConvexArea, Perimeter, Area, MinorAxisLength, because these attributes had strong relation with other features and didn't contribute relevant information to the classification models. Finally, our classifiers were built on other two features: MajorAxisLength and Eccentricity. The Fig. 3 shows correlation plots between two features.

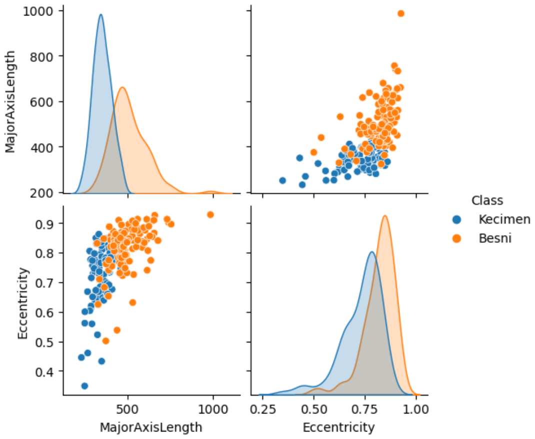


Figure 3: Correlation graphs of two features

2.4. Model based on all features

For each classifier, we also built a model based on all seven features. Sometimes training a model on the basis of all attributes can be a disadvantage, because this approach lead to slower learning of our classifier. However, the advantage of including all features is that in some cases it can lead to very high efficiency of our machine learning algorithm, because we don't lose any relevant information. Fig. 2 illustrates our feature and correlation graphs.

3. Methods

3.1. Decision Tree

3.1.1. Formulas

Entropy:

$$-\sum_{i=1}^{n} p_i \cdot \log_2(p_i) \tag{2}$$

Entropy $_{\text{after}}$:

$$-\sum_{i=1}^{n} \frac{S_i}{S} Entropy(S_i) \tag{3}$$

Information gain:

$$Entropy_{\text{before}} - Entropy_{\text{after}} \tag{4}$$

Gini coefficient:

$$-\sum_{i=1}^{n} p_i \cdot p_i^2 \tag{5}$$

Gini coefficient $_{\text{after}}$:

$$-\sum_{i=1}^{n} \frac{S_i}{S} Gini(S_i) \tag{6}$$

Information gain:

$$Gini_{\text{before}} - Gini_{\text{after}} \tag{7}$$

3.1.2. Algorithm

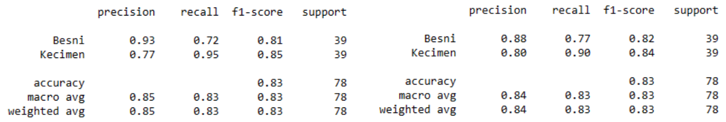
A Decision Tree is a directed model that consists of a root, nodes, leaves and edges. Root is top of the tree, passing through the edges, we come to the nodes and finally to the leaves, to the lowest layer of the tree. Leaves contain the answers, predictions of our model, to which class our data sample is classified. Nodes contain rules that are used to make decisions during testing. Rules are created using impurity measures. These are: entropy and gini coefficient. Our classifiers will create rules that will divide our sets into more pure subsets. The final conditions are those for which the information gain is the greatest. The Decision Tree has a tendency to overfitting, so we used the following as regularization parameters: number of max depth is 2 and 3, and the minimum amount of data in the set before the division can not be less than 2.

3.2. Random Forest

Random Forest algorithm creates a forest in a random manner. This "forest" you can think of as an ensemble of Decision Trees, most of the time trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. The Random Forest starts by selecting random samples from the given dataset. It selects these random subsets with replacement, meaning that some samples may be used multiple times in a single subset. e features at each split in the tree. This randomness in feature selection is what gives the Random Forest its name. The Random Forest consist of many decision trees. Test data is classified by decision trees. Next, voting takes place and we look at which class/forecast occurs most frequently. Random Forest is better option than Decision Tree, because this classifier has not a tendency to overfitting to training data. Our models include several dozen decision trees, where each of them has been trained for different training data that has been previously randomized from the main dataset intended for training.

