Fakultät für Informatik
Facoltà di Scienze e Tecnologie informatiche
Faculty of Computer Science

efre·fesr Südtirol · Alto Adige
Europäischer Fonds für regionale Entwicklung
Fondo europeo di sviluppo regionale
EUROPEAN UNION
AUTONOME PROVINZ BOZEN SÜDTIROL
PROVINCIA AUTONOMA DI BOLZANO ALTO ADIGE

SpectralGAN: Spectral Data Augmentation Using Generative Adversarial Networks for wood-type classification

Attaullah Buriro[1]

[1]Free University of Bozen-Bolzano

attaullah.buriro@unibz.it

16.9.2021

Presentation outline

- **What is Data augmentation and why it is needed?**

Presentation outline

- What is Data augmentation and why it is needed?
- **Why Generative adversarial Networks (GANs) for Data augmentation?**

Presentation outline

- What is Data augmentation and why it is needed?
- Why Generative adversarial Networks (GANs) for Data augmentation?
- **Our GAN**

Presentation outline

- What is Data augmentation and why it is needed?
- Why Generative adversarial Networks (GANs) for Data augmentation?
- Our GAN
- **Methodology**

Presentation outline

- What is Data augmentation and why it is needed?
- Why Generative adversarial Networks (GANs) for Data augmentation?
- Our GAN
- Methodology
- **Preliminary Results**

Data Augmentation Basics

Why Data Augmentation?

- **Advance machine learning models (e.g., CNN) perform well on increased number of samples**
  - **As collecting and labeling of data could be exhausting and may incur expenses**

Why Data Augmentation?

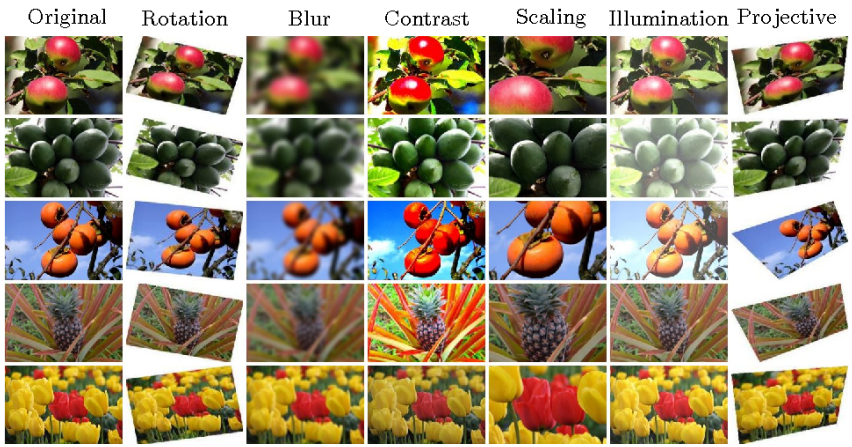
- Advance machine learning models (e.g., CNN) perform well on increased number of samples
  - As collecting and labeling of data could be exhausting and may incur expenses
- **Data augmentation is useful to improve performance of machine learning models by adding transformed or synthetic examples to the training datasets**.

Why Data Augmentation?

- Advance machine learning models (e.g., CNN) perform well on increased number of samples
  - As collecting and labeling of data could be exhausting and may incur expenses
- Data augmentation is useful to improve performance of machine learning models by adding *transformed* or *synthetic* examples to the training datasets.
- **Geometric Transformations (making simple alterations on visual data) has shown to be popular. However, Generative Adversarial Networks (GANs) are gaining the required momentum. Unlike geometric tranformation, GAN generate new synthetic samples**

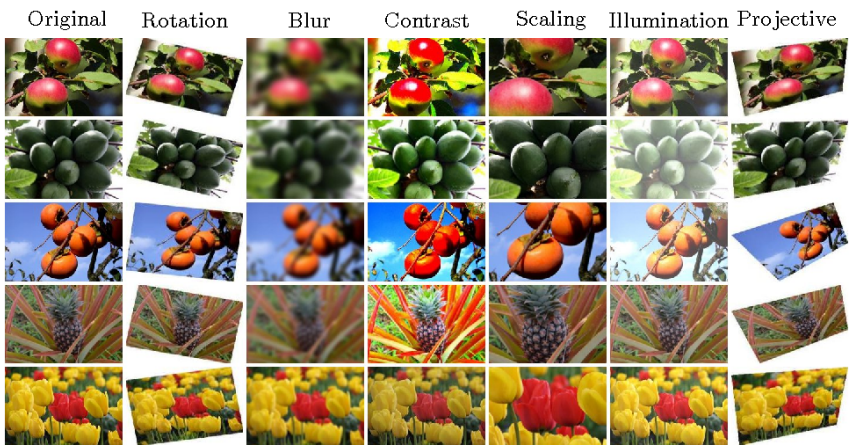
What is Data Augmentation?

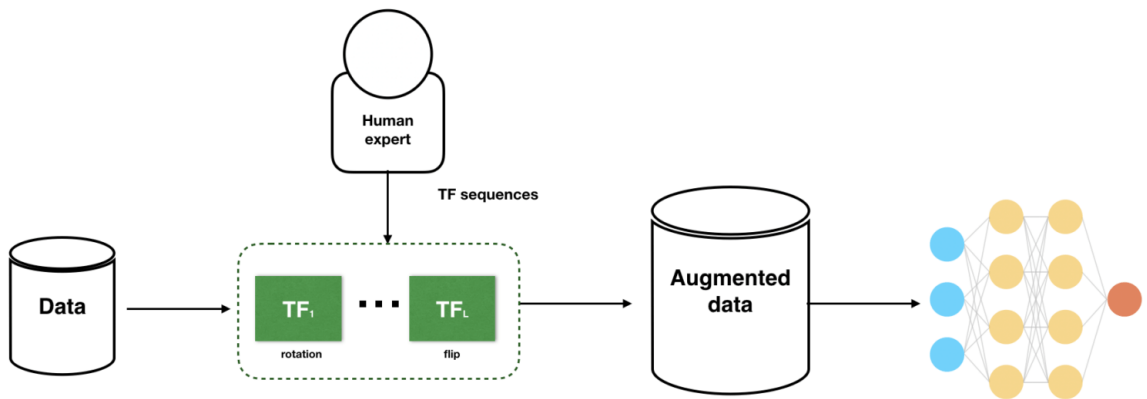
- **Techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data - Wikipedia.**



What is Data Augmentation?

- Techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data - Wikipedia.





[1]

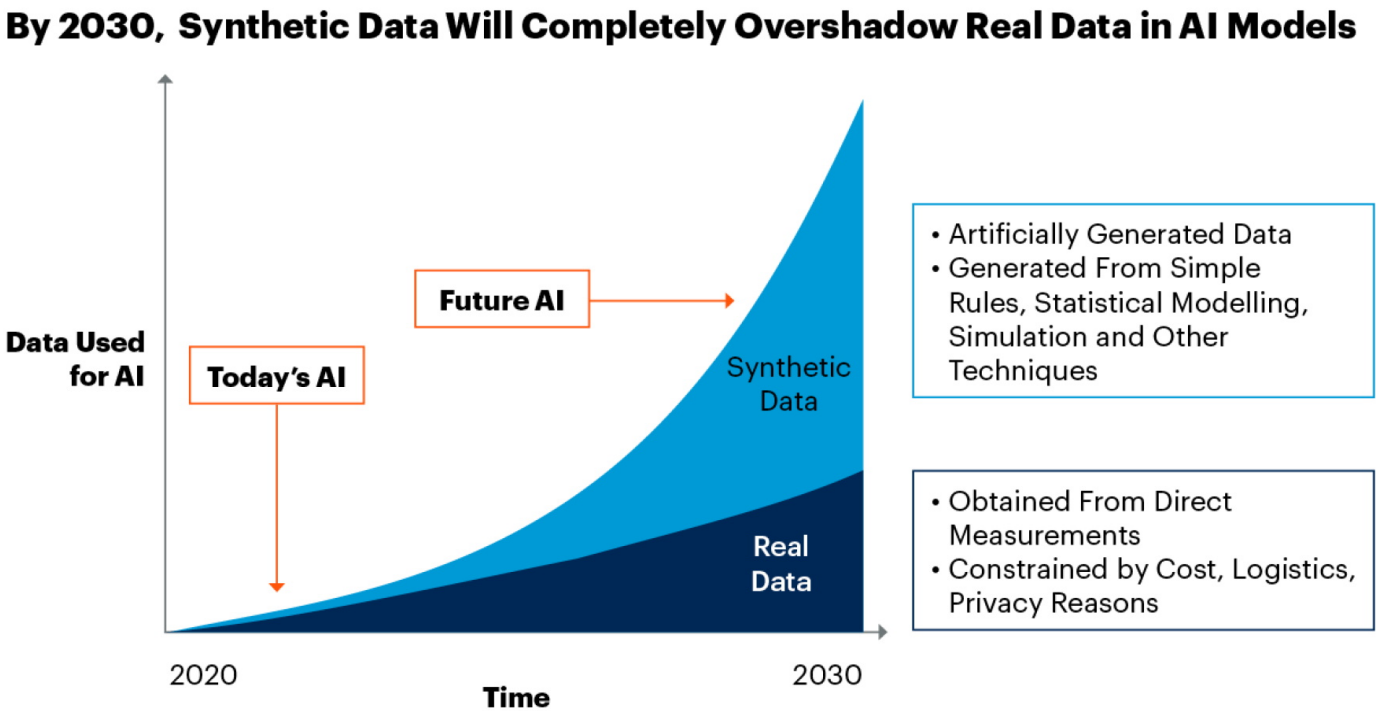
[1] https://research.aimultiple.com/data-augmentation/

Data Augmentation using GANs

- **GAN generates synthetic data to match sample data while ensuring that the important statistical properties of sample data are reflected in synthetic data.**

Data Augmentation using GANs

- GAN generates synthetic data to match sample data while ensuring that the important statistical properties of sample data are reflected in synthetic data.
- **It's estimated that by 2024, 60% of the data used for the development of AI and analytics projects will be synthetically generated**



