Design and Implementation of a Computing Scenario Forecasting System Based on Generative Adversarial Networks

Student: Jaime Pérez Sánchez

Advisor: Patricia Arroba García

Table of Contents

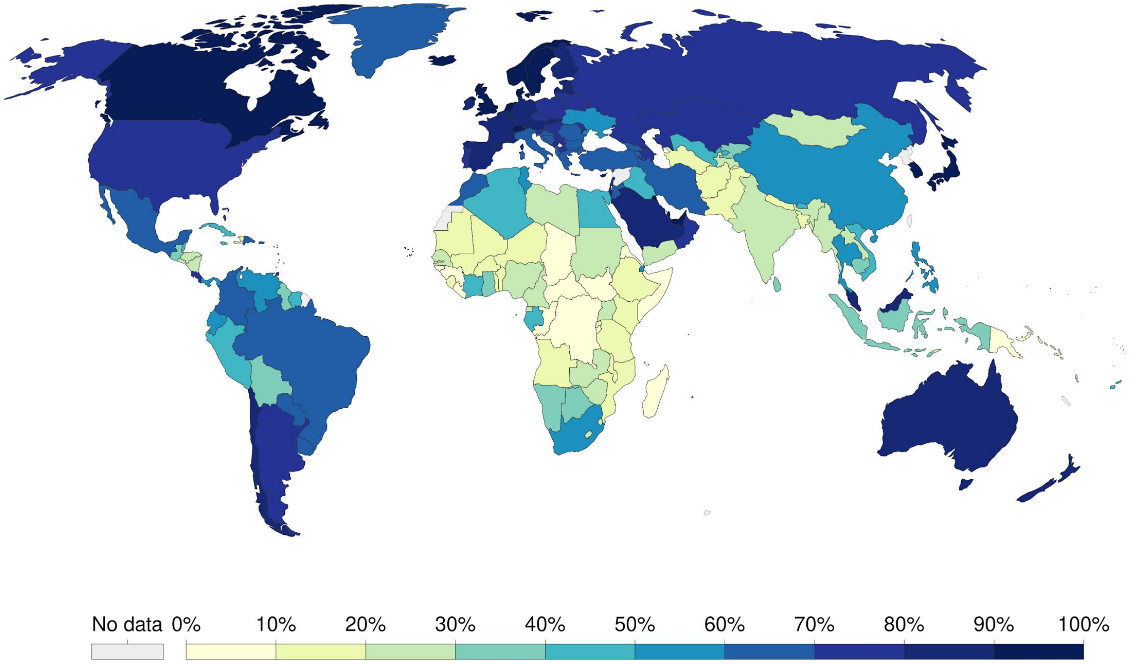
Digital Economy & Society

World's Population Using the Internet	
2018	51%
2023	66%

European Citizens Using Internet Daily	
2014	48%
2018	73%

Daily Hours Spent on Digital Media by U.S. Citizens	
2013	<5h
2018	>6h

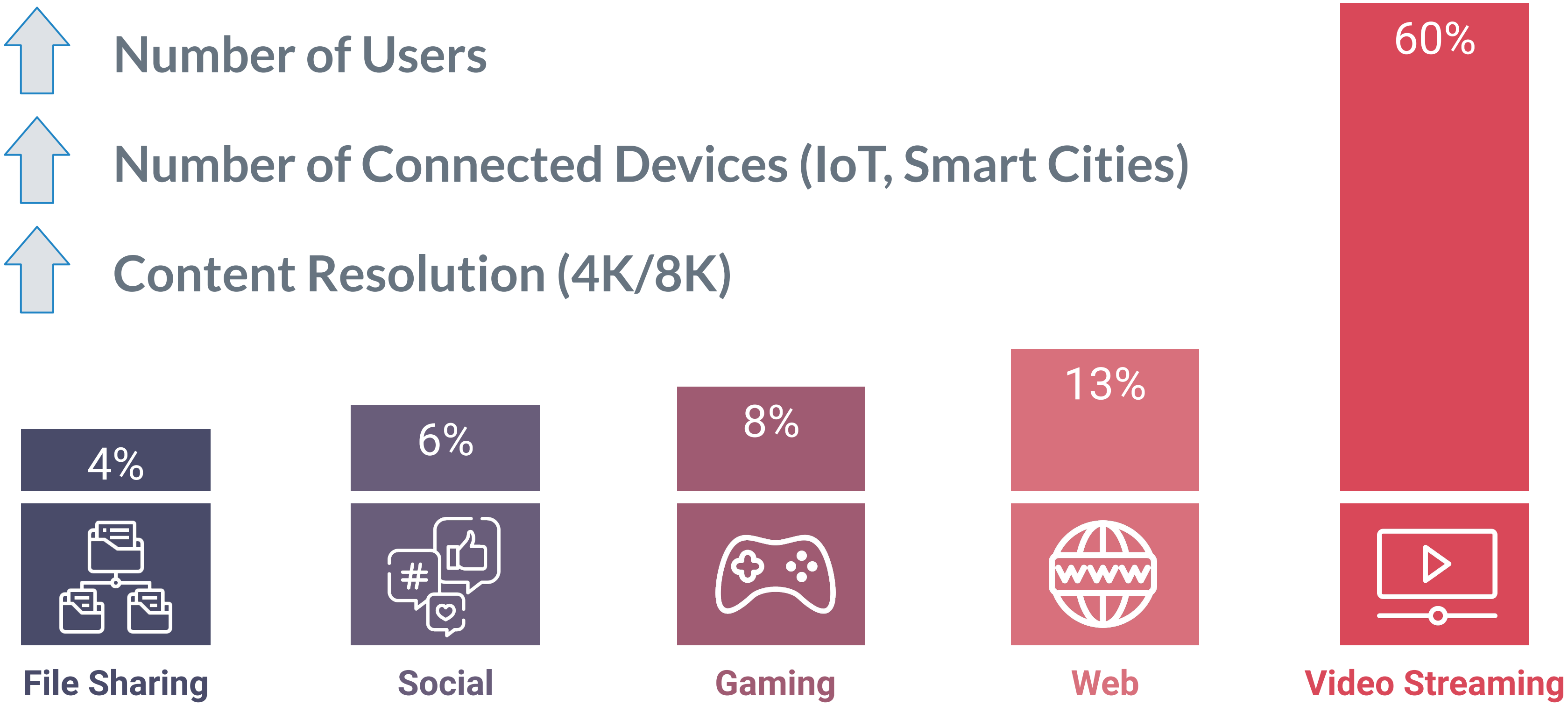
Share of population using the Internet (2017)



Source: World Bank Data (2017)

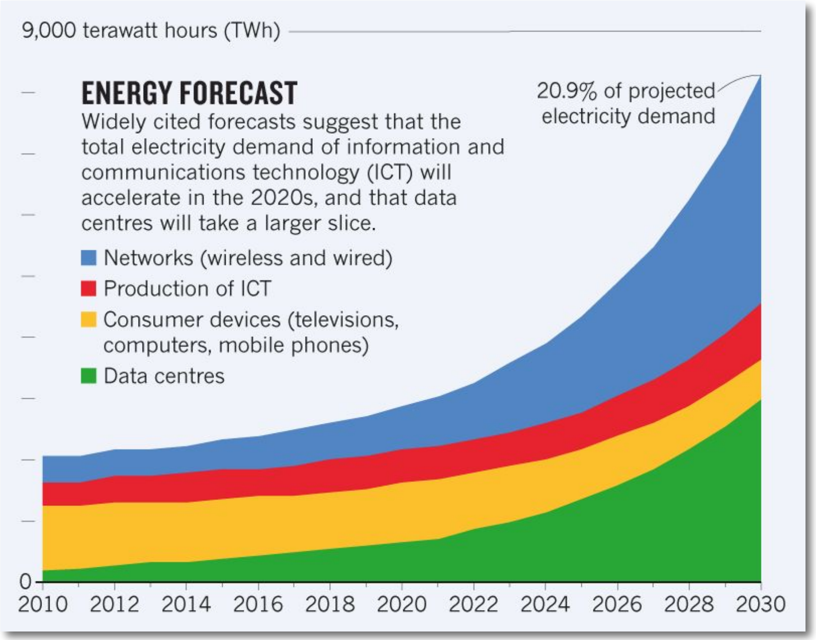
Internet Traffic Share

- **Number of Users**
- **Number of Connected Devices (IoT, Smart Cities)**
- **Content Resolution (4K/8K)**



