Data Augmentation for Pseudo-Time Series Using Generative Adversarial Networks

Zakaria Salmi[1,*,†], José Luis Seixas Junior[1,‡]

[1] ELTE – Eötvös Loránd University, Faculty of Informatics, Budapest, Hungary

Abstract

Data augmentation techniques have been developed to address the challenge of acquiring large and diverse datasets for training machine learning models. In this paper, the focus is on time series data and proposing a Generative Adversarial Network (GAN) architecture based on Long Short-Term Memory (LSTM) for generating synthetic pseudo-time series data. The dataset is preprocessed by normalizing the series lengths and then designing the LSTM-GAN architecture, which consists of a generator network and a discriminator network. The generator network uses an LSTM layer to generate synthetic time series data, while the discriminator network distinguishes between real and synthetic data. LSTM-GAN is trained using an adversarial approach and update the network parameters iteratively. To evaluate the quality of the generated data, the original and synthetic data are compared using metrics such as silhouette score, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Our results show that the LSTM-GAN is capable of generating synthetic time series data that closely resembles the real data, as evidenced by similar silhouette score and low MSE and RMSE values. This work contributes to the field of data augmentation for time series data and demonstrates the effectiveness of GANs in generating realistic and complex time series data.

Keywords

Generative Adversarial Networks (GANs), Long Short-Term Memory (LSTM), Pseudo time series, Data augmentation

1. Introduction

Data plays a vital role in training machine learning models across various domains [1]. However, acquiring large and diverse datasets can be challenging and expensive, which can limit the performance and generalization ability of models. Data Augmentation (DA) techniques have been developed to address this issue by generating new training data from existing data, often through transformations such as rotation, translation, or cropping.

Over time, various techniques have been employed to generate synthetic data. Among these, the autoencoder (AE) [2] has been widely used. The AE architecture is designed to learn an effective low-dimensional representation of the input data and then reconstruct it back to its original form with maximum similarity. The AE models consist of an encoder and a decoder neural network. Although AE has been successful in generating synthetic data, alternative generative models have gained attention due to their ability to produce high-quality data and incorporate privacy protection mechanisms [3].

Recently, Generative Adversarial Networks (GANs) [4] have emerged as a powerful tool for data augmentation. GANs are generative models that can learn the underlying distribution of a given dataset and generate new realistic samples that are similar to the original data. This makes them well-suited for generating new training data that can augment smaller datasets and improve the performance of machine learning models.

While GANs have primarily been used in computer vision tasks such as image and video generation, there has been a growing interest in applying GANs to time series data. Time series data often have complex temporal structures and dependencies that make them challenging to model and generate. However, GANs have shown promising results in generating realistic time series data [5], imputing missing or corrupted data, and denoising signals.

Generating and accessing time series or pseudo-time series datasets can be challenging due to privacy concerns and difficulty in obtaining balanced or large datasets. This can pose a problem when training models with incomplete or unbalanced data, which can affect the quality of the output. Preprocessing techniques, such as subsampling, can be used to address these issues, as well as DA techniques, which are commonly used in datasets that are not large enough.

This paper aims to develop a GAN architecture based on Long Short-Term Memory (LSTM) that can generate synthetic pseudo-time series data. In Section 2, a comprehensive review of relevant literature, which has inspired

ITAT'23: Information technologies – Applications and Theory, September 22–26, 2023, Tatranské Matliare, Slovakia

[*] Corresponding author.

[†] Methodology, Software, Validation, Investigation, Data Curation, Visualization, Writing - Original Draft, Writing - Review&Editing.

[‡] Methodology, Conceptualization, Validation, Investigation, Supervision, Writing - Review&Editing.

q60lw0@inf.elte.hu (Z. Salmi); jlseixasjr@inf.elte.hu (J. L. Seixas Junior)

0009-0002-5895-5451 (Z. Salmi); 0000-0003-3948-8798 (J. L. Seixas Junior)

CEUR Workshop Proceedings (CEUR-WS.org)

the proposed approach, is provided. Section 3 describes the datasets used, the steps for applying the models, the specific models employed, the procedures utilized and the evaluation metrics. Section 4 presents the results obtained and provides a detailed discussion of these outcomes. Finally, in Section 5, the main conclusions drawn from the study are presented as suggestions for future work are provided.