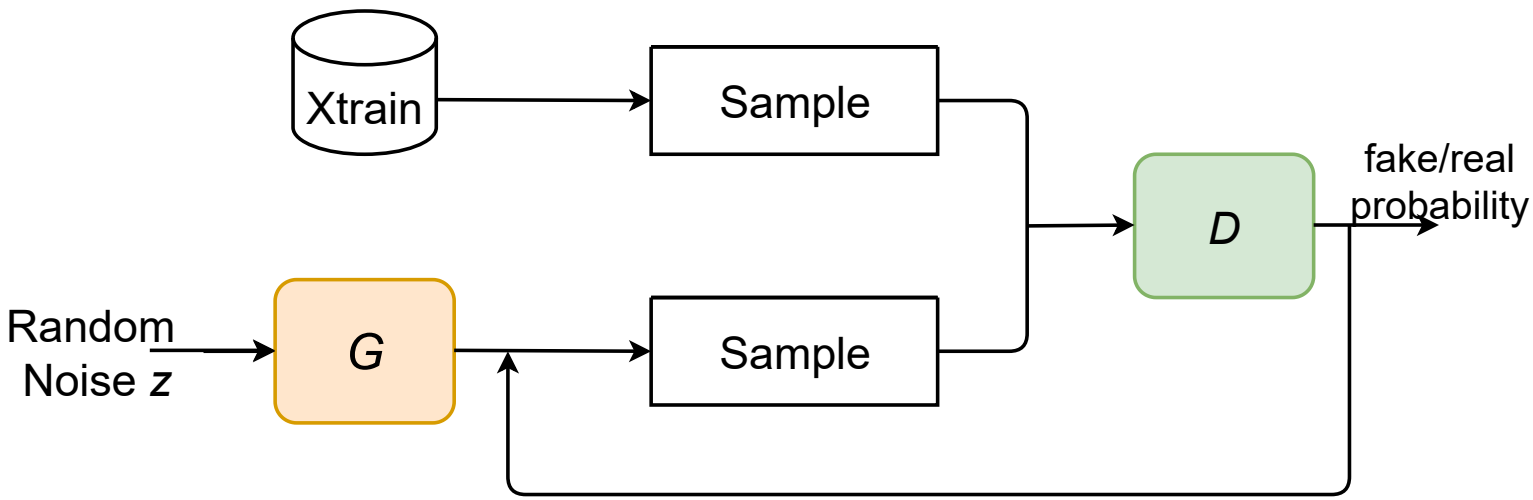
Generative Adversarial Networks

Generative Adversarial Networks

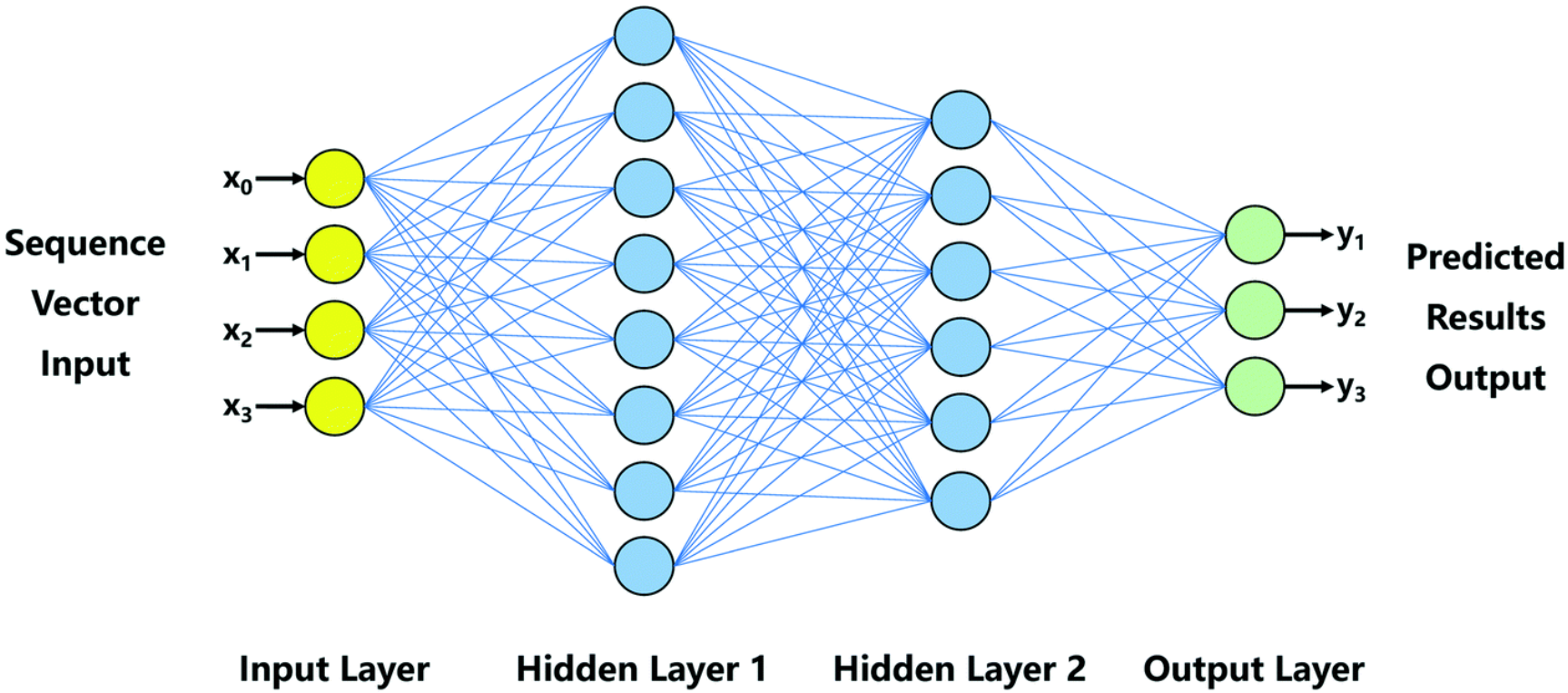
- **GANs have shown to be very popular in almost every field, i.e., image & video generation, speech generation, style transfer, biometrics, etc**

Generative Adversarial Networks

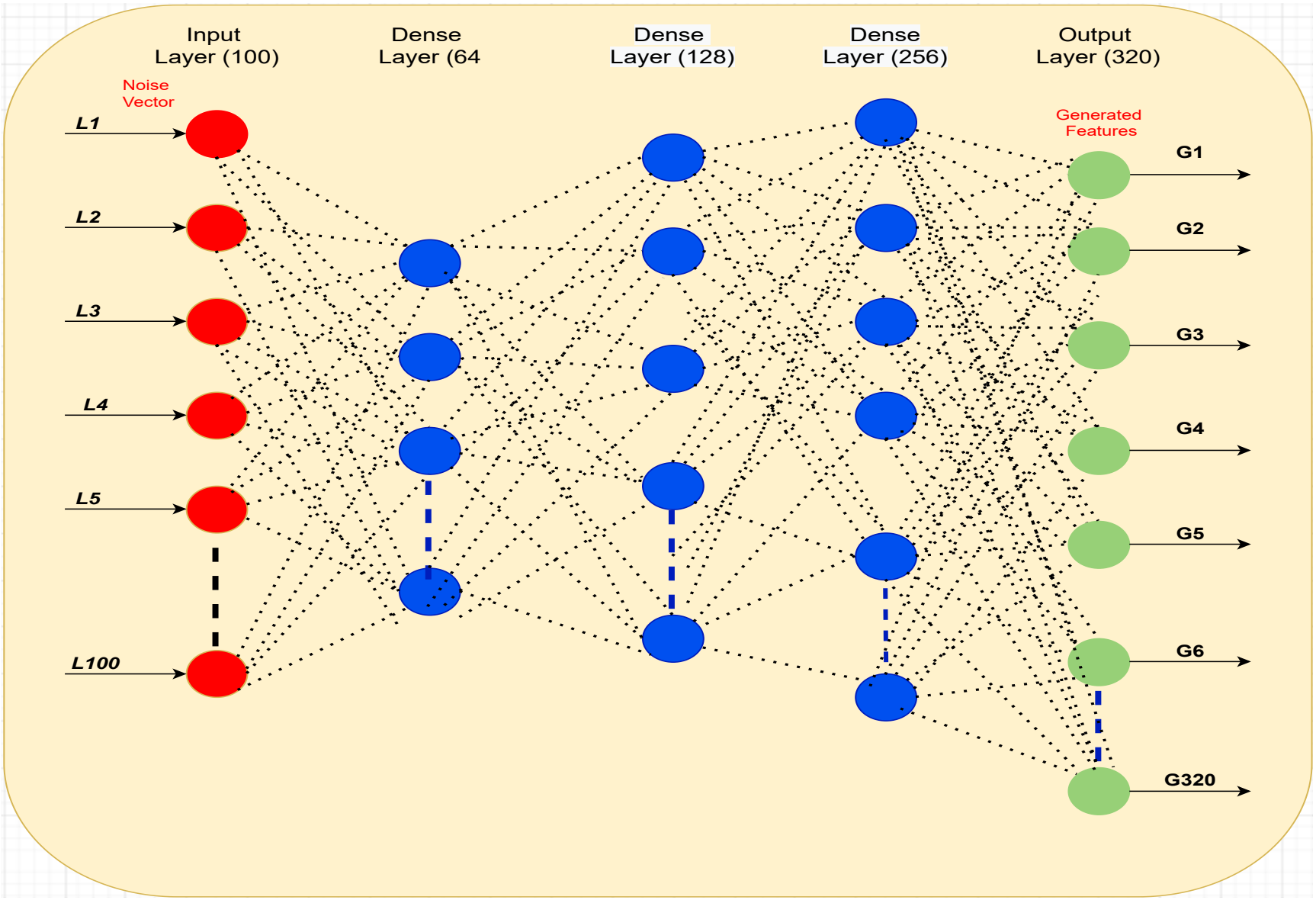
- GANs have shown to be very popular in almost every field, i.e., image & video generation, speech generation, style transfer, biometrics, etc
- **GAN Architecture**
  - **Two networks, i.e G for generator, and D for discriminator, are staked**



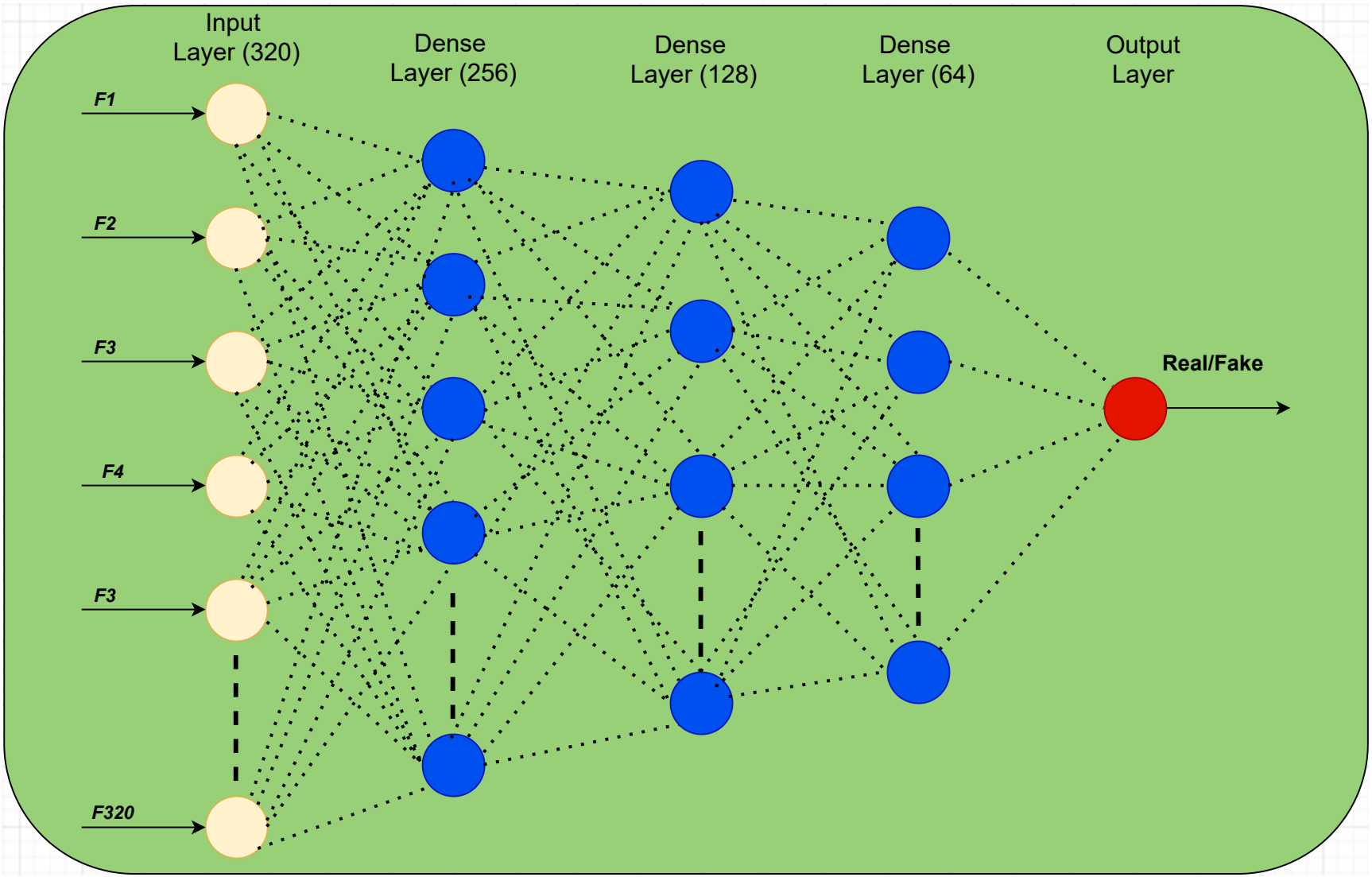
Networks development using Multilayer Perceptron