Cloud Computing Paradigm

- Vast majority of new technologies base their operations on **Cloud**
- 2010 → 2018: Compute instances increased by **550%** and IP traffic by **1,000%**
- 2018: Cloud Global Energy Use of **205 TWh** (**1%** of Global Energy Consumption)



Cloud Computing Paradigm

- Progress on Data Center **energy efficiency**
  - Global Power Usage Effectiveness (PUE) of 1.67 (ideal = 1)
  - Energy use per computation has dropped ~400%
- **How much longer can we improve energy efficiency?**
  - **Cooling energy** expenses represent, on average, up to **40%** of the total bill
  - In the next **3 or 4 years** the number of compute instances will **double** again

Cloud Computing Paradigm

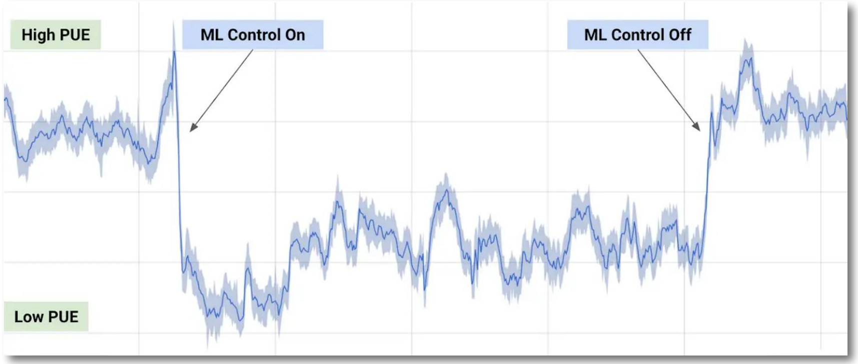
- Progress on Data Center **energy efficiency**
  - Global Power Usage Effectiveness (PUE) of 1.67 (ideal = 1)
  - Energy use per computation has dropped ~400%

- **How much longer can we improve energy efficiency?**
  - **Cooling energy** expenses represent, on average, up to **40%** of the total bill
  - In the next **3 or 4 years** the number of compute instances will **double** again

We urgently need better optimization in Data Centers!

Data-Driven Optimization

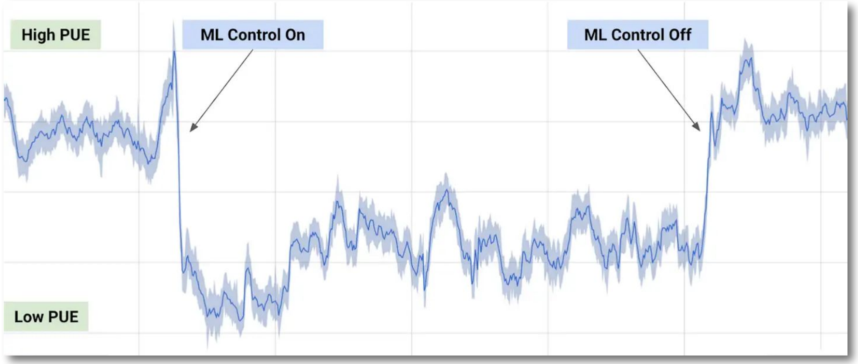
- Outstanding results of **Machine Learning** and **Deep Learning**. Why?
  - Massive amounts of data
  - Increased computing power (GPUs)
  - Transfer learning from expert pre-trained models
- ML and DL optimization in Data Centers
  - Systems incredibly challenging to optimize
  - Google achieved 40% of cooling energy saving (15% reduction of PUE)



Source: DeepMind

Data-Driven Optimization

- Outstanding results of **Machine Learning** and **Deep Learning**. Why?
  - Massive amounts of data
  - Increased computing power (GPUs)
  - Transfer learning from expert pre-trained models
- ML and DL optimization in Data Centers
  - Systems incredibly challenging to optimize
  - Google achieved 40% of cooling energy saving (15% reduction of PUE)

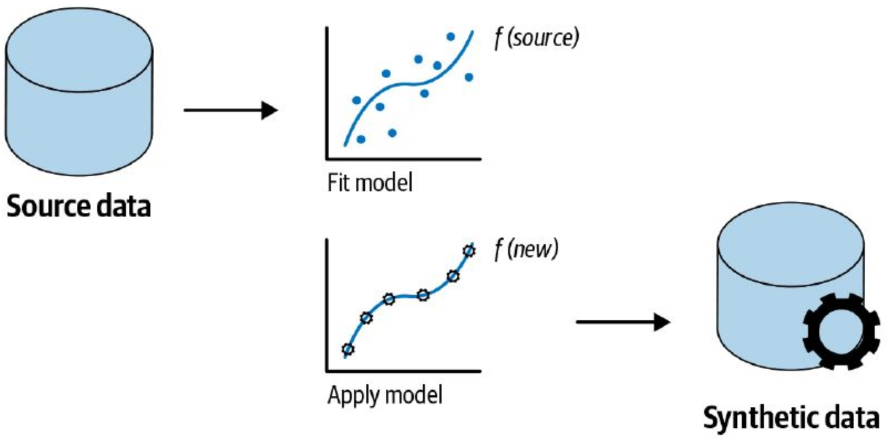


Source: DeepMind

What can we do if we do not have enough data?