4. Experiments

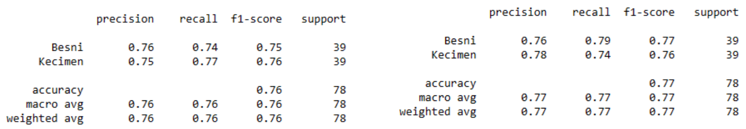
4.1. Decision Tree



```
              precision    recall  f1-score   support

       Besni       0.93      0.72      0.81        39
     Kecimen       0.77      0.95      0.85        39

    accuracy                           0.83        78
   macro avg       0.85      0.83      0.83        78
weighted avg       0.85      0.83      0.83        78
```

```
              precision    recall  f1-score   support

       Besni       0.88      0.77      0.82        39
     Kecimen       0.80      0.90      0.84        39

    accuracy                           0.83        78
   macro avg       0.84      0.83      0.83        78
weighted avg       0.84      0.83      0.83        78
```

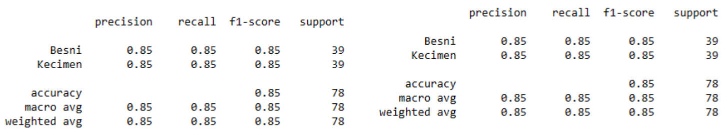
Figure 4: Classification reports for Decision Tree with PCA for depth equal to 2. The results are shown in order for the measures: entropy and gini



```
              precision    recall  f1-score   support

       Besni       0.76      0.74      0.75        39
     Kecimen       0.75      0.77      0.76        39

    accuracy                           0.76        78
   macro avg       0.76      0.76      0.76        78
weighted avg       0.76      0.76      0.76        78
```

```
              precision    recall  f1-score   support

       Besni       0.76      0.79      0.77        39
     Kecimen       0.78      0.74      0.76        39

    accuracy                           0.77        78
   macro avg       0.77      0.77      0.77        78
weighted avg       0.77      0.77      0.77        78
```

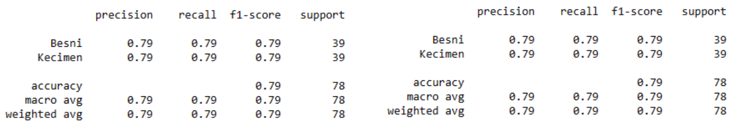
Figure 5: Classification reports for Decision Tree with PCA for depth equal to 3. The results are shown in order for the measures: entropy and gini



```
              precision    recall  f1-score   support

       Besni       0.85      0.85      0.85        39
     Kecimen       0.85      0.85      0.85        39

    accuracy                           0.85        78
   macro avg       0.85      0.85      0.85        78
weighted avg       0.85      0.85      0.85        78
```

```
              precision    recall  f1-score   support

       Besni       0.85      0.85      0.85        39
     Kecimen       0.85      0.85      0.85        39

    accuracy                           0.85        78
   macro avg       0.85      0.85      0.85        78
weighted avg       0.85      0.85      0.85        78
```

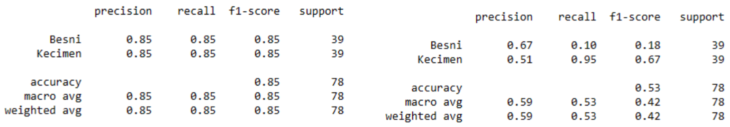
Figure 6: Classification reports for Decision Tree with two features for depth equal to 2. The results are shown in order for the measures: entropy and gini



```
              precision    recall  f1-score   support

       Besni       0.79      0.79      0.79        39
     Kecimen       0.79      0.79      0.79        39

    accuracy                           0.79        78
   macro avg       0.79      0.79      0.79        78
weighted avg       0.79      0.79      0.79        78
```

```
              precision    recall  f1-score   support

       Besni       0.79      0.79      0.79        39
     Kecimen       0.79      0.79      0.79        39

    accuracy                           0.79        78
   macro avg       0.79      0.79      0.79        78
weighted avg       0.79      0.79      0.79        78
```