Generator


Input Layer (100)
Dense Layer (64
Dense Layer (128)
Dense Layer (256)
Output Layer (320)
Noise Vector
L1
L2
L3
L4
L5
L100
Generated Features
G1
G2
G3
G4
G5
G6
G320

Discriminator



Methodology

SpectralGAN Dataset

- **Hyperspectral Wood dataset[2] of 3 classes (Heartwood Vs Sapwood Vs Background)**

[2] provided by Microtec - thanks to them

SpectralGAN Dataset

- Hyperspectral Wood dataset[2] of 3 classes (Heartwood Vs Sapwood Vs Background)
- **# of processed cuboids = 132 + 132 + 66 + 66 = 396 ($32 \times 32 \times 320$)**

[2] provided by Microtec - thanks to them

SpectralGAN Dataset

- Hyperspectral Wood dataset[2] of 3 classes (Heartwood Vs Sapwood Vs Background)
- # of processed cuboids = 132 + 132 + 66 + 66 = 396 (32×32×320)
- **# of training cuboids = 264 (132 for each class)**
- **# of testing cuboids = 132 (66 for each class)**

[2] provided by Microtec - thanks to them

SpectralGAN Dataset

- Hyperspectral Wood dataset[2] of 3 classes (Heartwood Vs Sapwood Vs Background)
- # of processed cuboids = 132 + 132 + 66 + 66 = 396 (32×32×320)
- # of training cuboids = 264 (132 for each class)
- # of testing cuboids = 132 (66 for each class)
- **# of training samples = 270336**
- **# of testing samples = 135768**

[2] provided by Microtec - thanks to them

SpectralGAN Dataset

- Hyperspectral Wood dataset[2] of 3 classes (Heartwood Vs Sapwood Vs Background)
- # of processed cuboids = 132 + 132 + 66 + 66 = 396 (32×32×320)
- # of training cuboids = 264 (132 for each class)
- # of testing cuboids = 132 (66 for each class)
- # of training samples = 270336
- # of testing samples = 135768
- **We have a baseline CNN-based Spectral classifier (implemented by Sun Boyuan) for evaluation**