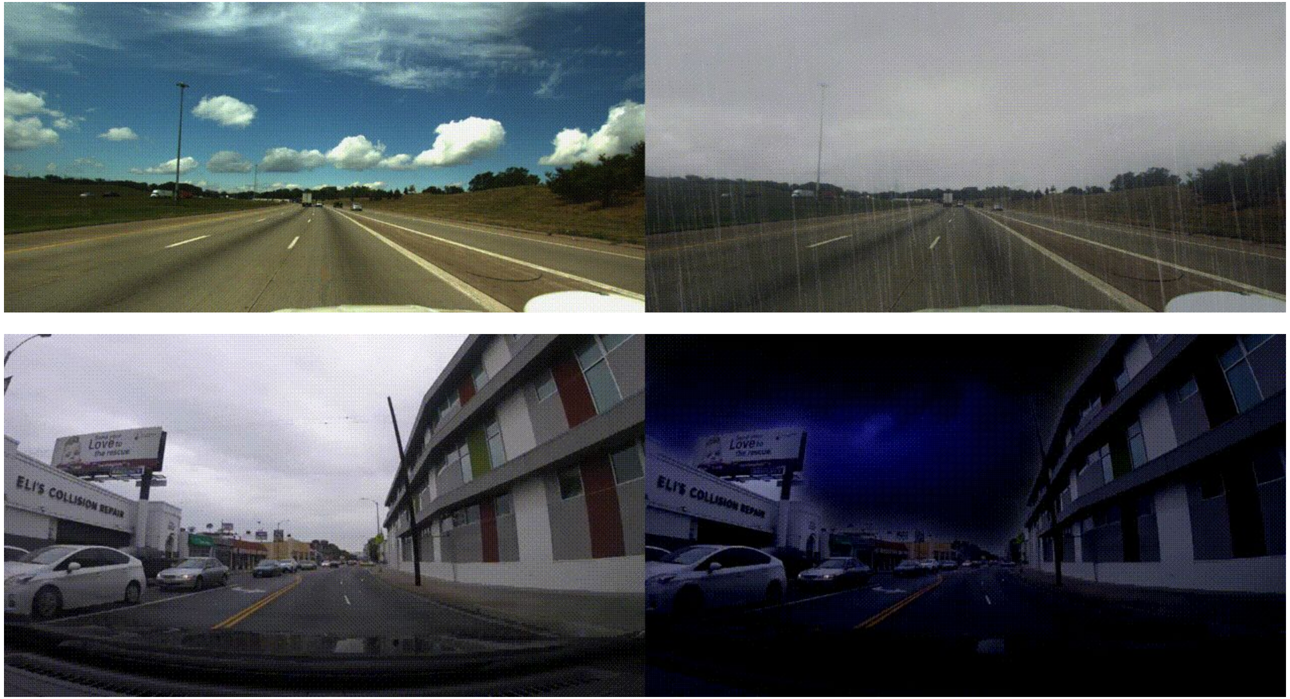
Synthetic Data Generation

- Adoption of synthetic data
  - Companies: Google, IBM, NVIDIA
  - Agencies: US Census Bureau
- More **efficient access to data**
- Enable better analytics when gather real data is too **costly, dangerous, or unethical**



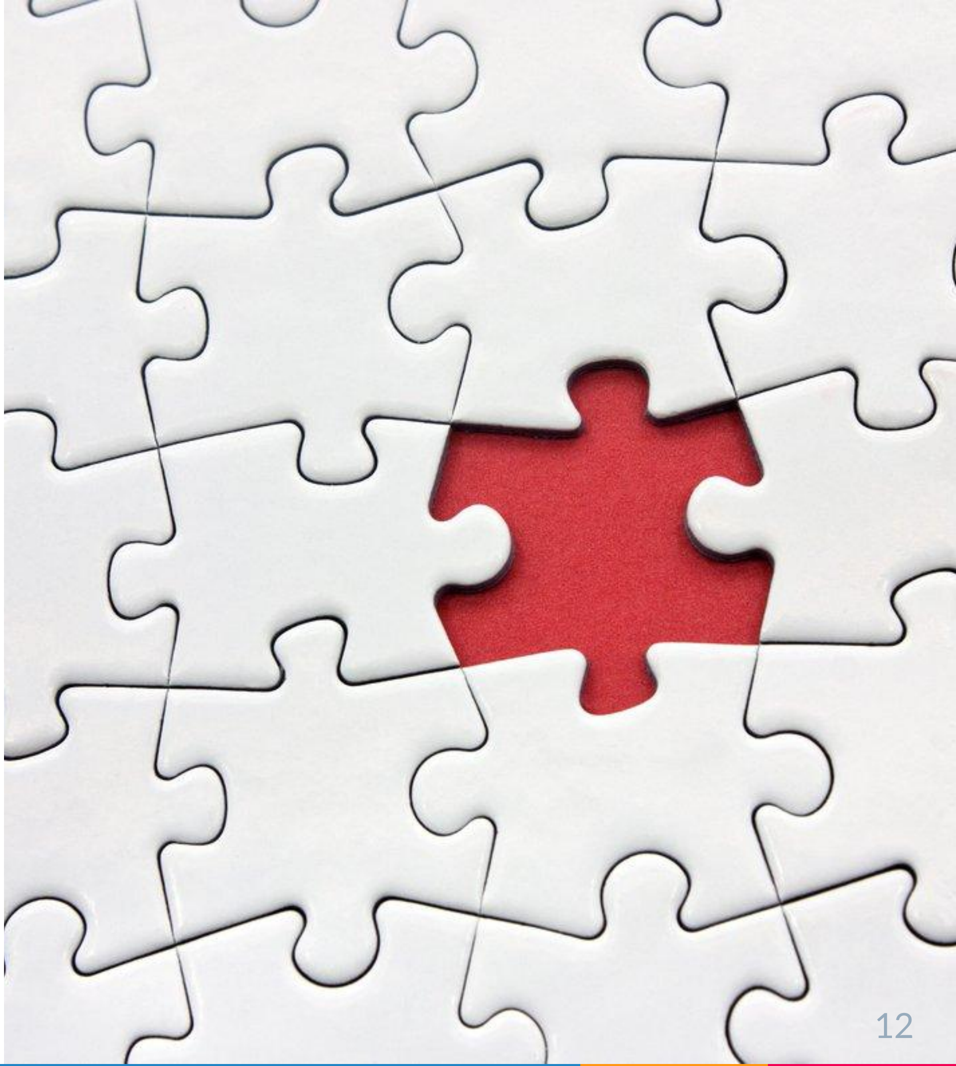
Generative Adversarial Networks (GANs)

Real Image | **Synthetic Image**



Source: NVIDIA

Table of Contents



State Of the Art: Synthetic Time-Series Data

Generation Based on Statistical Methods:

- ▷ Naive: Gaussian Noise, Rotation, Scaling, Warping...
- ▷ AutoRegressive Models: ARIMA, ARMA...
- ▷ Markov Models: Hidden Markov Models
- ▷ Bayesian Models: Dynamic Bayesian Networks, Bayesian Structural Time-Series...

PROS

- ❑ Interpretability
- ❑ Can be applied in small datasets

CONS

- ❑ Require expertise knowledge
- ❑ Poor work on multivariate data with non-linear relationships
- ❑ Do not always improve with more data

State Of the Art: Synthetic Time-Series Data

Generation Based on GANs:

PROS

- ❑ Outstanding empirical results
- ❑ Can handle multi-variable data with non-linear relationships
- ❑ Flexible tune generation
- ❑ Can be applied in conjunction with traditional data augmentation methods

CONS

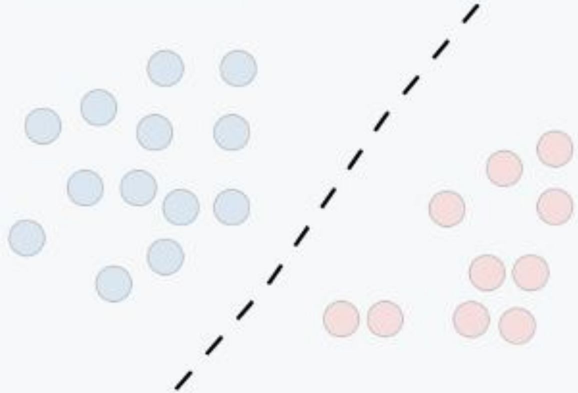
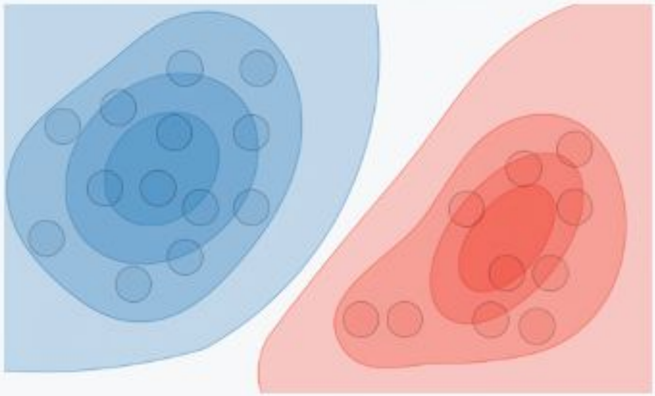
- ❑ Unstable training (partially solved)
- ❑ Diversity of the generated data can be limited and biased by the available data
- ❑ It needs relatively large amounts of data for efficient training
- ❑ It does not compute an explicit density estimation (i.e., black-box model)