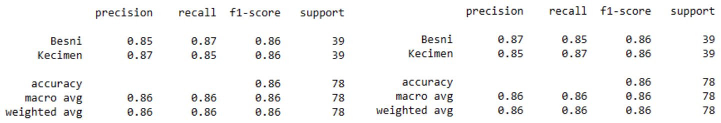
Figure 7: Classification reports for Decision Tree with two features for depth equal to 3. The results are shown in order for the measures: entropy and gini



```
              precision    recall  f1-score   support

       Besni       0.85      0.85      0.85        39
     Kecimen       0.85      0.85      0.85        39

    accuracy                           0.85        78
   macro avg       0.85      0.85      0.85        78
weighted avg       0.85      0.85      0.85        78
```

```
              precision    recall  f1-score   support

       Besni       0.67      0.10      0.18        39
     Kecimen       0.51      0.95      0.67        39

    accuracy                           0.53        78
   macro avg       0.59      0.53      0.42        78
weighted avg       0.59      0.53      0.42        78
```

Figure 8: Classification reports for Decision Tree with all features for depth equal to 2. The results are shown in order for the measures: entropy and gini



```
              precision    recall  f1-score   support

       Besni       0.85      0.87      0.86        39
     Kecimen       0.87      0.85      0.86        39

    accuracy                           0.86        78
   macro avg       0.86      0.86      0.86        78
weighted avg       0.86      0.86      0.86        78
```

```
              precision    recall  f1-score   support

       Besni       0.87      0.85      0.86        39
     Kecimen       0.85      0.87      0.86        39

    accuracy                           0.86        78
   macro avg       0.86      0.86      0.86        78
weighted avg       0.86      0.86      0.86        78
```

Figure 9: Classification reports for Decision Tree with all features for depth equal to 3. The results are shown in order for the measures: entropy and gini

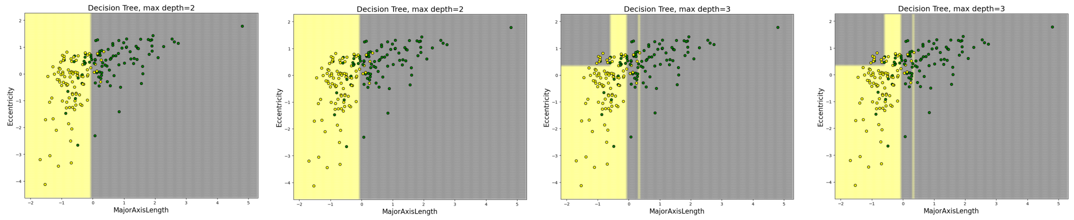
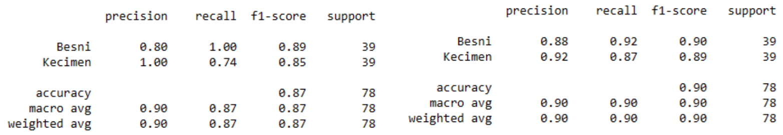


Figure 10: Decision boundaries for a Decision Tree with two features. The results are presented in order for depths equal to 2 and 3, where for each depth for the measure of entropy and gini

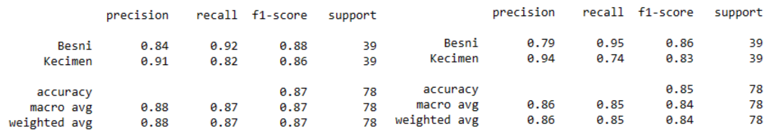
4.2. Random Forest



```
              precision    recall  f1-score   support

       Besni       0.80      1.00      0.89        39
     Kecimen       1.00      0.74      0.85        39

    accuracy                           0.87        78
   macro avg       0.90      0.87      0.87        78
weighted avg       0.90      0.87      0.87        78
```

```
              precision    recall  f1-score   support

       Besni       0.88      0.92      0.90        39
     Kecimen       0.92      0.87      0.89        39

    accuracy                           0.90        78
   macro avg       0.90      0.90      0.90        78
weighted avg       0.90      0.90      0.90        78
```

