Data Augmentation for Pseudo-Time Series Using Generative Adversarial Networks

Zakaria Salmi, José Luis Seixas Junior
q60lw0@inf.elte.hu, jlseixasjr@inf.elte.hu

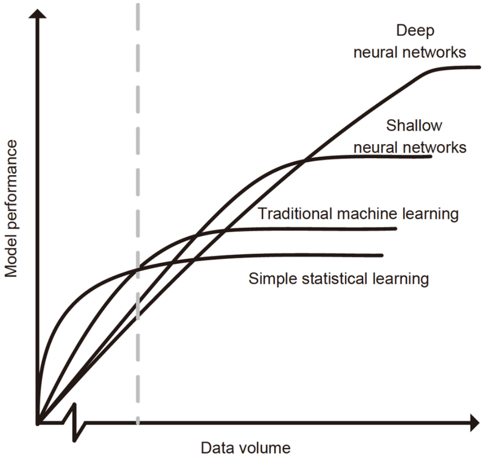
ITAT Conference 2023

Agenda

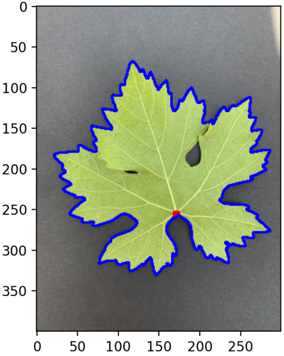
- Introduction
- Objective
- Methodology
- Results

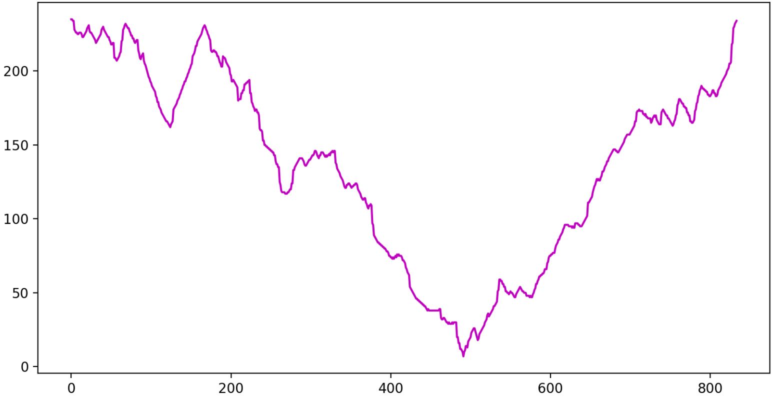
Data is expensive!

- Obtaining extensive and diverse datasets can pose significant challenges and financial constraints.
- This limitation can adversely affect the performance and generalization capabilities of machine learning models.
- To mitigate this challenge, Data Augmentation (DA) techniques have been developed as effective solutions.

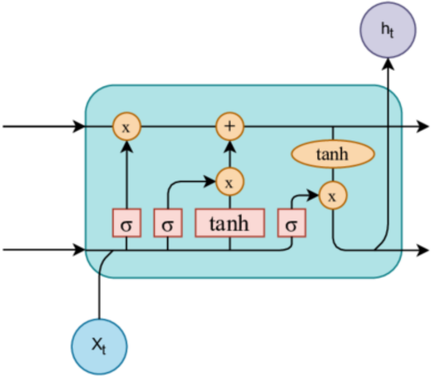


Pseudo-Time Series



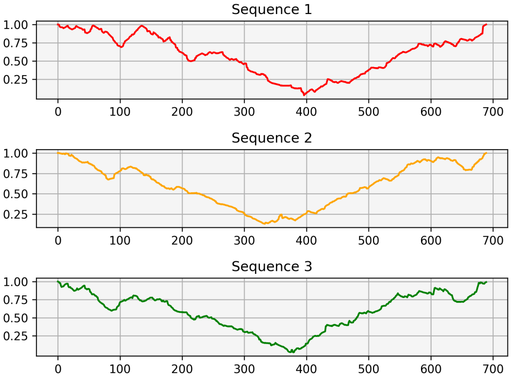


Objective



- This paper aims to develop a GAN architecture based on Long Short-Term Memory (LSTM) that can generate synthetic pseudo-time series data.

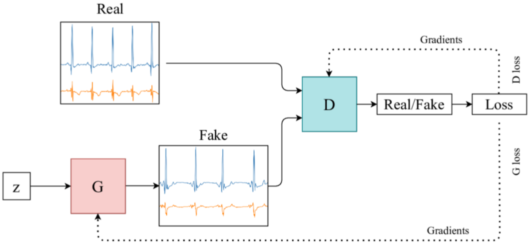
Dataset



- **Pseudo-time series**: there is no time relationship in the series.
- **Standardization**: length of the shortest series.

Generative Adversarial Network

- GANs are generative models that can learn the underlying distribution of a given dataset.
- GAN generates new realistic samples that are similar to the original data.



Model Design

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:

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

Model Design

- The discriminator, denoted as D, strives to correctly classify real data and assign a high probability to genuine samples.
- The generator, represented as G, aims to create synthetic data ($G(z)$) that confuses the discriminator D, making it classify the generated data as fake. This involves minimizing the term $(1 - D(G(z)))$.
- In the GAN framework, the goal is to achieve a balance where the generator generates synthetic data that is indistinguishable from real data. This balance is achieved when the generator minimizes the objective function while the discriminator maximizes it, resulting in the creation of high-quality synthetic data.

Model Design

Generator: The generator network comprises a single LSTM layer with 128 cells followed by a dropout then a fully connected layer and an output layer.