[2] provided by Microtec - thanks to them

Spectral Classifier

- **We have a baseline CNN-based Spectral classifier for evaluation**

```
class Cifar10(nn.Module):
    def __init__(self, in_channels=1,
                 num_classes=2,
                 kernel_size=(1, 3),
                 padding_size=(0, 1),
                 num_feature_fc=64):
        super(Cifar10, self).__init__()
        self.conv1 = nn.Conv2d(in_channels, 32, kernel_size=kernel_size, padding=padding_size)
        self.conv2 = nn.Conv2d(32, 32, kernel_size=kernel_size, padding=padding_size)
        self.conv3 = nn.Conv2d(32, 32, kernel_size=kernel_size, padding=padding_size)
        #self.pool = nn.MaxPool2d((1, 3), stride=(1, 2),padding=(0,1))
        self.pool = nn.MaxPool2d((1, 3),stride=(1,2))
        self.fc1 = nn.Linear(32*39, num_feature_fc)
        self.fc2 = nn.Linear(num_feature_fc, 2)

    def forward(self, x):
        in_size = x.size(0)
        x = F.relu(self.conv1(x))
        x = self.pool(x)
        x = F.relu(self.conv2(x))
        x = self.pool(x)
        x = F.relu(self.conv3(x))
        x = self.pool(x)
        x = x.reshape(x.shape[0], -1)  # flatten the tensor
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

SpectralGAN Evaluation

- train and test on all the original samples
- generation of synthetic heartwood and sapwood samples (after 10000 epochs)
- train the model on 25%, 50%, and 75%, real data simulataneoulsy and test test on the original test set (to get the reference result).
- train the classifier on augmented dataset[3] and test on the original test data
- compute and report the contribution of synthetic samples towards accuracy

[3] on 25% original and 75% generated, on 50% original and 50% generated, and 75% original and 25% generated samples

SpectralGAN Synthetic wood sample generation

- **Setting-1**
  - **Discriminator training on 25%cubes**
  - **generation of 75% synthetic cubes**

SpectralGAN Synthetic wood sample generation

- Setting-1
  - Discriminator training on 25%cubes
  - generation of 75% synthetic cubes
- **Setting-2**
  - **Discriminator training on 50% cubes**
  - **generation of 50% cubes**