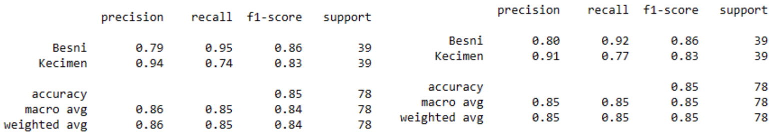
Figure 11: Classification reports for Random Forest with PCA. The results are shown in order for the number of Decision Trees: 50 and 100



```
              precision    recall  f1-score   support

       Besni       0.84      0.92      0.88        39
     Kecimen       0.91      0.82      0.86        39

    accuracy                           0.87        78
   macro avg       0.88      0.87      0.87        78
weighted avg       0.88      0.87      0.87        78
```

```
              precision    recall  f1-score   support

       Besni       0.79      0.95      0.86        39
     Kecimen       0.94      0.74      0.83        39

    accuracy                           0.85        78
   macro avg       0.86      0.85      0.84        78
weighted avg       0.86      0.85      0.84        78
```

Figure 12: Classification reports for Random Forest with two features. The results are shown in order for the number of Decision Trees: 50 and 100



```
              precision    recall  f1-score   support

       Besni       0.79      0.95      0.86        39
     Kecimen       0.94      0.74      0.83        39

    accuracy                           0.85        78
   macro avg       0.86      0.85      0.84        78
weighted avg       0.86      0.85      0.84        78
```

```
              precision    recall  f1-score   support

       Besni       0.80      0.92      0.86        39
     Kecimen       0.91      0.77      0.83        39

    accuracy                           0.85        78
   macro avg       0.85      0.85      0.85        78
weighted avg       0.85      0.85      0.85        78
```

Figure 13: Classification reports for Random Forest with all features. The results are shown in order for the number of Decision Trees: 50 and 100

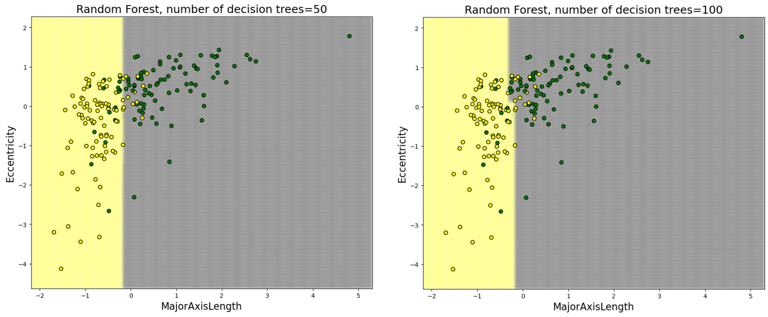


Figure 14: Decision boundaries for Random Forest with two features. The results are presented for the entropy measure

5. Conclusions

After an in-depth analysis carried out on Decision tree and Random Forest models, it can be concluded that using PCA to reduce dimensionality for our dataset is good idea. Presented models of decision trees achieve high accuracies for a depth equal of 2 at level 83 %, which were trained on a training dataset using PCA. In addition, after analyzing the correlation, we were able to find two features for which the models made predictions as good as the models for which PCA were used. Random Forest is the model which make even more effective predictions. Classifier of this type achieved an accuracy of 90 % using 100 decision trees. Additionally, an important element turned out the right choice of impurity measure, our research confirm that classifiers using the entropy measure gave better accuracy results than models that used the gini coefficient. To sum up, the use of PCA for our database allowed us to achieve equally high accuracies, while reducing computational complexity.