2. Related Work

In recent years, there has been a surge in the publication of high-quality data augmentation papers [6, 7, 8]. However, it is noteworthy that a significant portion of these papers primarily concentrates on well-established domains such as image, video, or Natural language processing (NLP). Nonetheless, there is an emerging interest in investigating data augmentation techniques specifically tailored for time series data and pseudo-time series. These data types pose distinctive challenges that set them apart from other data formats. The evolving interest in this area underscores the recognition of the need for effective DA methods in addressing the unique requirements and complexities associated with time series and pseudo-time series data.

Wen et al. [9] presented a comprehensive taxonomy of data augmentation techniques for medical time series leveraging GANs. Their taxonomy encompasses a spectrum of methods, from fundamental to more advanced approaches. The authors also delve into deep generative models, such as the Recurrent GAN (RGAN) and Recurrent Conditional GAN (RCGAN) proposed by Esteban et al. [10]. These models demonstrate the capability to generate real-valued multi-dimensional time series data. This work contributes to the field by providing a systematic overview of GAN-based data augmentation methods in the context of medical time series, while also highlighting the relevance of deep generative models like RGAN and RCGAN for generating realistic and complex time series data.

In their study, Iglesias et al. [5] conducted an analysis of various GAN architectures and assessed their effectiveness in handling time series data. The paper specifically explores the utilization of recurrent neural networks (RNNs) within GAN frameworks for time series data. The authors introduce the Continuous Recurrent Neural Networks (C-RNN-GAN) model, which incorporates LSTM blocks as the primary learning structure and employs bidirectional recurrent networks in the discriminator. This approach aims to enhance the generation and evaluation of time series data, shedding light on the potential of utilizing RNN-based GAN architectures for time series data generation and provides valuable insights into the design of such models.

Brophy et al. [11] offer a comprehensive survey of GANs, encompassing their challenges, variations, and taxonomy. The paper extensively discusses different types of GANs, including discrete-variant and continuous-variant GANs, and presents a detailed overview of their taxonomy. Notable examples covered in the paper include Quant GAN, Sequentially Coupled GAN, and various other variants. This research serves as a valuable resource for both researchers and practitioners seeking to explore the application of GANs for time series analysis. The insights provided in this paper can aid in understanding the diverse landscape of GANs and inform the design and implementation of GAN-based approaches for analyzing time series data.

Farou et al. [12] and Singh et al. [13], even if working in different domains, cite problems of obtaining data such as high cost, difficulty due to privacy or location, these problems are inherent in research related to biological processes, which are also the object of this paper's studies. GANs can be beneficial in these cases because they have the ability to generate consistent data, where these synthetic data maintain the distribution of the originals. Furthermore, it is interesting to generate data not too far from decision boundaries, as points far from the boundaries do not change much the classification models.

3. Materials and Methods

To achieve the proposed goal of developing a GAN architecture based on LSTM for generating synthetic data, there are several steps, including data preprocessing, model design, training, and evaluation. The LSTM-GAN architecture is designed to generate synthetic time series data that closely resembles real data. The model configuration is as follows:

- **Generator:** The generator network comprises a single LSTM layer with 128 cells followed by a fully connected layer and an output layer. Dropout regularization is applied to mitigate overfitting and enhance generalization, thereby improving the model's performance.
- **Discriminator:** The discriminator network is composed of one LSTM layer, also with 128 cells, followed by fully connected layers and an output layer. Dropout regularization is incorporated in attempt to improve the model's ability to distinguish between real and generated samples.

3.1. Dataset

The dataset used was obtained by transforming leaf images into pseudo time-series data [14], which refers to series that have no time relationship between values, this dataset was chosen because the general objective of the project is to classify the vine varieties through leaves images. In the aforementioned work, the images were transformed into series, but for greater robustness, the transformation is invariant to translation, rotation or stretching, in addition to being dependent on color, thus, the DA operations commonly applied in images are not applicable in the original dataset, requiring the DA operations being performed on the series instead of images.

To ensure a focused analysis, dataset was divided into class-oriented subsets, which allows single class focus, where instead of generating data from all classes, the generation can happen in one class taking into account the others (or one another). Following this, the dataset was randomly split into training and testing sets, ensuring that time series samples from the same source or category are not present in both sets to prevent data leakage.