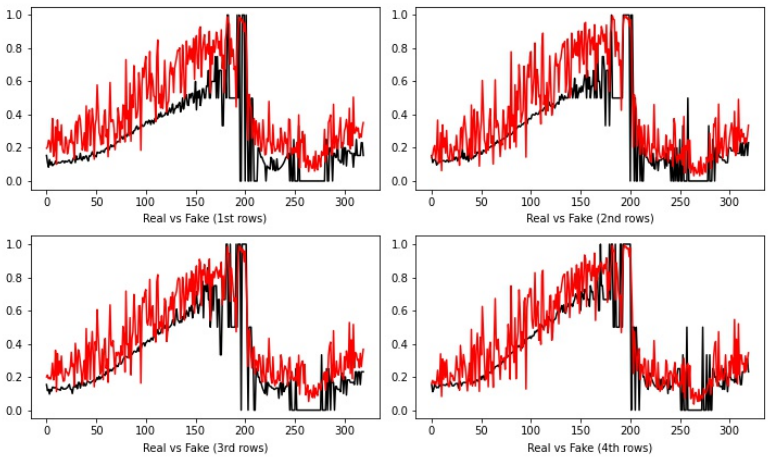
SpectralGAN Synthetic wood sample generation

- Setting-1
  - Discriminator training on 25%cubes
  - generation of 75% synthetic cubes
- Setting-2
  - Discriminator training on 50% cubes
  - generation of 50% cubes
- **Setting-3**
  - **Discriminator training on of 75% cubes**
  - **generation of 25% cubes**

Visual Evaluation

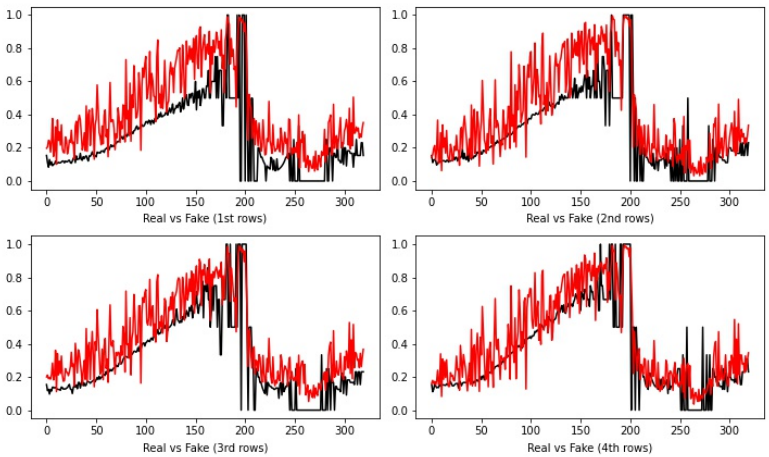
SpectralGAN Visual Analysis

- **Generation of synthetic samples**



SpectralGAN Visual Analysis

- Generation of synthetic samples



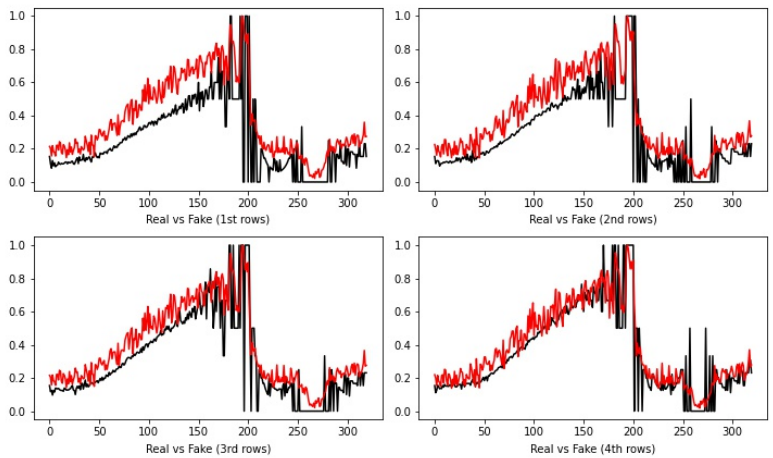
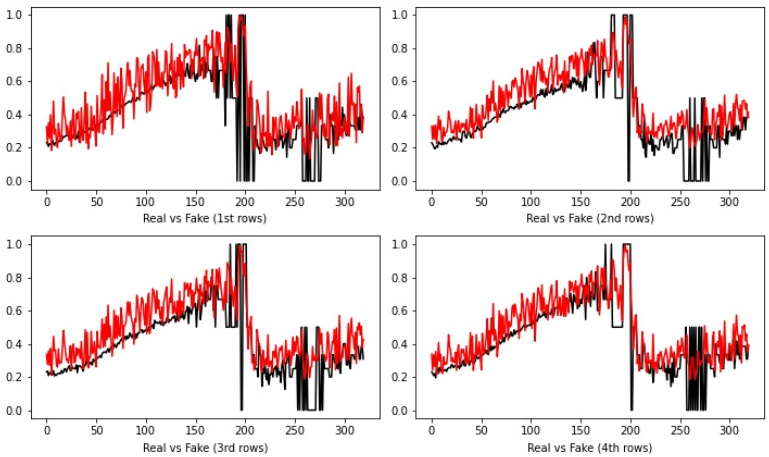


Figure: The comparison of original samples of Hrt (black) and the GAN generated non-real (red) samples (for training on 75% samples) (a), generated after 1000 epochs and (b) after 10000 epochs

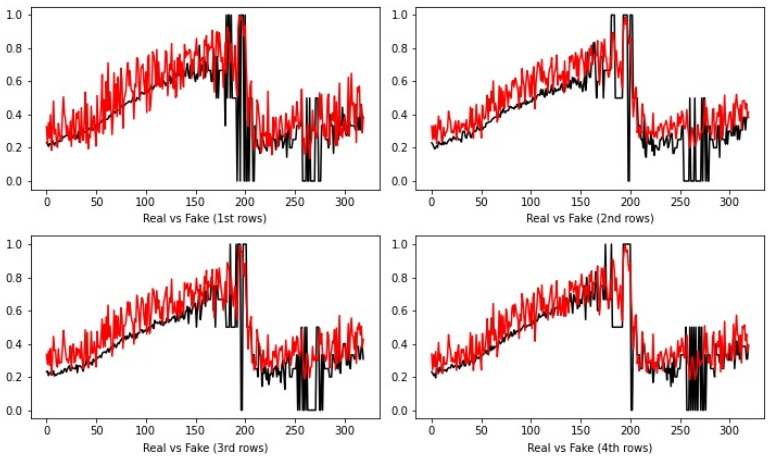
SpectralGAN Visual Analysis

- **Generation of synthetic samples**



SpectralGAN Visual Analysis

- Generation of synthetic samples



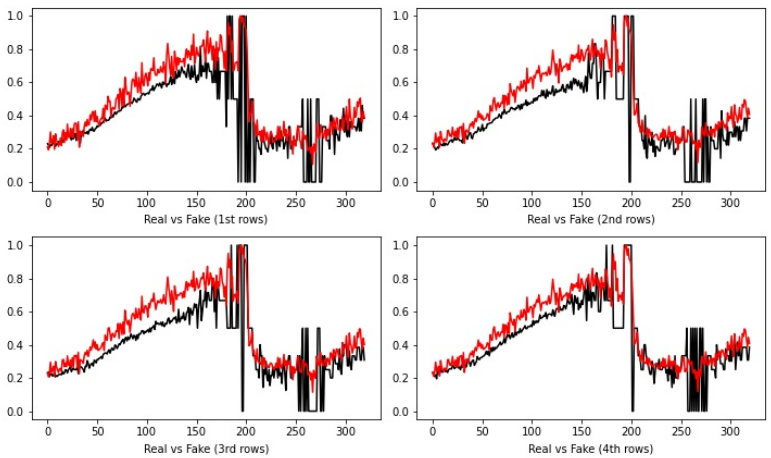
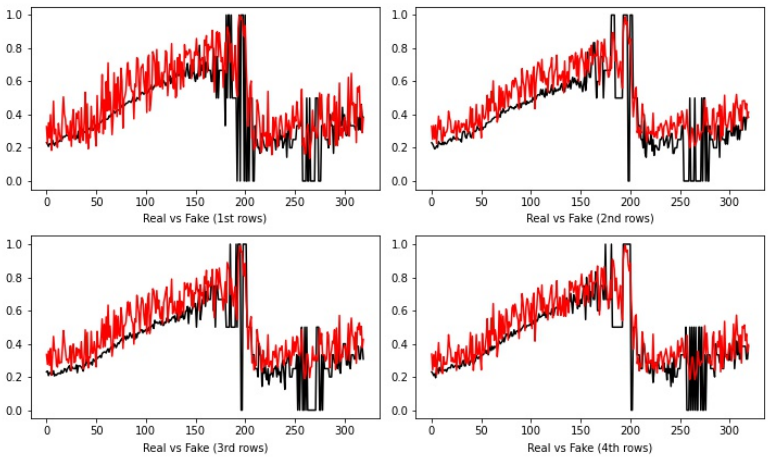


Figure: The comparison of original samples of Sap (black) and the GAN generated non-real (red) samples (for training on 50% samples) (a), generated after 1000 epochs and (b) after 10000 epochs

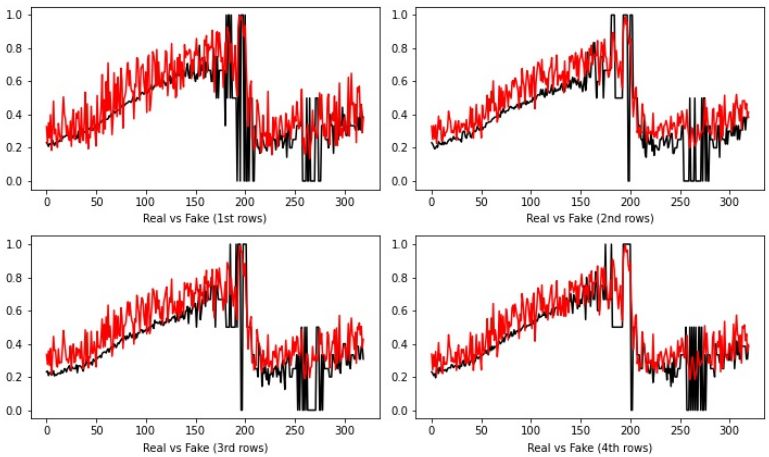
SpectralGAN Visual Analysis

- **Generation of synthetic samples**



SpectralGAN Visual Analysis

- Generation of synthetic samples



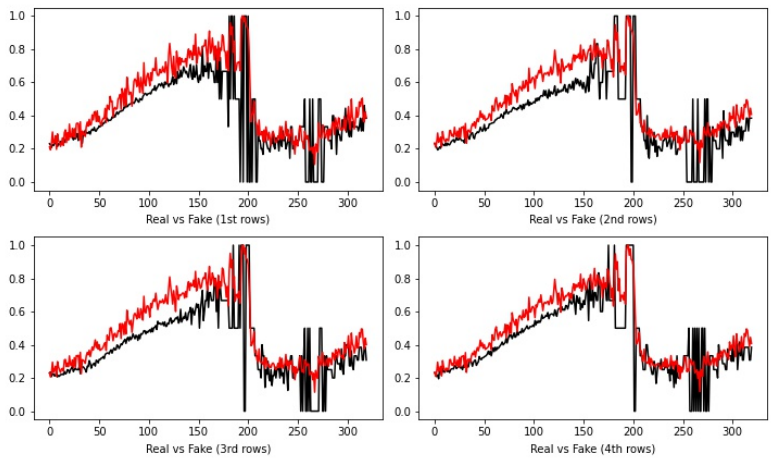
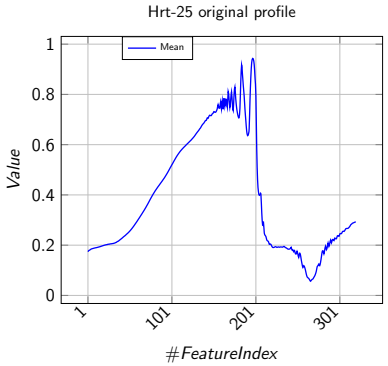
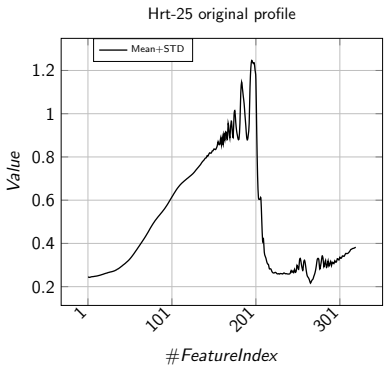
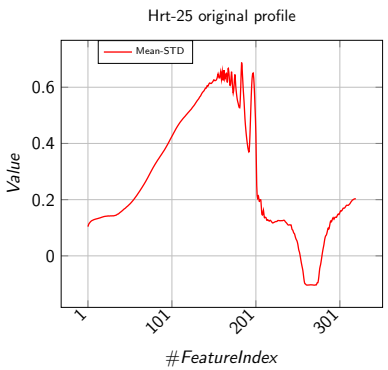


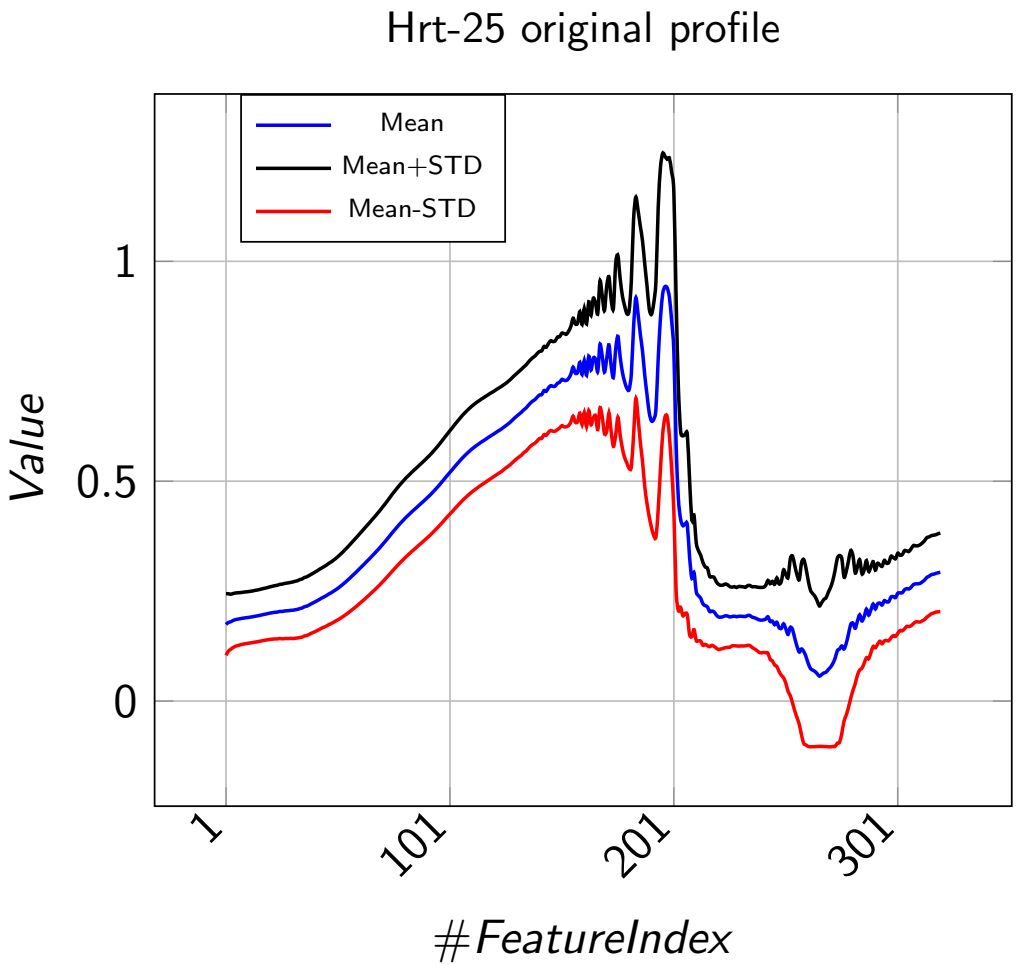
Figure: The comparison of original samples of Sap (black) and the GAN generated non-real (red) samples (for training on 50% samples) (a), generated after 1000 epochs and (b) after 10000 epochs

SpectralGAN Visual Analysis

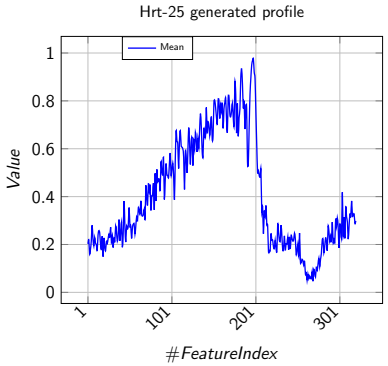


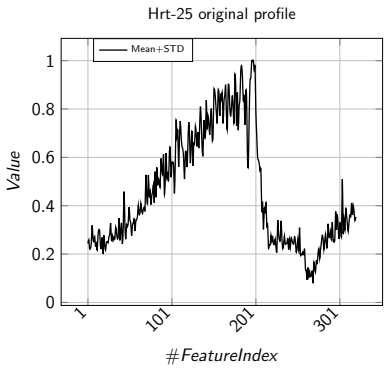


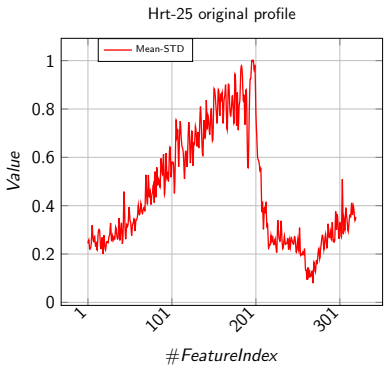


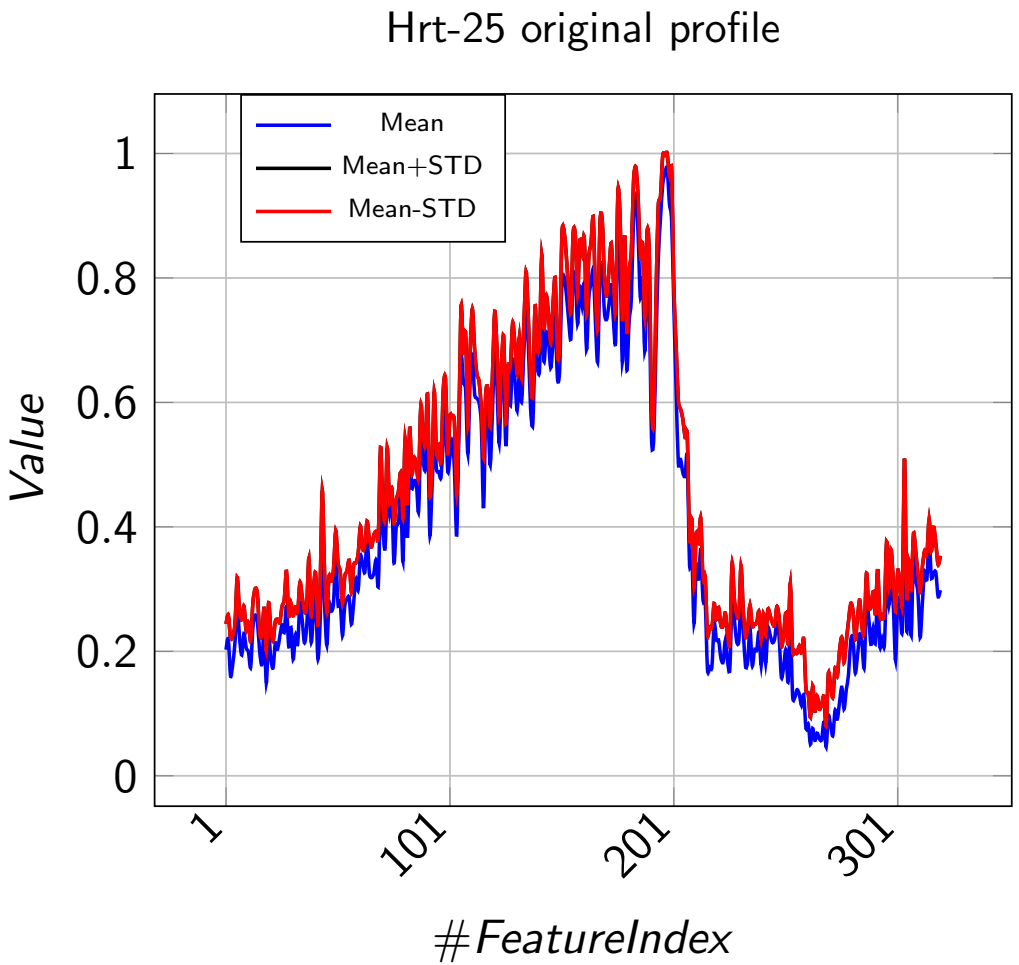


SpectralGAN Visual Analysis



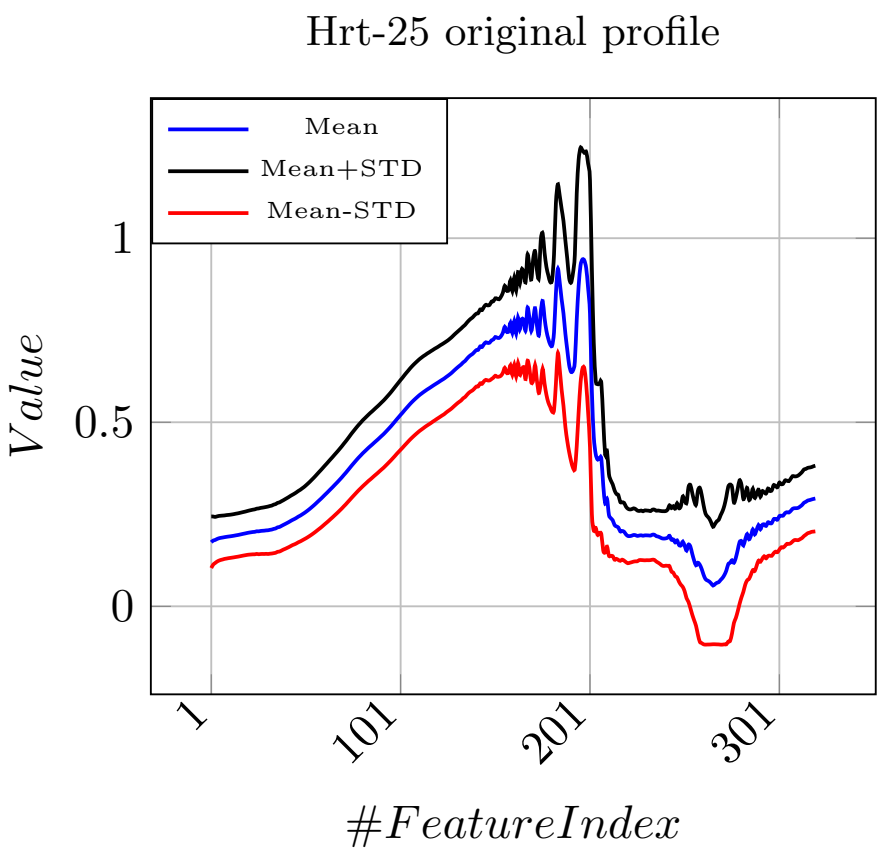