State Of the Art: Synthetic Time-Series Data using GANs

	Data Augmentation	Single-Variable Scenario Generation					*Ours*
	[42]	[49]	[50]	[51]	[53]	[54]	
Realistic Data Generation	✓	✓	✓	✓	✓	✓	✓
Scenario Generation	✗	✓	✓	✓	✓	✓	✓
Generate Data from Particular Time Instants	✗	✓	✗	✗	✗	✗	✓
Generate on-demand anomalous situations	✗	✗	✗	✗	✗	✗	✓
Multi-variable Generation	✓	✗	✗	✗	✗	✗	✓
Introduce Categorical Variables	✓	✗	✗	✗	✓	✗	✓
Introduce Spatial Information	**Not Tested**	✗	✓	✗	✓	✗	✓
Disentangled Latent Space	✗	✗	✗	✗	✗	✓	**Future Work**

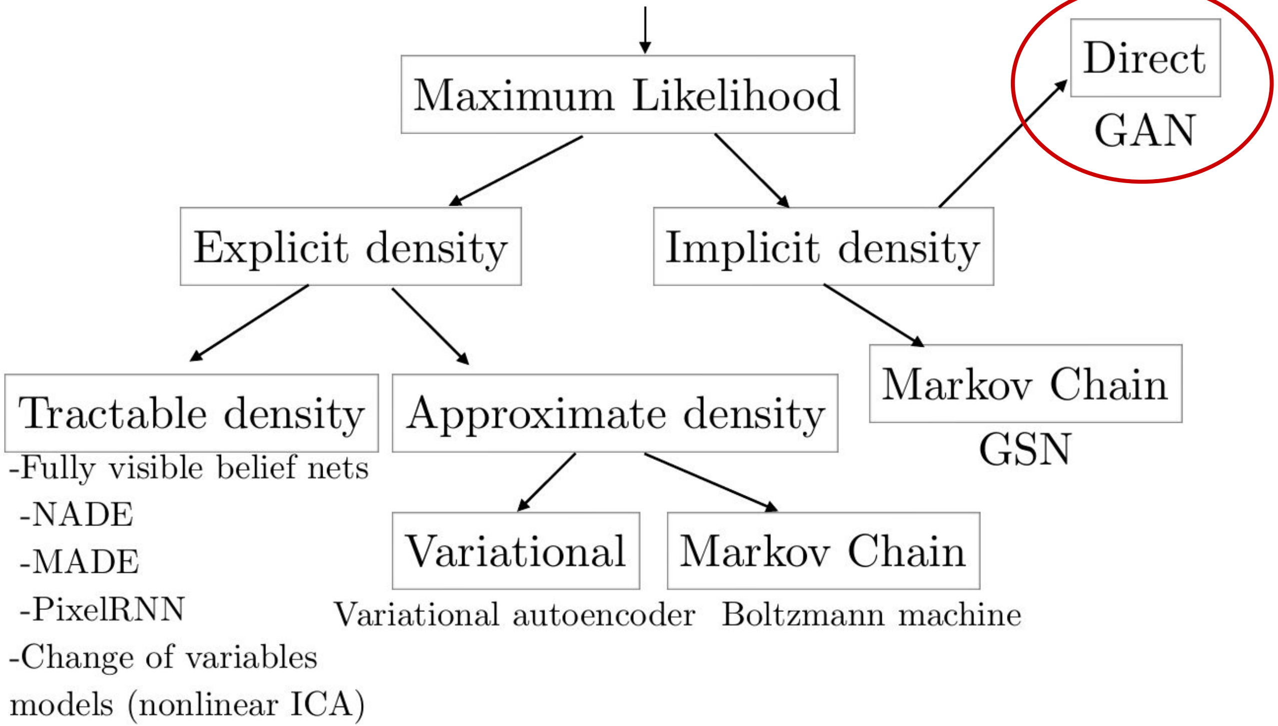
Table of Contents

Discriminative Models vs. Generative Models

	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

Generative Models Taxonomy

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

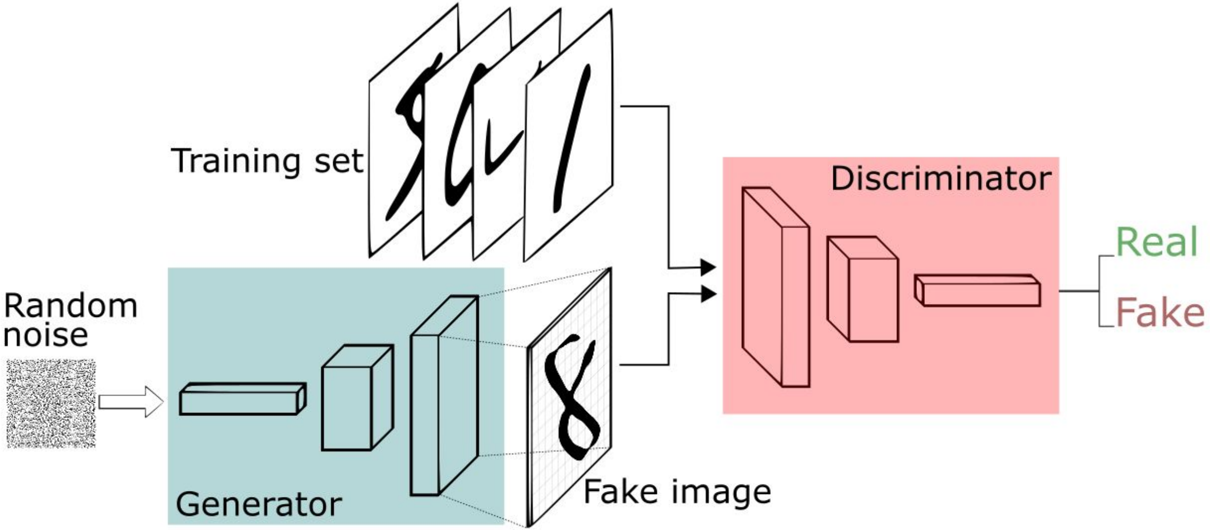


Generative Adversarial Networks (GANs)

Training GANs: Two-Player Game

- Minimax Objective Function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$



Generative Adversarial Networks (GANs)