3.2. Data Preprocessing

Data preprocessing is the first step in the presented approach. To ensure uniformity in the input data the LSTM-GAN, series length normalization was applied by identifying the smallest series length among all the time series samples in the dataset, this reference length is denoted as l. For each time series sample, l points are randomly selected from the series indices. This random selection process guarantees that all time series have equal length l.

The indices are randomly generated, however the resampling process maintains the series order, because, even though, there is no time relationship among values in the series, there is a neighborhood relationship, which, in this way, is preserved.

Figure 1 visually depicts the shape of the original time series with different length.

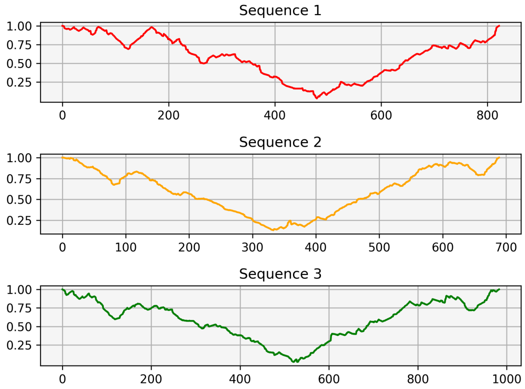

Figure 1: Original series with varying lengths.

On the other hand, Figure 2 shows series after they underwent the length normalization process. The figures exemplify how the overall shape of series are preserved after the procedure, but resulting in series with the same length.

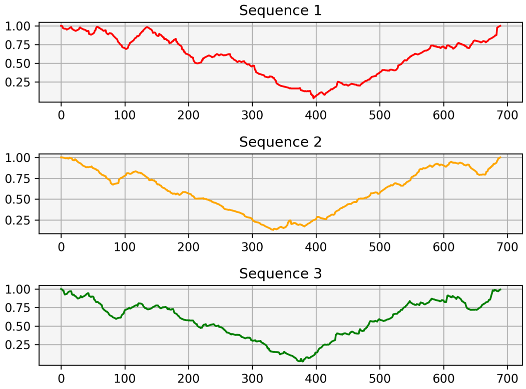

Figure 2: Normalized series with uniform length.

This procedure, being based on the shortest series, also helps to obtain shorter series from the other samples, facilitating the algorithm's performance.

3.3. Model Design

The LSTM-GAN architecture is specifically designed to capture the long-term dependencies present in time series data. This architecture consists of two key components: a generator network (G) and a discriminator network (D).

The generator network takes random noise $z \in \mathbb{R}^n$ as input and employs an LSTM layer to generate synthetic time series data. This LSTM layer is followed by fully connected layers and an output layer, which collectively transform the random noise into meaningful synthetic data.

On the other hand, the discriminator network's primary objective is to differentiate between real and synthetic time series data. It also employs an LSTM layer, followed by fully connected layers and an output layer, which enable it to effectively discern the authenticity of the input data.

In summary, the generator network utilizes the LSTM layer and subsequent layers to generate realistic synthetic time series data, while the discriminator network leverages the same architecture to accurately classify whether the input data is real or synthetic.

The two networks engage in a two-player minimax game defined by the value function $V(G, D)$, where $D(x)$ represents the probability that x comes from the real data rather than the generated data:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

The objective function, presented in Equation 1, consists of two terms, representing expectations over different distributions:

- The first term, $E_{x\sim p_{\text{data}}}(x)[\log D(x)]$, represents the expectation over real data samples (x) drawn from the real data distribution p_{data}. The discriminator D aims to maximize this term by correctly classifying real data instances and assigning a high probability to real data samples.
- The second term, $E_{z\sim p_z}(z)[\log(1 - D(G(z)))]$, represents the expectation over latent space samples (z) drawn from the latent distribution p_z. The generator G aims to minimize this term by producing synthetic data $(G(z))$ that the discriminator D incorrectly classifies as fake, as $(1 - D(G(z)))$ represents the probability of the discriminator D classifying the generated data as real.

The objective of the GAN framework is to find an equilibrium where the generator produces synthetic data that is indistinguishable from real data. This equilibrium is reached when the generator minimizes the objective function while the discriminator maximizes it, resulting in the generation of high-quality synthetic data [4].

3.4. Training

The training phase involves iteratively updating the parameters of the generator and discriminator networks to optimize their performance. The training process utilizes an adversarial approach, where the generator strives to deceive the discriminator by generating synthetic time series data that closely resembles real data. Conversely, the discriminator aims to effectively distinguish between real and synthetic data.