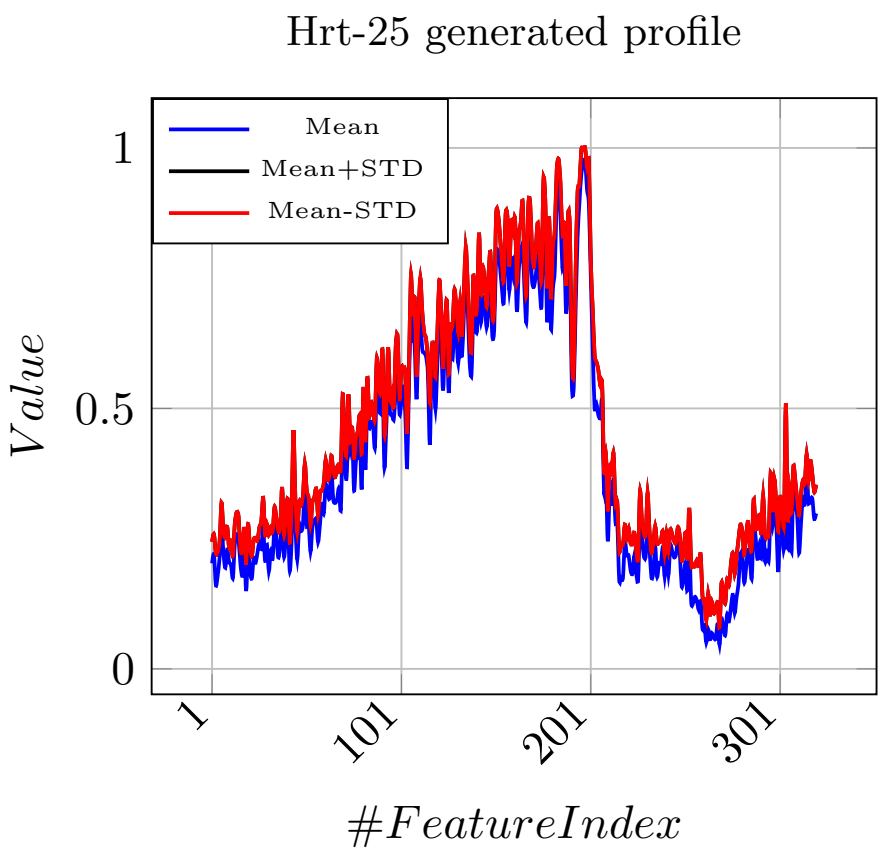




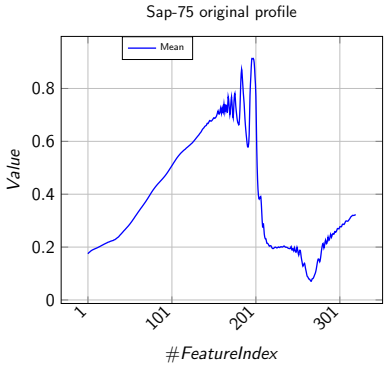
SpectralGAN Visual Analysis

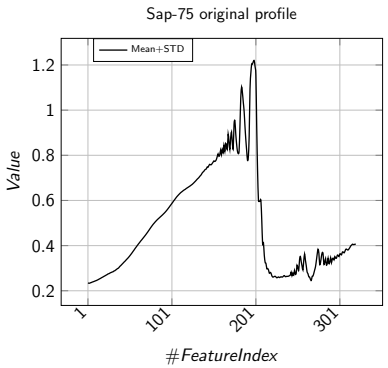
- **Generation of synthetic samples (Hrt_25_profile comparison)**

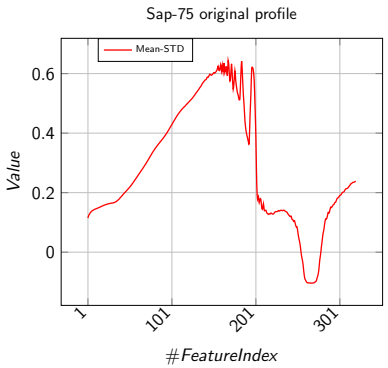


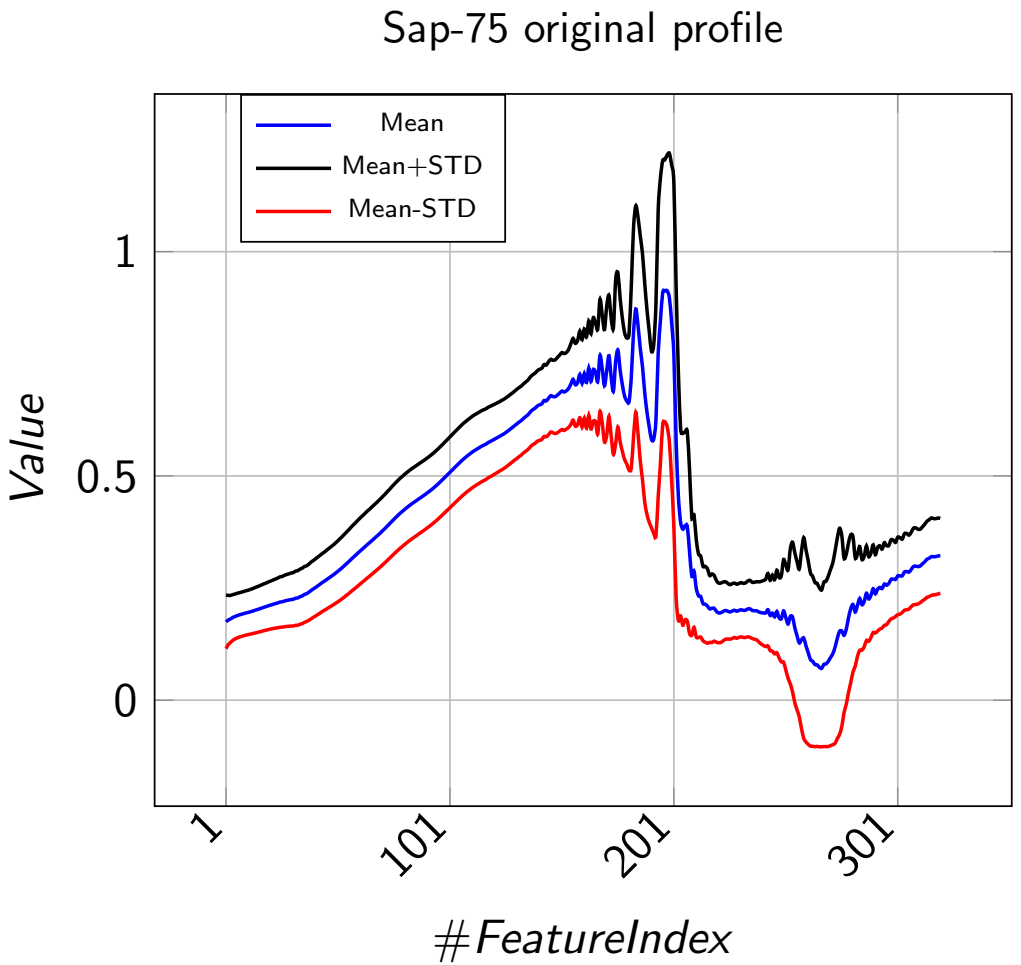


SpectralGAN Profile Comparison

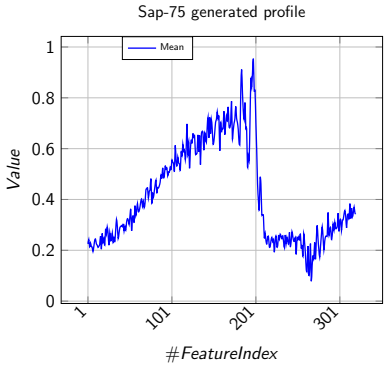


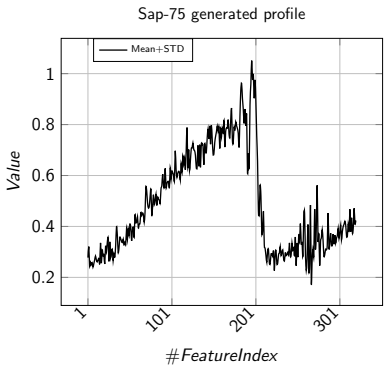


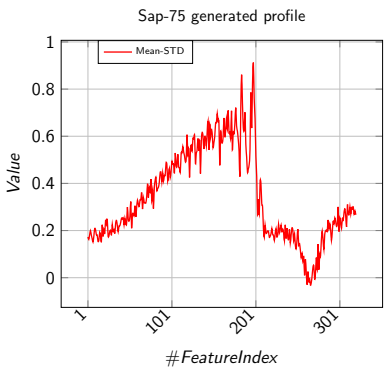


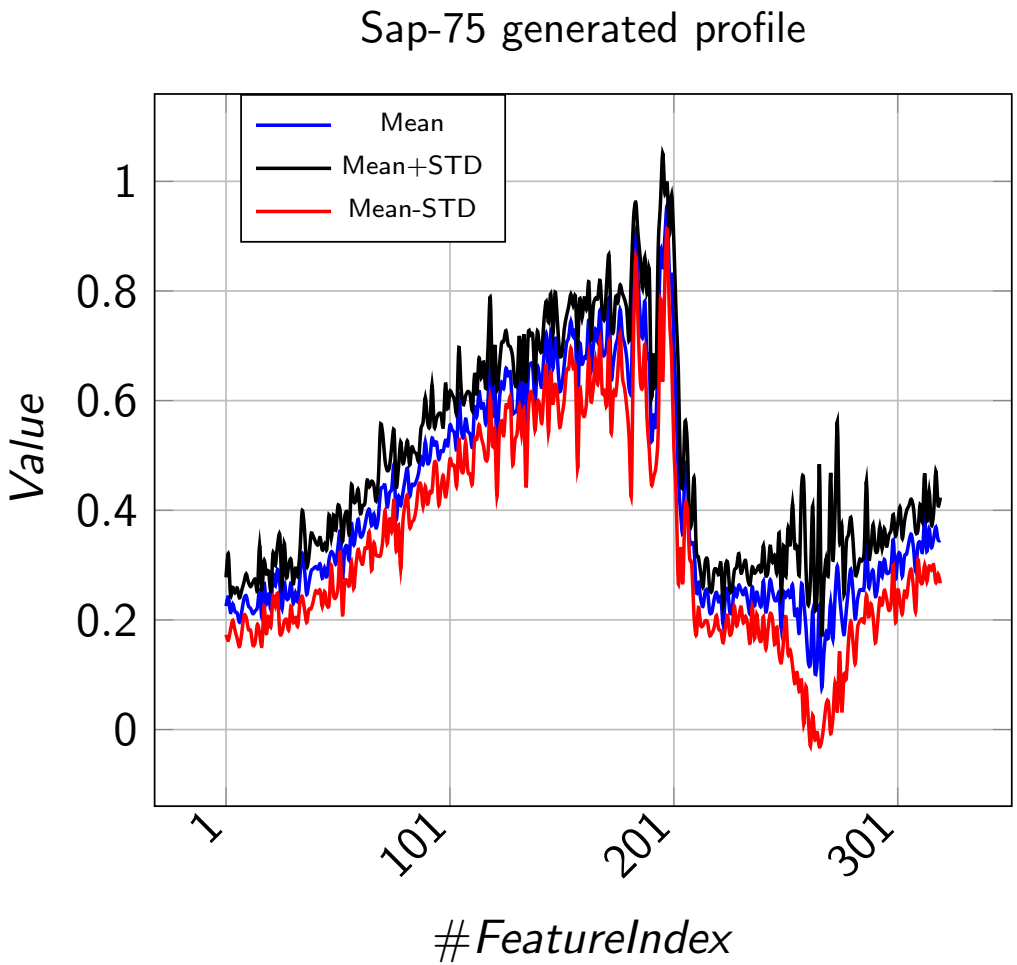


SpectralGAN Profile Comparison



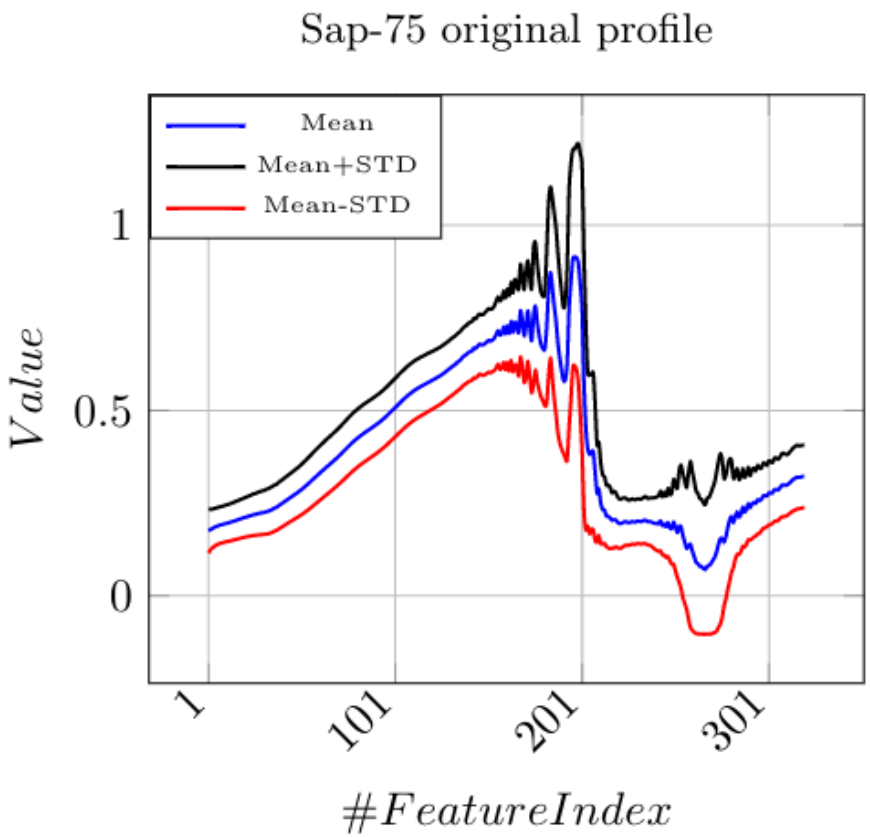


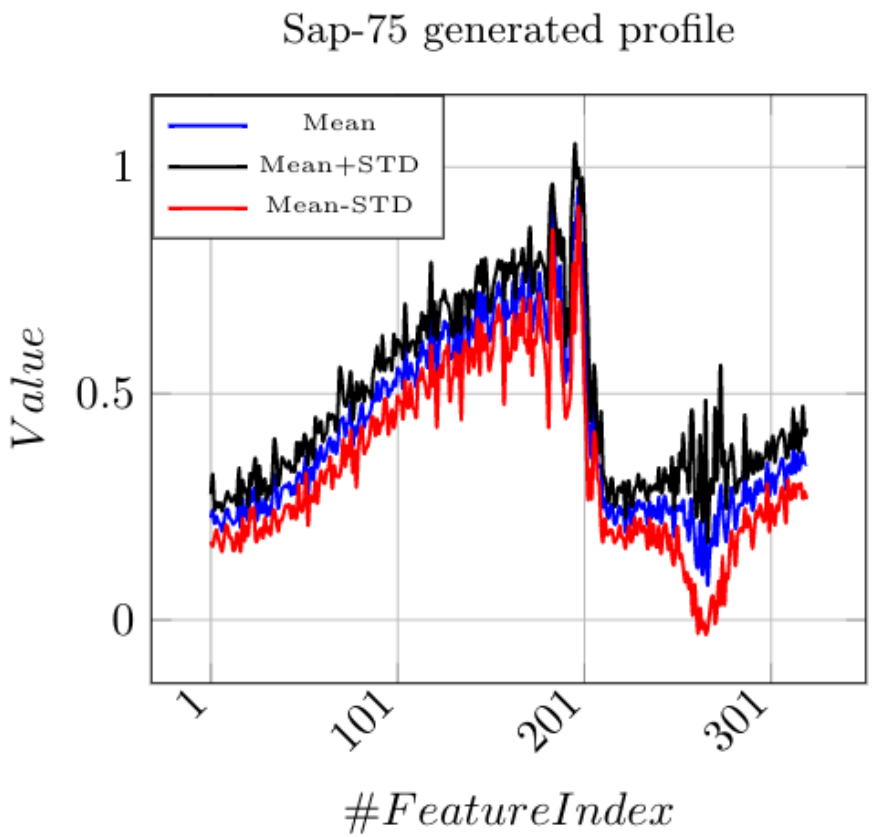




SpectralGAN Profile Comparison

- **Generation of synthetic samples (Sap_75_profile comparison)**





SpectralGAN Results

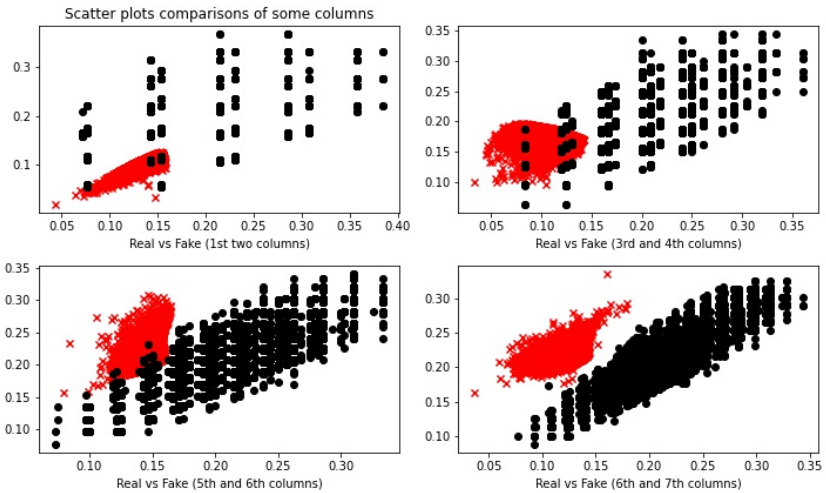
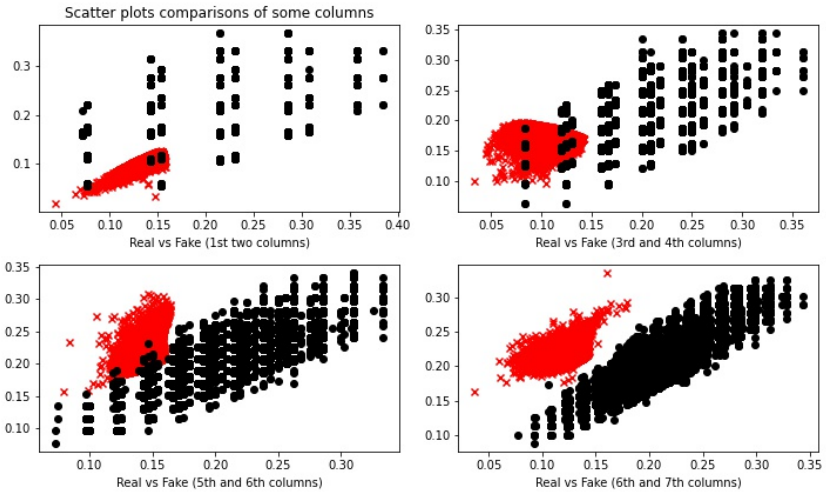


Figure: Comparison of real and generated features on real and augmented dataset (generated after 1000 and 10000 epochs) for Heartwood (25%setting

SpectralGAN Results



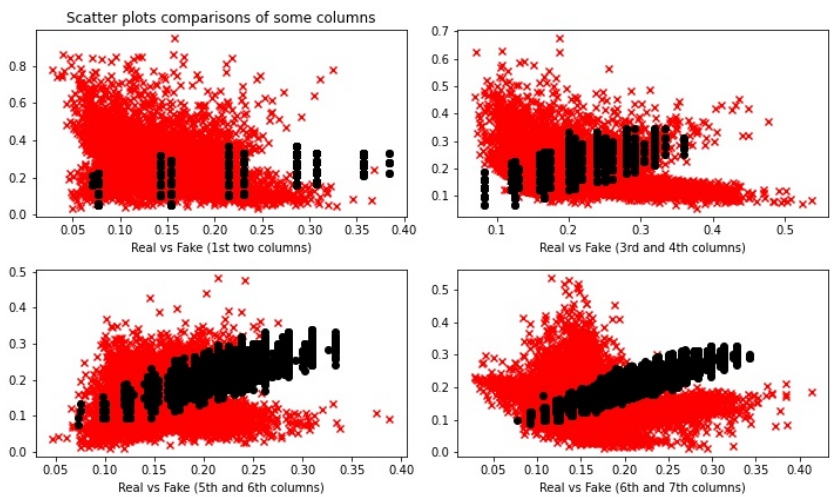


Figure: Comparison of real and generated features on real and augmented dataset (generated after 1000 and 10000 epochs) for Heartwood (25%setting

SpectralGAN Results

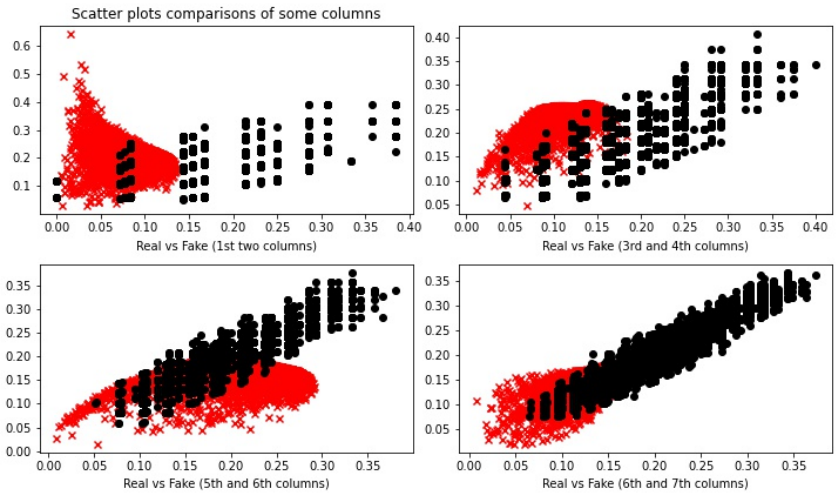
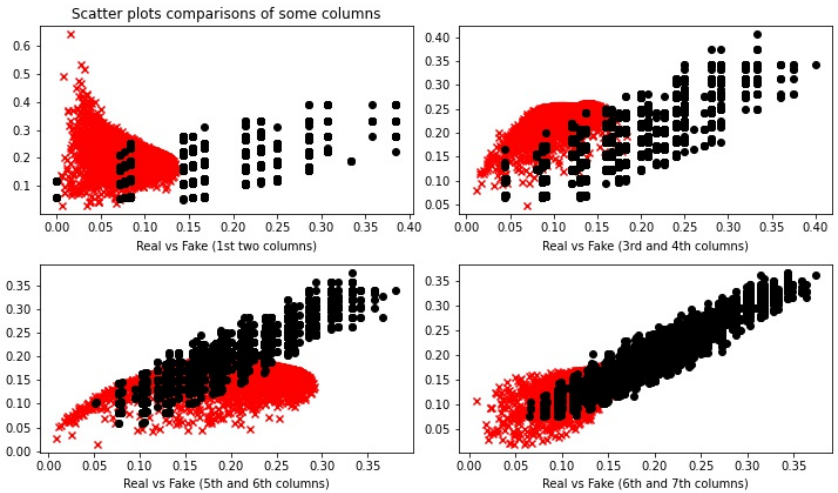


Figure: Comparison of real and generated features on real and augmented dataset (generated after 1000 and 10000 epochs) for sapwood (25% setting

SpectralGAN Results



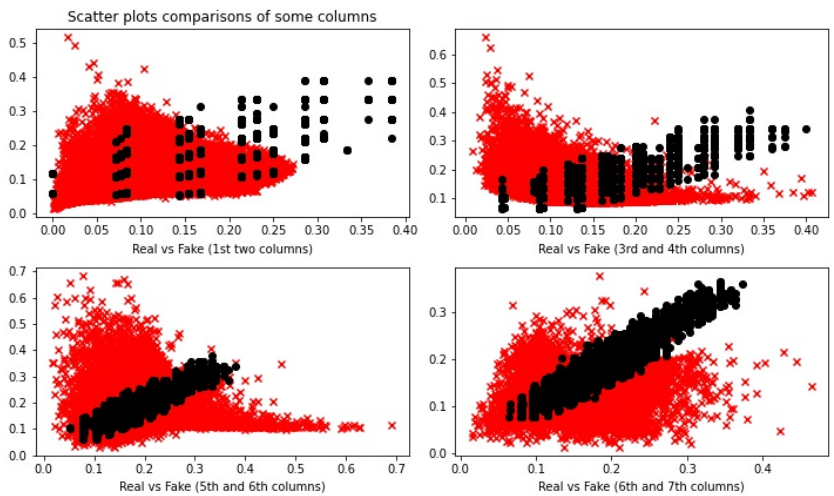
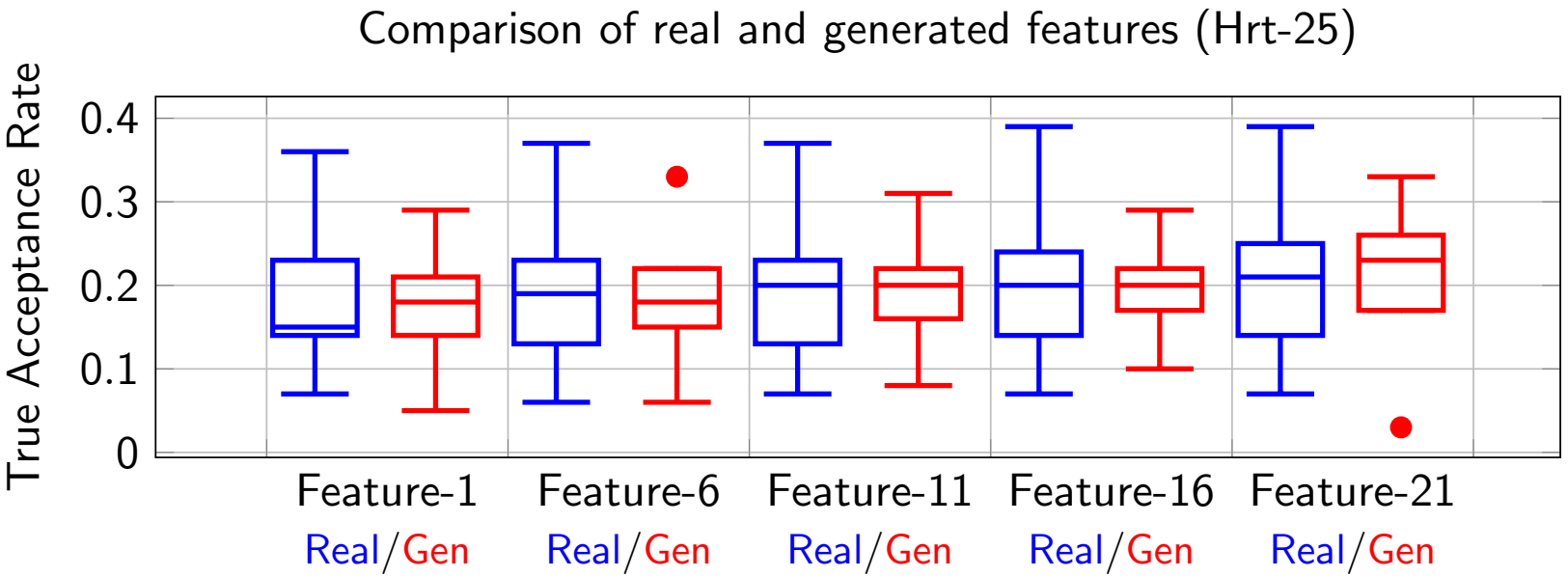
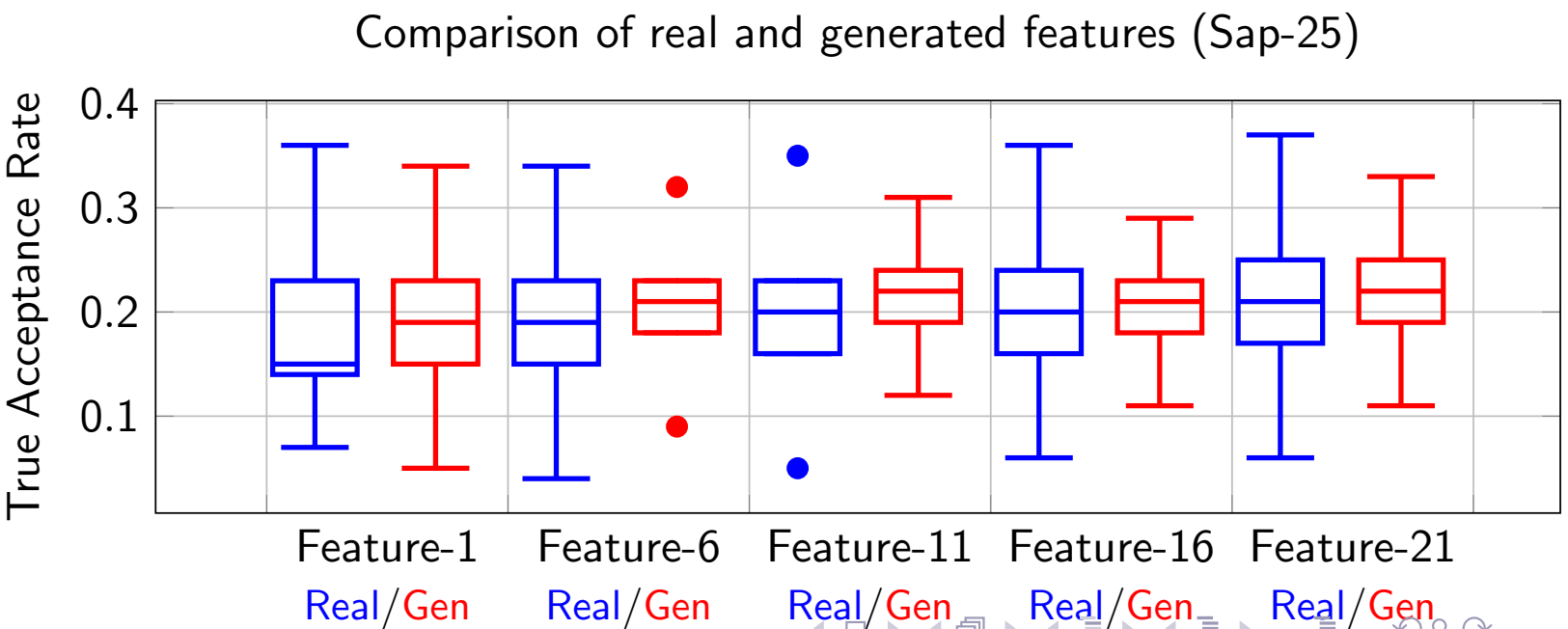


Figure: Comparison of real and generated features on real and augmented dataset (generated after 1000 and 10000 epochs) for sapwood (25% setting

SpectralGAN Features Comparison





Results

- **Success Metric**

True Positives (TPs): 1	False Positives (FPs): 1
False Negatives (FNs): 8	True Negatives (TNs): 90

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{1}{1+8} = 0.11$$

Figure: Recall

- **Appropriate when minimizing false negatives is the focus**.

True Positives (TPs): 1	False Positives (FPs): 1