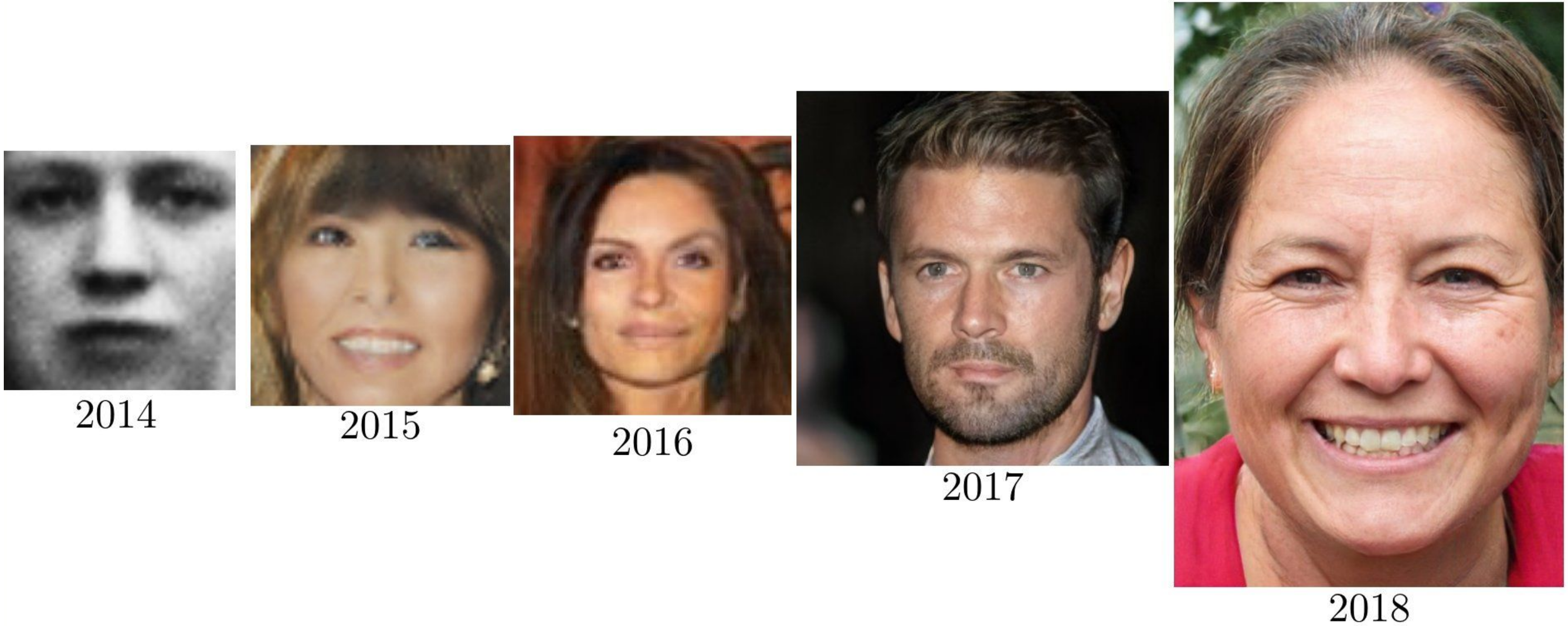
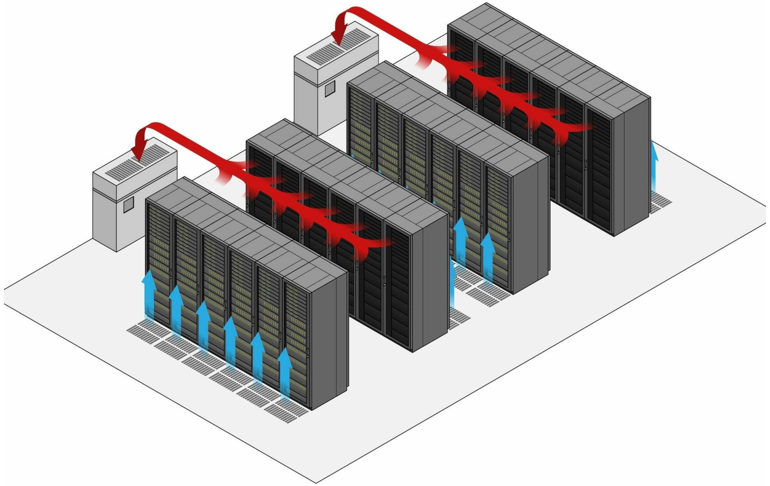


Table of Contents

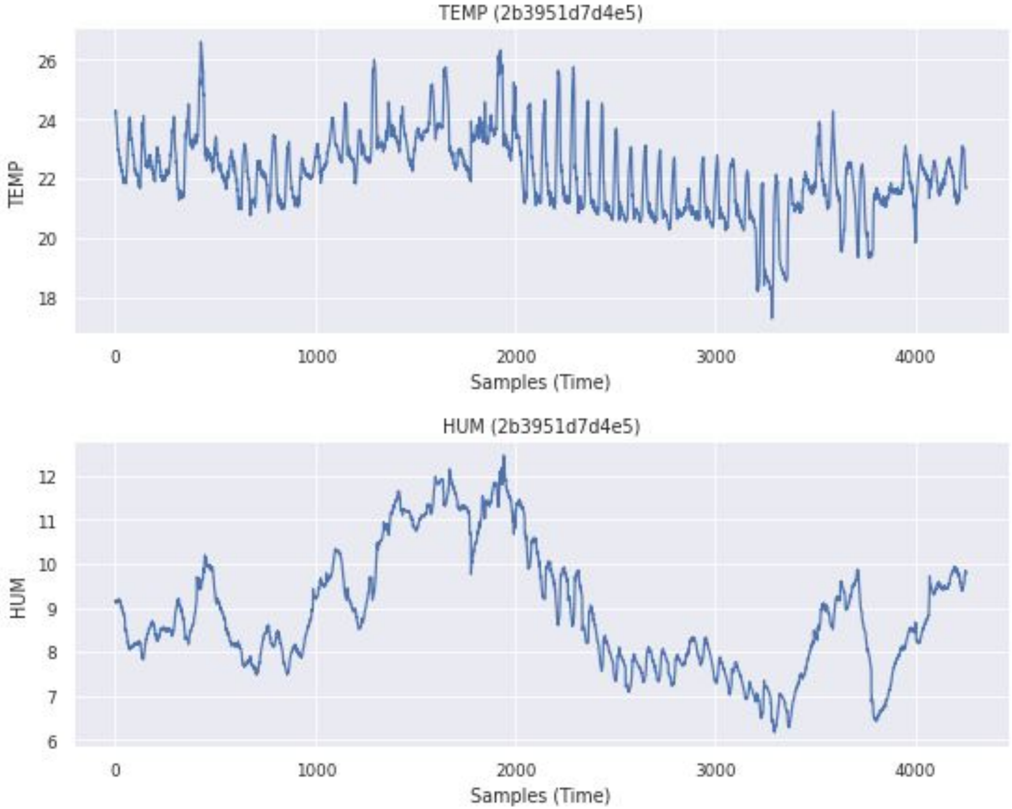
Sensor Data in Data Center Facility

Real data gathered from an operating Data Center:

- Facility: *Adam Data Centers* [https://adam.es/data-center/]
- Sensors: *TycheTools* [https://www.tychetools.com]
    - 35 sensors → Temperature and Relative Humidity collected every 10 minutes