References

[1] W. Feng, J. Zhang, Y. Chen, Z. Qin, Y. Zhang, M. Ahmad, M. Woźniak, Exploiting robust quadratic polynomial hyperchaotic map and pixel fusion strategy for efficient image encryption, Expert Systems with Applications 246 (2024) 123190.

[2] P. Naga Srinivasu, M. F. Ijaz, M. Woźniak, Xai-driven model for crop recommender system for use in precision agriculture, Computational Intelligence 40 (2024) e12629.

[3] V. Ponzi, S. Russo, V. Bianco, C. Napoli, A. Wajda, Psychoeducative social robots for an healthier lifestyle using artificial intelligence: a case-study, volume 3118, 2021, pp. 26 – 33.

[4] G. De Magistris, R. Caprari, G. Castro, S. Russo, L. Iocchi, D. Nardi, C. Napoli, Vision-based holistic scene understanding for context-aware human-robot interaction 13196 LNAI (2022) 310 – 325. doi:10.1007/978-3-031-08421-8_21.

[5] N. Brandizzi, S. Russo, G. Galati, C. Napoli, Addressing vehicle sharing through behavioral analysis: A solution to user clustering using recency-frequency-monetary and vehicle relocation based on neighborhood splits, Information (Switzerland) 13 (2022). doi:10.3390/info13110511.

[6] K. Huang, Image classification using the method of convolutional neural networks, 2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS) (2022) 827–832.

[7] S. Zhang, X. Li, M. Zong, X. Zhu, R. Wang, Efficient knn classification with different numbers of nearest neighbors, IEEE Transactions on Neural Networks and Learning Systems (2018) 1774–1785.

[8] G. D. Magistris, C. Rametta, G. Capizzi, C. Napoli, Fpga implementation of a parallel dds for wide-band applications, volume 3092, 2021, pp. 12 – 16.

[9] H. S. Parmar, S. Mitra, B. Nutter, R. Long, S. Antani, Visualization and detection of changes in brain states using t-sne, 2020 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI) (2020) 14–17.

[10] C. Napoli, G. Pappalardo, E. Tramontana, A hybrid neuro-wavelet predictor for qos control and stability 8249 LNAI (2013) 527 – 538. doi:10.1007/978-3-319-03524-6_45.

[11] M. Woźniak, M. Wieczorek, J. Siłka, Bilstm deep neural network model for imbalanced medical data of iot systems, Future Generation Computer Systems 141 (2023) 489–499.

[12] F. Bonanno, G. Capizzi, G. L. Sciuto, C. Napoli, G. Pappalardo, E. Tramontana, A novel cloud-distributed toolbox for optimal energy dispatch management from renewables in igss by using wrnn predictors and gpu parallel solutions, 2014, pp. 1077 – 1084. doi:10.1109/SPEEDAM.2014.6872127.

[13] A. Sikora, A. Zielonka, M. F. Ijaz, M. Woźniak, Digital twin heuristic positioning of insulation in multimodal electric systems, IEEE Transactions on Consumer Electronics (2024).

[14] S. Pepe, S. Tedeschi, N. Brandizzi, S. Russo, L. Iocchi, C. Napoli, Human attention assessment using a machine learning approach with gan-based data augmentation technique trained using a custom dataset, OBM Neurobiology 6 (2022). doi:10.21926/obm.neurobiol.2204139.

[15] A. Alfarano, G. De Magistris, L. Mongelli, S. Russo, J. Starczewski, C. Napoli, A novel convmixer transformer based architecture for violent behavior detection 14126 LNAI (2023) 3 – 16. doi:10.1007/978-3-031-42508-0_1.

[16] W. Dong, M. Woźniak, J. Wu, W. Li, Z. Bai, Denoising aggregation of graph neural networks by using principal component analysis, IEEE Transactions on Industrial Informatics 19 (2022) 2385–2394.