False Negatives (FNs): 8	True Negatives (TNs): 90

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{1}{1+1} = 0.5$$

- **Appropriate when minimizing false positives is the focus**.

SpectralGAN Evaluation Results

- **Real Vs Real+Gen (25% or 33 cubes)**

	Heartwood			*Sapwood*		
	Recall	Precision	F_Score	Recall	Precision	F_Score
Real_25%	89.61	94.9	92.68	95.18	90.16	92.60
Real+Gen_25%	93.28	93.26	93.27	93.26	93.28	93.27

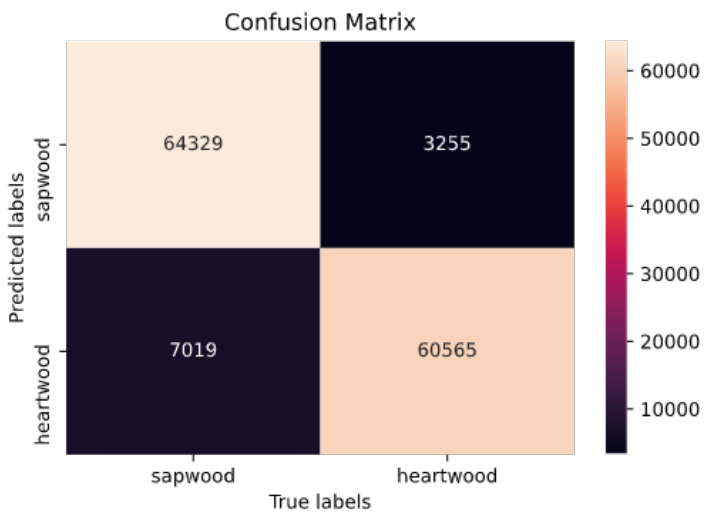


Figure: on training on real samples

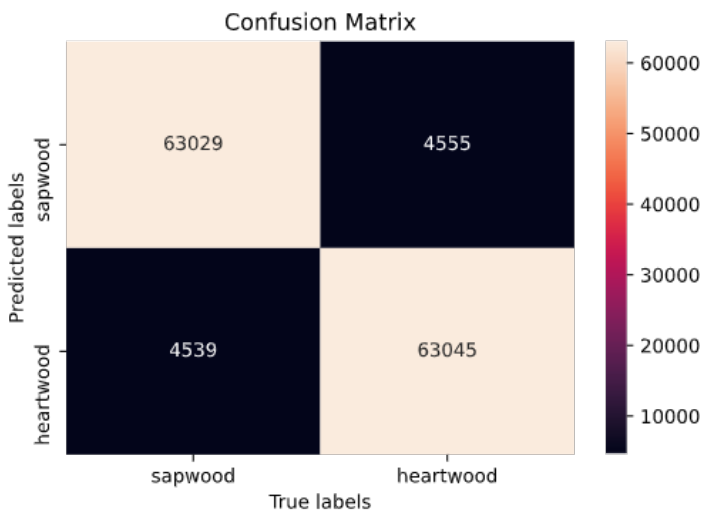


Figure: on training on real + gen samples

SpectralGAN Evaluation Results

- **Real Vs Real+Gen (50% or 66 cubes)**

	Heartwood			*Sapwood*		
	Recall	Precision	F_Score	Recall	Precision	F_Score
Real_50%	57.51	98.41	72.60	99.07	69.98	82.07
Real+Gen_50%	86.76	96.45	91.35	96.81	87.97	92.18

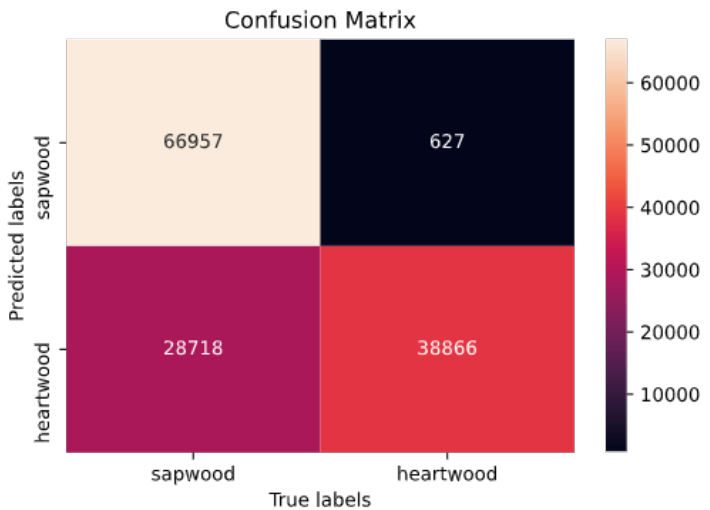


Figure: on training on real samples

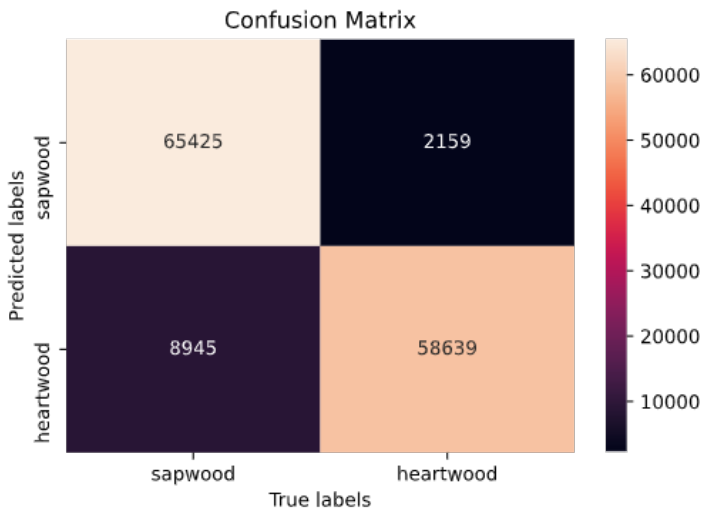


Figure: on training on real + gen samples

SpectralGAN Evaluation Results

- **Real Vs Real+Gen (75% or 99 cubes)**

	Heartwood			*Sapwood*		
	Recall	Precision	F_Score	Recall	Precision	F_Score
Real_75%	90.22	90.16	90.19	90.15	90.21	90.18
Real+Gen_75%	96.16	77.99	86.13	72.87	94.99	82.47

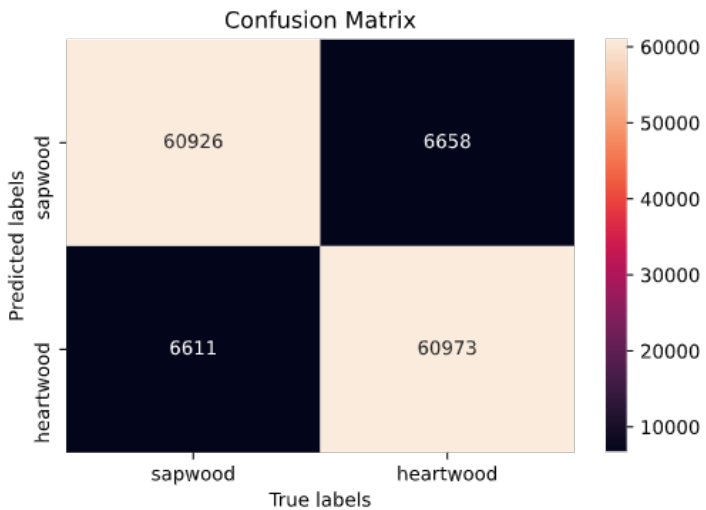


Figure: on training on real samples

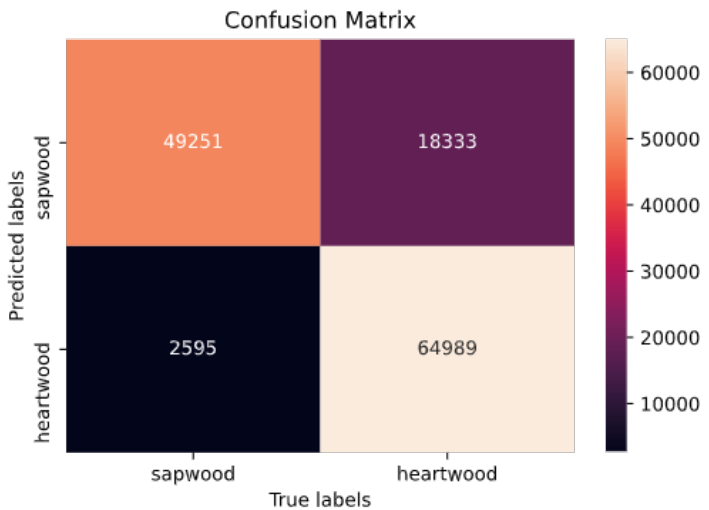


Figure: on training on real + gen samples

SpectralGAN Evaluation Results

- **Real (all cubes)**

	Heartwood			*Sapwood*		
	Recall	Precision	F_Score	Recall	Precision	F_Score
Real_100%	94.33	93.50	93.91	93.45	94.28	93.86

Conclusions

- **We have discussed in details the importance of data augmentation**
- **Why we should use GANs for data augmentation**
- **We empirically show the improvement of results when the model is trained on increased number of samples**
- **Results are preliminary: we have been exploring multiple strategies, network architectures, optimization parameters, for reaching the maximum improvement.**

Thank You!

SpectralGAN Results - with newer generator and training settings (Real Vs Real+Generated)

- **Training on Real and Real + Generated Samples (25%)**