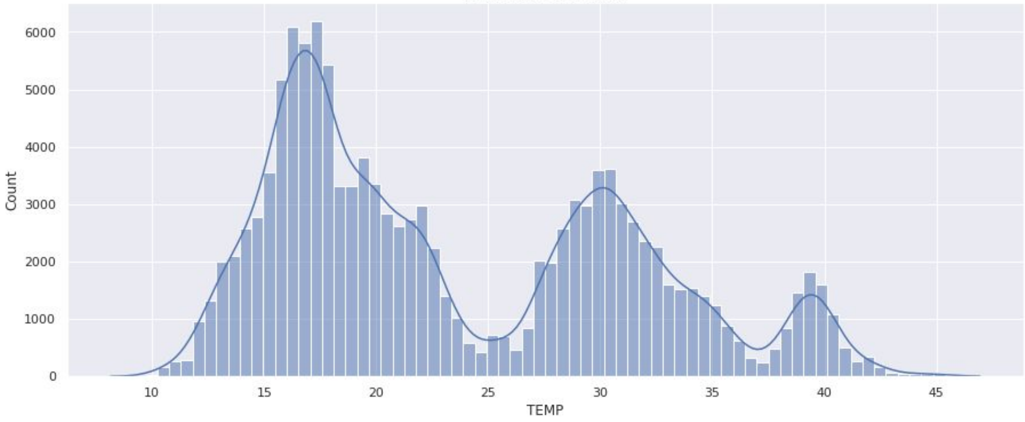


Example of Data Gathered by One Sensor

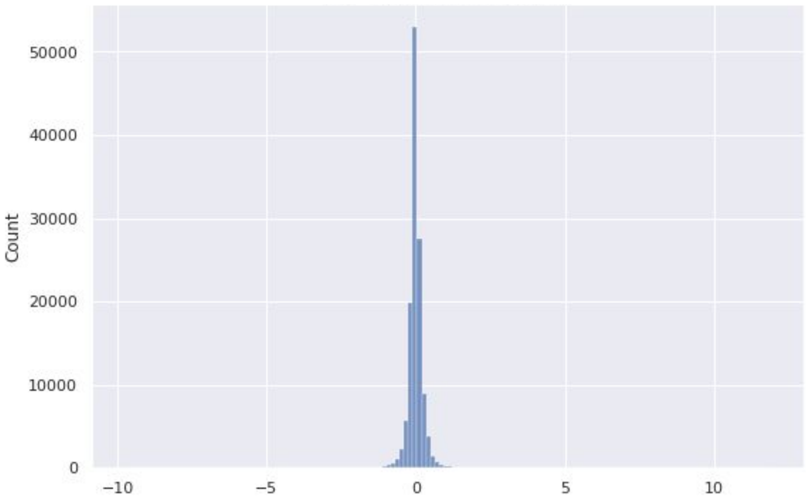


Exploratory Data Analysis: Temperature

Temperature Histogram



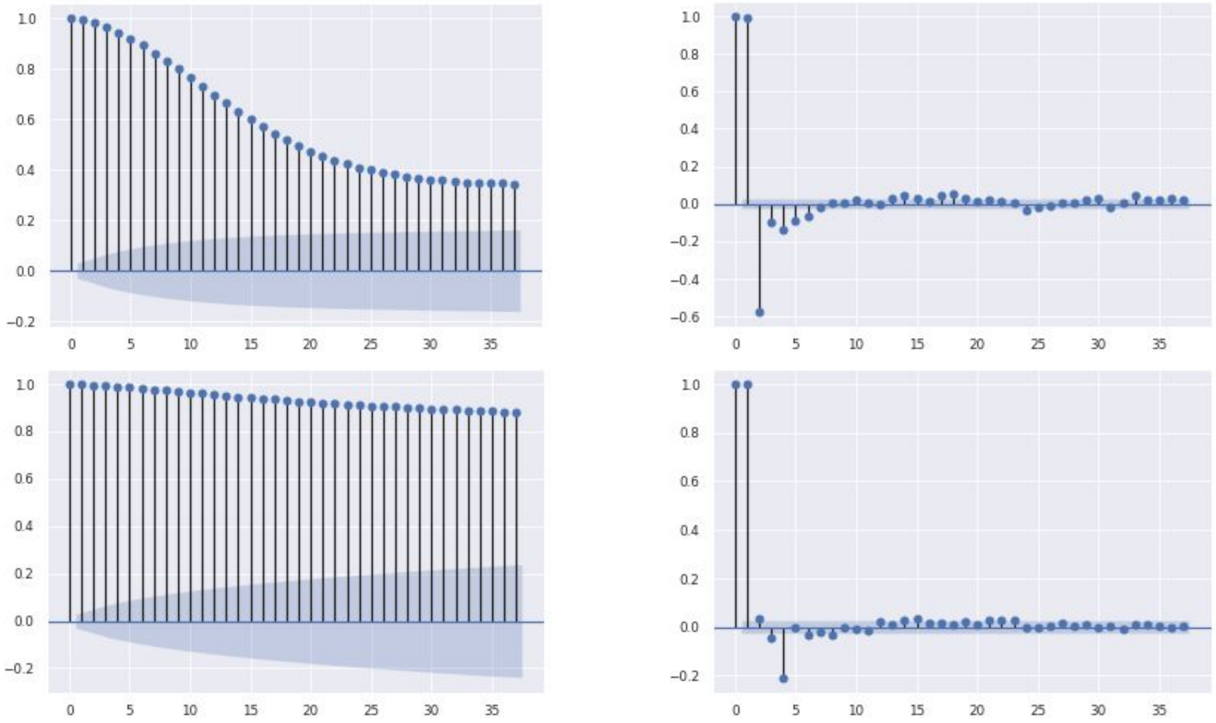
Temperature Derivative Histogram



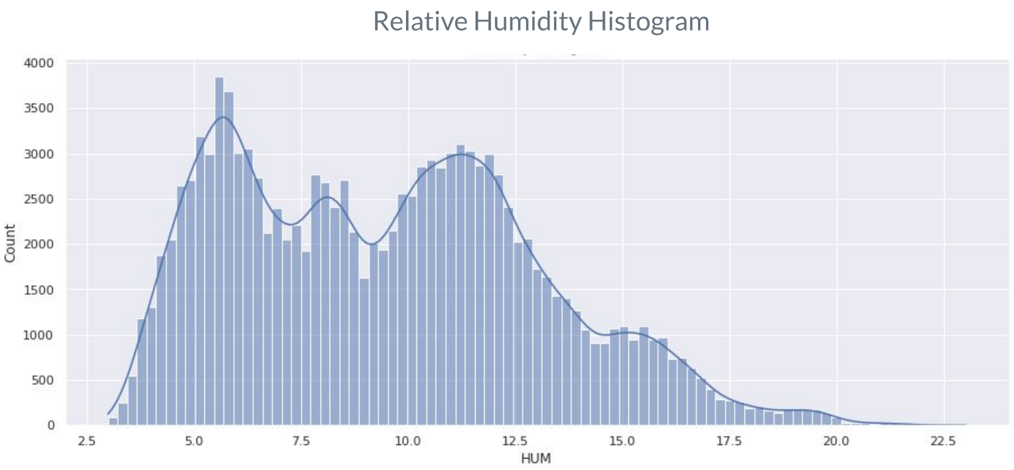
Max. Increase.: 11.91ºC
Max. Decrease: -9.79ºC