The network parameters are updated using backpropagation, which calculates the gradients based on the discriminator's feedback. These gradients are then used to update the weights of the generator and discriminator networks. The training continues until a convergence criterion is met, such as achieving a desired level of performance or reaching a maximum number of training iterations.

3.5. Evaluation with Post-Processing

A known problem in neural networks is the need for large volumes of information. Karras et al. [15] explain that small datasets make the feedback from the discriminator to the generator to be irrelevant and the network would diverge. Data augmentation would be a common technique for augmentation, but that, in this case, is exactly the problem to be solved while applying the algorithm.

Karras et al. [15] also state that this leads to noise being part of the generated data, so the application of filters that manages to smooth out noise that can be caused by the generator, which, even if not enough to confuse the discriminator, generate behaviors that are not suitable for series that describe shapes.

So, the quality assessment and evaluation of the generated synthetic data by the LSTM-GAN is performed after applying a post-processing techniques to the generated time series to make them resemble the original data more closely. One such technique used was the Gaussian filter, which helps smooth out the generated time series as it incorporates the neighborhood relation present in the series, which can be noisy due to the small amount of data inherent in the problem.

After applying the post-processing techniques, metrics that measure the similarity between the modified generated data and the real data were analyzed, to further demonstrate the effectiveness of the generator.

The evaluation process involved comparing the silhouette score between classes 1 and 2 for both the original and modified synthetic time series data, which has originally close to one hundred samples each. The silhouette score is a measure of cluster cohesion and separation, which in this case is applied to differentiate classes rather than clusters.

The silhouette score S_i can be calculated for any data point i as follows:

$$S_i = \frac{b_i - a_i}{\max(b_i, a_i)}$$

Here, b_i represents the average distances d of points j belonging to classes C_k different from the class assigned to the point i:

$$b_i = \min_{k\neq i} \frac{1}{|C_k|} \sum_{j\in C_k} d(i,j)$$

On the other hand, a_i represents the average distances of points j belonging to the same class C_i as the generated point i:

$$a_i = \frac{1}{|C_i| - 1} \sum_{j\in C_i, i\neq j} d(i,j)$$

The overall silhouette score is calculated as the average of all scores in the dataset.

Ideally, silhouette values are close to one between classes and close to zero between original and synthetic data within the same class [16]. Negative values would be values generated by one class but that actually should belong to another (misclassification).

In addition to the silhouette score, the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) were calculated to quantify the dissimilarity between the modified generated data and the real data.

MSE is a commonly used metric that measures the average squared difference between the predicted and actual values. In the context of evaluating synthetic data, MSE can be used to assess how closely the modified generated time series aligns with the real data. A lower MSE value indicates a higher similarity between the two.

RMSE is the square root of MSE and provides a more interpretable measure since it is in the same unit as the original data. RMSE allows us to understand the average magnitude of the prediction error in the original scale of the data. Similar to MSE, a lower RMSE value signifies a better alignment between the modified generated data and the real data.

To calculate MSE and RMSE, each data point in the modified generated time series is compared to its corresponding real data point. The squared differences are summed up and then divided by the total number of data points to obtain the average squared difference (MSE). Taking the square root of MSE gives us the RMSE value.

By applying post-processing techniques such as the Gaussian filter and incorporating MSE and RMSE in our evaluation, a comprehensive assessment of the similarity between the modified generated time series and the real data can be provided, complementing the silhouette score measure.

4. Results

The quality of the data generated by the LSTM-GAN model is evaluated by comparing it to the original data and analyzing the characteristics of different classes.

4.1. Comparison of Original and Synthetic Time Series

To assess the performance of the model, original and synthetic time series data from classes 1 and 2 are compared. Figures 3 show the original and synthetic time series of class 1, while Figures 4 show the original and synthetic time series of class 2.

From the figures, it is possible to observe that the synthetic time series data closely resembles the patterns and characteristics of the original data. The synthetic time series of class 1 (Figure 3) exhibits similar trends, peaks, and fluctuations as the original data. Similarly, the synthetic time series of class 2 captures the distinctive patterns and variations present in the original data (Figure 4).