Discriminator: The discriminator network is composed of one LSTM layer, also with 128 cells, followed by a dropout then a fully connected layers and an output layer.

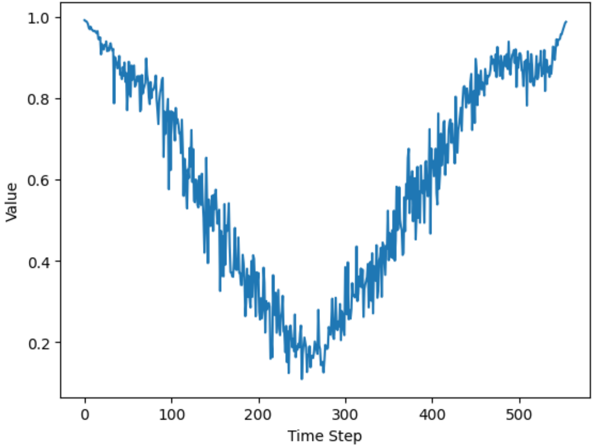
Training

Algorithm 1 Training Loop

```
function TRAIN
    for iteration ← 1 to Num_Iterations do
        for real_data in Train_Data_Loader do
            DiscriminatorOptimizer.zero_grad()                              ▷ Training Discriminator
            noise ← random(size)
            fake_data ← Generate_Fake_Data(noise)
            discriminator_loss ← Calculate_Discriminator_Loss(real_data, fake_data)
            discriminator_loss.backward()
            DiscriminatorOptimizer.step()

            GeneratorOptimizer.zero_grad()                                  ▷ Training Generator
            noise ← random(size)
            fake_data ← Generate_Fake_Data(noise)
            generator_loss ← Calculate_Generator_Loss(fake_data)
            generator_loss.backward()
            GeneratorOptimizer.step()
```

Post Processing

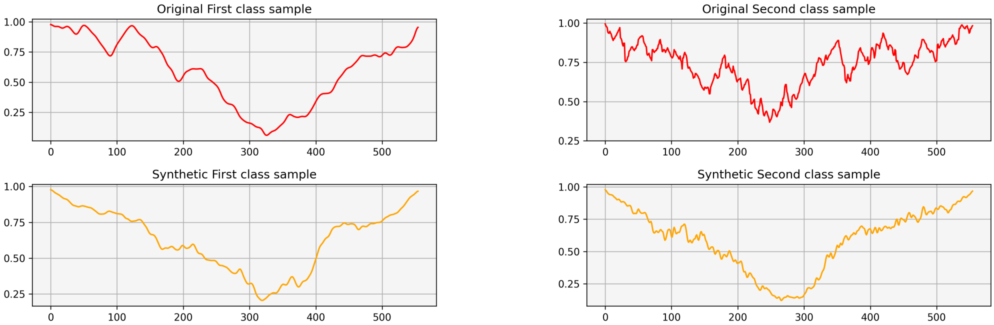


- Applied post-processing techniques, such as the Gaussian filter, to refine generated time series, aligning them more closely with the original data.

Evaluation Metrics

- The evaluation process involved comparing the silhouette score between classes 1 and 2 for both the original and modified synthetic time series data.
- 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.

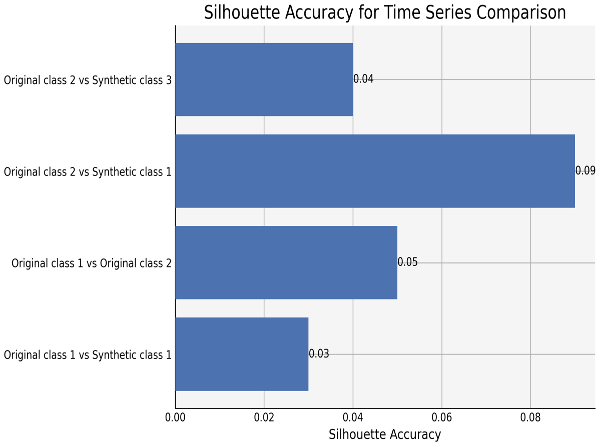
Results



Synthetic time series in class 1 mirrors the trends, peaks, and fluctuations of the original data, while class 2's synthetic time series faithfully replicates distinctive patterns and variations.

Silhouette Score

- Silhouette scores for classes 1 and 2 consistently hover near 0 in both the original and synthetic data, indicating substantial overlap and limited separation between these classes.
- The model adeptly reproduces this inherent overlap during data generation, resulting in synthetic data that faithfully replicates the original dataset's characteristics.



Silhouette Score

- Silhouette scores in the generated data closely mimic those in the original data, showing a balance between preserving the distribution and allowing some overlap. This equilibrium is crucial for meaningful synthetic data generation.
- Synthetic data aims to strengthen class distinctions without making data points entirely separable, which would be unrealistic. The objective is to maintain representational fidelity while enhancing class separability.
- Post-processing, using a Gaussian filter, underscores the importance of a series' general shape for identification. This approach maintains the integrity of the series without introducing significant deviations between original and synthetic data.

MSE and RMSE

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

Class	MSE	RMSE
Class 1	0.0286	0.0326
Class 2	0.0326	0.0326

Conclusion

- The results indicate that the LSTM-GAN can successfully generate synthetic time series data that closely resemble the real data.
- Although silhouette scores are low, synthetic data remains valuable for applications like data augmentation and training classifiers. Despite the class overlap, it enhances data diversity and quantity, benefiting model performance and generalization.
- Research directions could explore diverse model architectures, integrate attention mechanisms, and employ transfer learning techniques using pre-trained models to enhance performance.

Thank you!