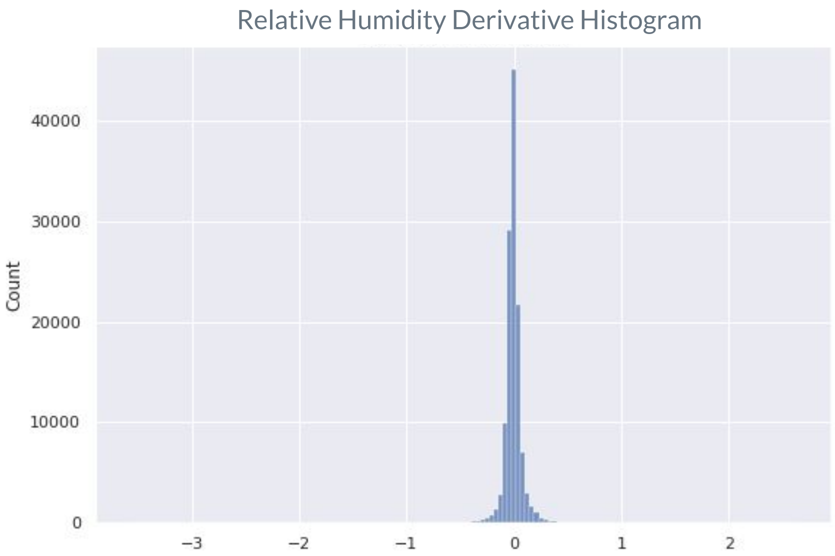
Exploratory Data Analysis: Temperature

Autocorrelation | Partial Autocorrelation



Exploratory Data Analysis: Relative Humidity

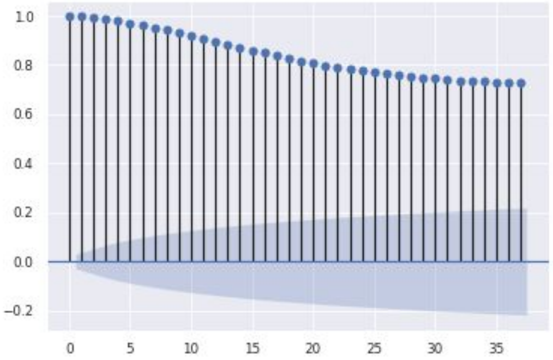


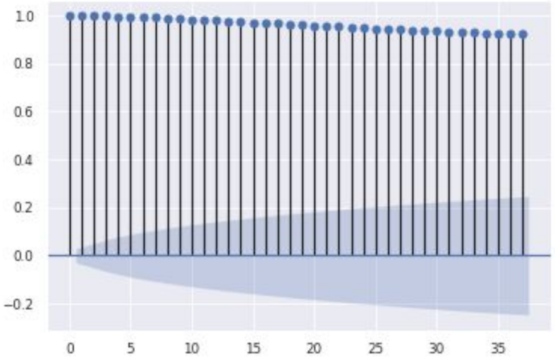


Max. Increase.: 2.64%
Max. Decrease: -3.59%

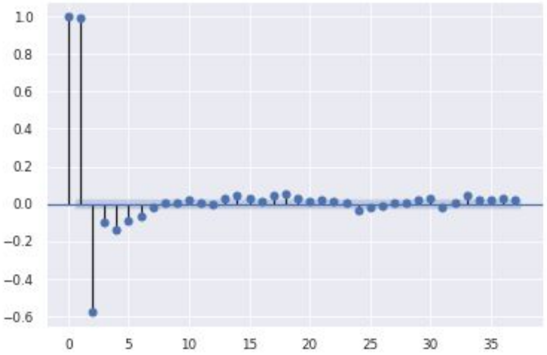
Exploratory Data Analysis: Relative Humidity