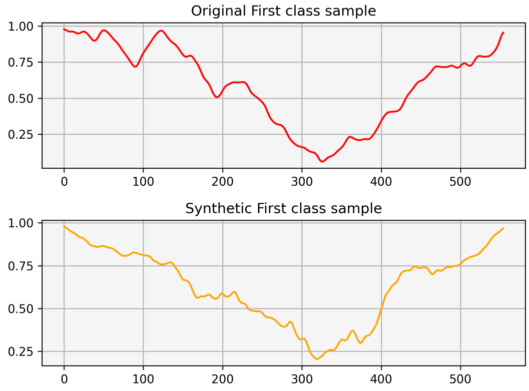

Figure 3: Original and Synthetic series of class 1.

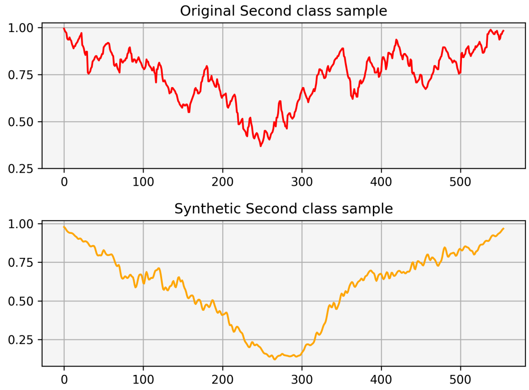

Figure 4: Original and Synthetic series of class 2.

These results indicate that our LSTM-GAN model successfully learns the underlying patterns and structures of the original time series data and generates synthetic data that preserve these characteristics.

4.2. Evaluation Metrics

To quantitatively assess the quality of the synthetic time series data, three evaluation metrics were computed, mean squared error (MSE), root means squared error (RMSE), and silhouette score. These metrics measure the similarity between the original and synthetic data, providing insights into the accuracy and fidelity of the generated time series.

Table 1 presents the evaluation metrics for the comparison between the original and synthetic time series of different classes.

The evaluation metrics demonstrate that the synthetic time series data achieves low MSE and RMSE values, indicating a close resemblance to the original data.

Table 1
Evaluation metrics for original and synthetic time series.

Class	MSE	RMSE
Class 1	0.0286	0.0326
Class 2	0.0326	0.0326

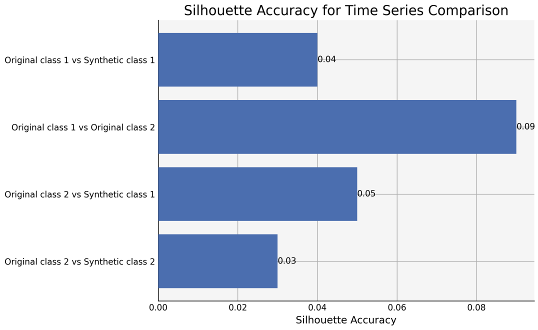

Figure 5: Comparison of silhouette score between original and synthetic data for class 1 and class 2.

From Figure 5, the fact that silhouette scores for both the original and synthetic data are consistently close to 0, when comparing classes 1 and 2, suggests a high degree of overlap and limited separation between these classes in both the real and synthetic time-series data. It can be observed that the values have the same behavior even with the original series.

The presence of overlapping classes in the original data explains the similar silhouette scores obtained with the synthetic data. The model captures this inherent overlap during the data generation process, resulting in synthetic data that faithfully reflects the characteristics of the original data.

An important characteristic to be observed by the silhouette scores is that the comparisons between classes generated with originals are smaller, consequently closer than between classes (original or generated). That is, the generation preserves the distribution with some overlapping, even having some distance from the original points, which is required for synthetic data.

Thus, as classes have overlapping, shown by the values between original classes 1 and 2, the synthetic data will also have some overlapping, not necessarily generating fully separable values. So, what is expected from synthetic data is that they can reinforce their classes, but not create data that, even if beneficial if fully separable, would not represent their respective classes well.

The post-processing of the series indicates that the general shape of the series is more important for its identification than specific points, since the Gaussian filter did not deteriorate the series, nor did it cause large distances between original and synthetic data.

5. Conclusion

In this paper, experiments were conducted to develop an LSTM-GAN architecture for generating synthetic pseudo-time series data. The experiments involved dataset preparation, model configuration, training, post-processing, and evaluation. The results indicate that the LSTM-GAN can successfully generate synthetic time series data that closely resemble the real data. These findings contribute to the field of time series data generation and showcase the potential applications of GANs in this domain.