	Heartwood					
	Real (%)			*Generated (%)*		
	Recall	Precision	F_Score	Recall	Precision	F_Score
25_1	93	94	94	**93**	**95**	**94**
25_2	81	81	81	**76**	**78**	**77**
25_3	78	76	77	**91**	**77**	**83**
25_4	93	89	91	**93**	**85**	**89**
25_5	90	89	90	**91**	**90**	**90**
25_6	77	79	78	**71**	**79**	**75**
25_7	92	90	91	**94**	**87**	**90**
Avg	86	85	86	**87**	**84**	**85**

SpectralGAN Results - Real Vs Real+Generated

- **Training on Real and Real + Generated Samples (25%)**

	Sapwood					
	Real (%)			*Generated (%)*		
	Recall	Precision	F_Score	Recall	Precision	F_Score
25_1	95	93	94	**95**	**93**	**94**
25_2	81	81	81	**78**	**77**	**77**
25_3	76	78	77	**91**	**77**	**83**
25_4	88	93	91	**83**	**93**	**88**
25_5	89	90	90	**90**	**90**	**90**
25_6	80	78	79	**74**	**81**	**77**
25_7	90	92	91	**86**	**93**	**89**
Avg	85	86	86	**85**	**86**	**85**

SpectralGAN Results - Real Vs Real+Generated

- **Training on Real and Real + Generated Samples (50%)**

	Heartwood					
	Real (%)			*Generated (%)*		
	Recall	Precision	F_Score	Recall	Precision	F_Score
50_1	94	87	90	**94**	**87**	**90**
50_2	91	91	91	**92**	**88**	**91**
50_3	96	91	93	**88**	**93**	**91**
Avg	93	89	91	**91**	**89**	**90**

SpectralGAN Results - Real Vs Real+Generated

- **Training on Real and Real + Generated Samples (50%)**

	Sapwood					
	Real (%)			*Generated (%)*		
	Recall	Precision	F_Score	Recall	Precision	F_Score
50_1	84	89	87	**85**	**93**	**89**
50_2	90	91	91	**88**	**92**	**90**
50_3	90	95	93	**94**	**89**	**91**
Avg	88	91	90	**89**	**91**	**89**

SpectralGAN Results - Real Vs Real+Generated

- **Training on Real and Real + Generated Samples (75%)**

	Heartwood					
	Real (%)			*Generated (%)*		
	Recall	Precision	F_Score	Recall	Precision	F_Score
75_1	94	89	91	**89**	**95**	**92**
75_2	84	92	87	**87**	**92**	**90**
Avg	89	90	89	**88**	**93**	**91**

SpectralGAN Results - Real Vs Real+Generated

- **Training on Real and Real + Generated Samples (75%)**

	Sapwood					
	Real (%)			*Generated (%)*		
	Recall	Precision	F_Score	Recall	Precision	F_Score
75_1	88	93	91	**95**	**90**	**92**
75_2	92	85	88	**93**	**88**	**90**
Avg	90	89	89	**94**	**89**	**91**