Overall, the experiment chapter provides insights into the development and evaluation of an LSTM-GAN architecture, laying the foundation for further advancements in synthetic time series data generation.

While the silhouette scores may be low, it is crucial to emphasize that the synthetic data still carries valuable information and can be effectively utilized in various applications, such as data augmentation or training robust classifiers. Despite the overlapping nature of the classes 1 and 2, the synthetic data serves as a valuable resource for enhancing the diversity and quantity of available data, contributing to the overall performance and generalization capability of models trained on it.

These results confirm that our LSTM-GAN model successfully generates synthetic time series data that accurately captures the patterns, trends, and characteristics of the original data, making it a valuable resource for various applications in time series analysis and modeling.

Future research directions could focus on exploring different model architectures, incorporating additional components like attention mechanisms, or applying transfer learning techniques to leverage pre-trained models for improved performance.

Acknowledgement

We would like to thank Telekom who has us as one of its technology partners on Telekom Innovation Laboratories and the Tempus Public Foundation for the financial support through the Stipendium Hungaricum Scholarship Programme.

References

[1] G. Foody, M. B. McCulloch, W. B. Yates, The effect of training set size and composition on artificial neural network classification, International Journal of Remote Sensing 16 (1995) 1707–1723. doi:10.1080/01431169508954507.

[2] D. Bank, N. Koenigstein, R. Giryes, Autoencoders, CoRR abs/2003.05991 (2020). arXiv:2003.05991.

[3] Y. Qu, S. Yu, J. Zhang, H. T. T. Binh, L. Gao, W. Zhou, GAN-DP: Generative adversarial net driven differentially privacy-preserving big data publishing, in: ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–6. doi:10.1109/ICC.2019.8761070.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Communications of the ACM 63 (2020) 139–144.

[5] G. Iglesias, E. Talavera, Á. González-Prieto, A. Mozo, S. Gómez-Canaval, Data augmentation techniques in time series domain: a survey and taxonomy, Neural Computing and Applications (2023) 1–23.

[6] Y. Chen, X.-H. Yang, Z. Wei, A. A. Heidari, N. Zheng, Z. Li, H. Chen, H. Hu, Q. Zhou, Q. Guan, Generative adversarial networks in medical image augmentation: a review, Computers in Biology and Medicine (2022) 105382.

[7] A. Kammoun, R. Slama, H. Tabia, T. Ouni, M. Abid, Generative adversarial networks for face generation: A survey, ACM Computing Surveys 55 (2022) 1–37.

[8] S. Yu, J. Tack, S. Mo, H. Kim, J. Kim, J.-W. Ha, J. Shin, Generating videos with dynamics-aware implicit generative adversarial networks (2022). arXiv:202202.1057102.12478.

[9] Q. Wen, L. Sun, X. Song, J. Gao, X. Wang, H. Xu, Time series data augmentation for deep learning: A survey, CoRR (2020). arXiv:2002.12478.

[10] C. Esteban, S. L. Hyland, G. Rätsch, Real-valued (medical) time series generation with recurrent conditional gans (2017). arXiv:1706.02633.

[11] E. Brophy, Z. Wang, Q. She, T. Ward, Generative adversarial networks in time series: A survey and taxonomy (2021). arXiv:2107.11098.

[12] Z. Farou, N. Mouhoub, T. Horváth, Data generation using gene expression generator, in: C. Analide, P. Novais, D. Camacho, H. Yin (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2020, Springer International Publishing, Cham, 2020, pp. 54–65.

[13] D. Singh, E. Merdivan, S. Hanke, J. Kropf, M. Geist, A. Holzinger, Convolutional and recurrent neural networks for activity recognition in smart environment, in: A. Holzinger, R. Goebel, M. Ferri, V. Palade (Eds.), Towards Integrative Machine Learning and Knowledge Extraction, Springer International Publishing, Cham, 2017, pp. 194–205.

[14] J. L. Seixas Jr., T. Horváth, KNN algorithm with dtw distance for signature classification of wine leaves, in: 20th Conference Information Technologies - Applications and Theory (ITAT 2020), Oravská Lesná, Slovakia, 2020, pp. 130–136.

[15] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, T. Aila, Training generative adversarial networks with limited data, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[16] K. R. Shahapure, C. Nicholas, Cluster quality analysis using silhouette score, in: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), 2020, pp. 747–748. doi:10.1109/DSAA49011.2020.00096.