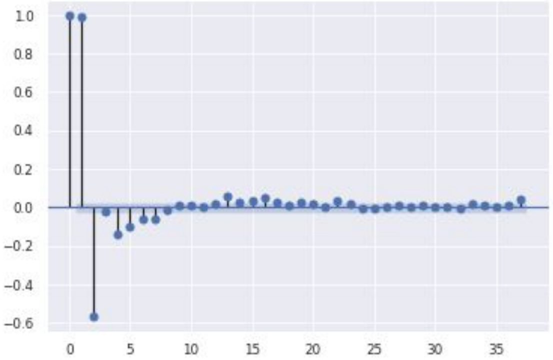
Autocorrelation



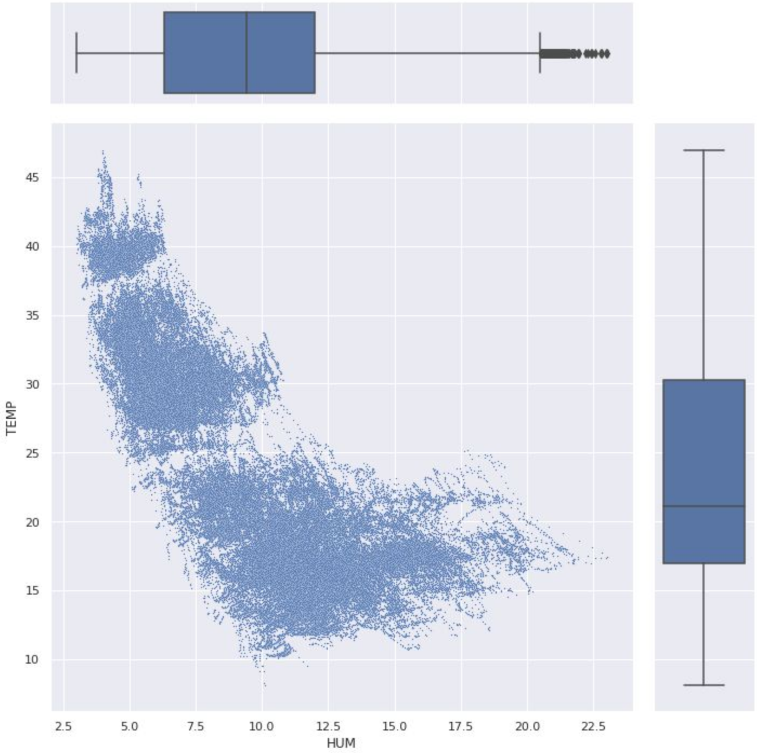


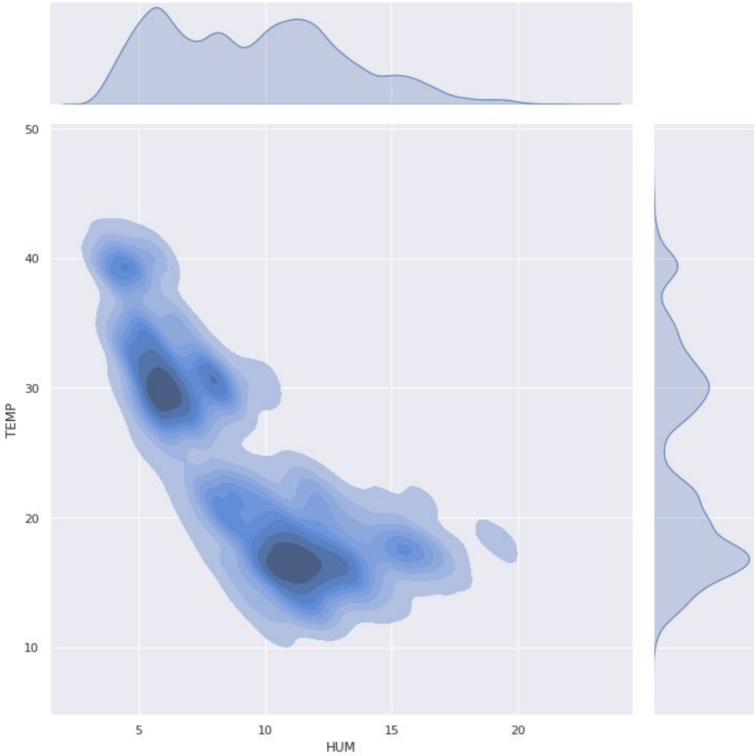
Partial Autocorrelation





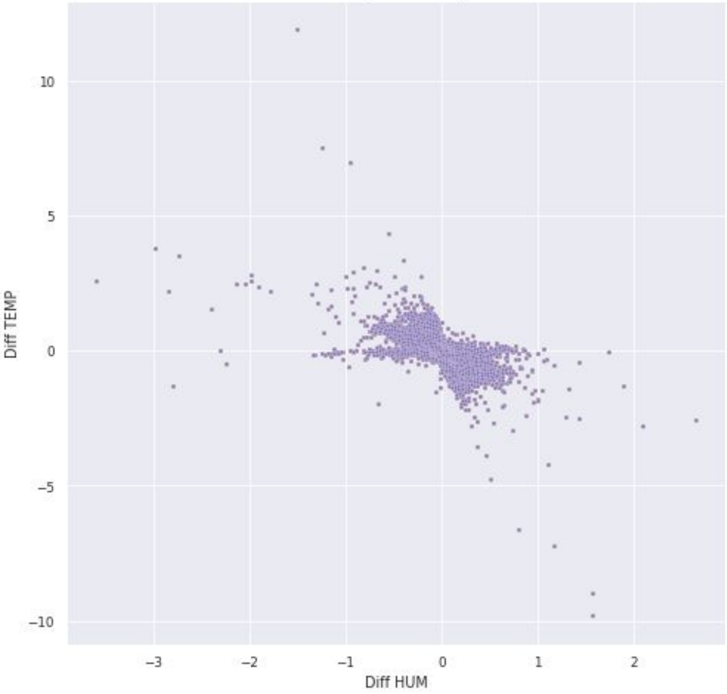
Exploratory Data Analysis: Correlations



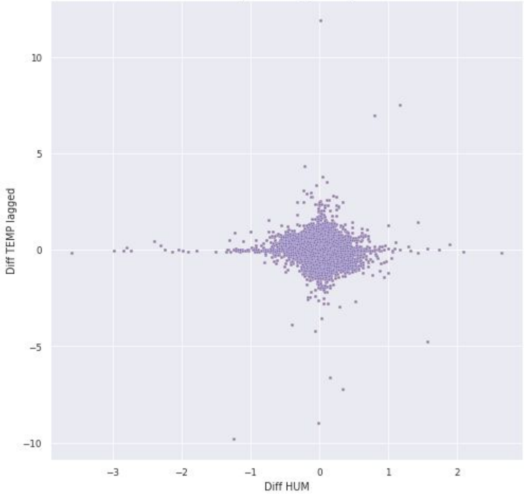


Exploratory Data Analysis: Correlations

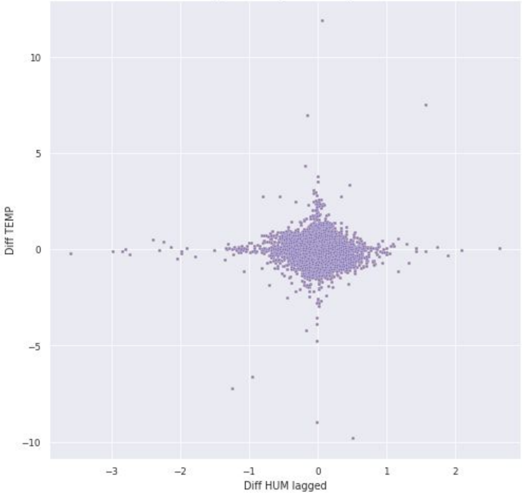
Rel. Humidity Derivative vs. Temperature Derivative



Rel. Humidity Derivative vs. Temperature Derivative
With lagged Temperature



Rel. Humidity Derivative vs. Temperature Derivative
With lagged Humidity



Methodology

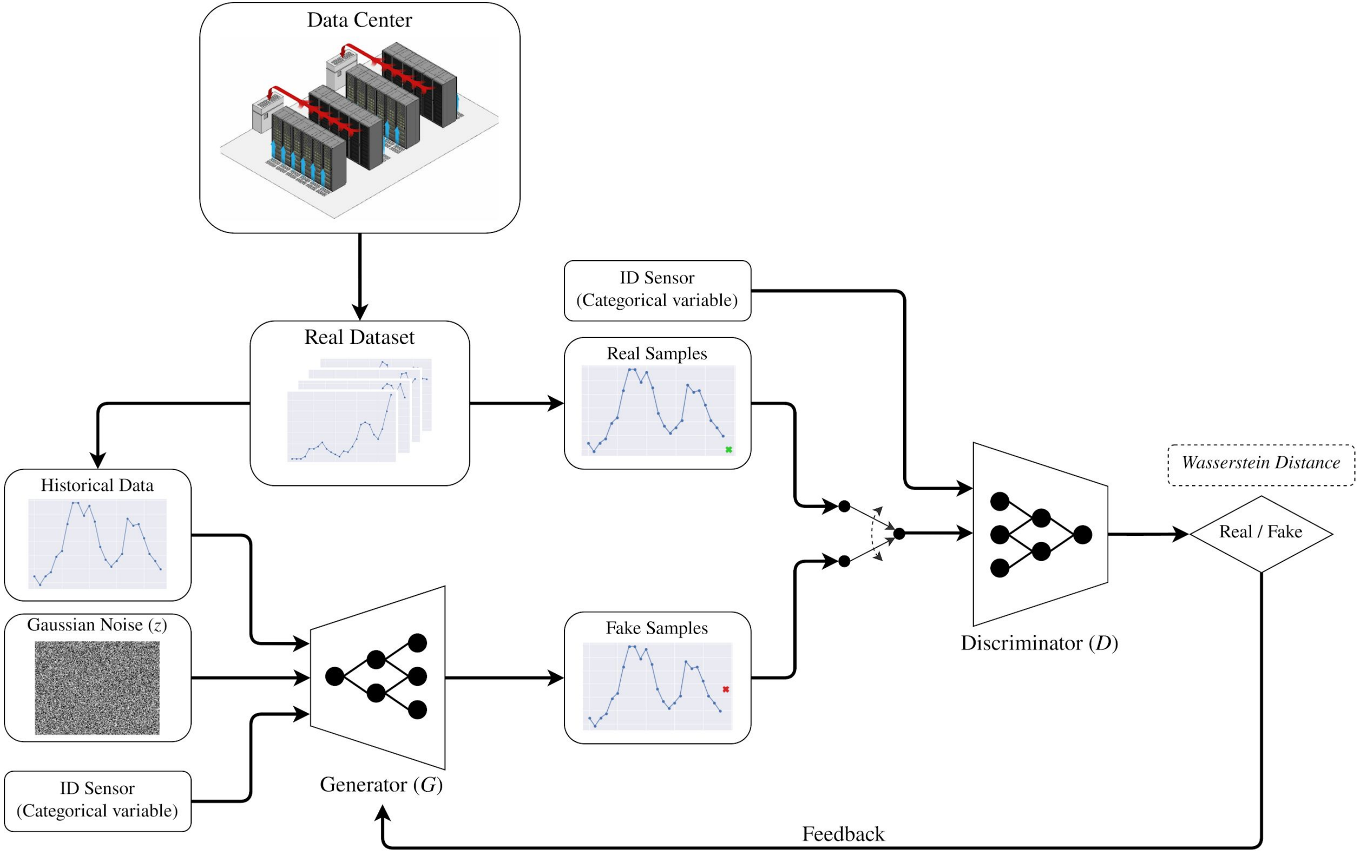
Training Improvements:

- Wasserstein-Loss with Gradient Penalty
- Spectral Normalization
- Two Time-Scale Update Rule (TTUR)
- *GanHacks* by DCGAN authors [https://github.com/soumith/ganhacks]
  - Embedding layers for categorical variables

Data Generation Improvements:

- Metropolis-Hastings GAN (MH-GAN)

Methodology



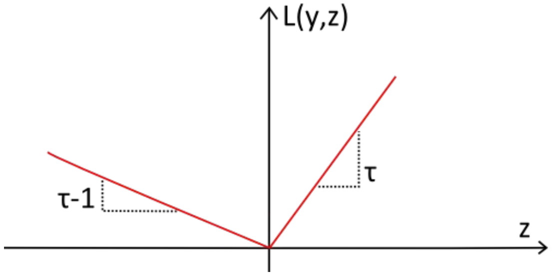
Evaluation Metrics

- Kullback-Leibler (KL) Divergence

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

- Pinball Loss Function

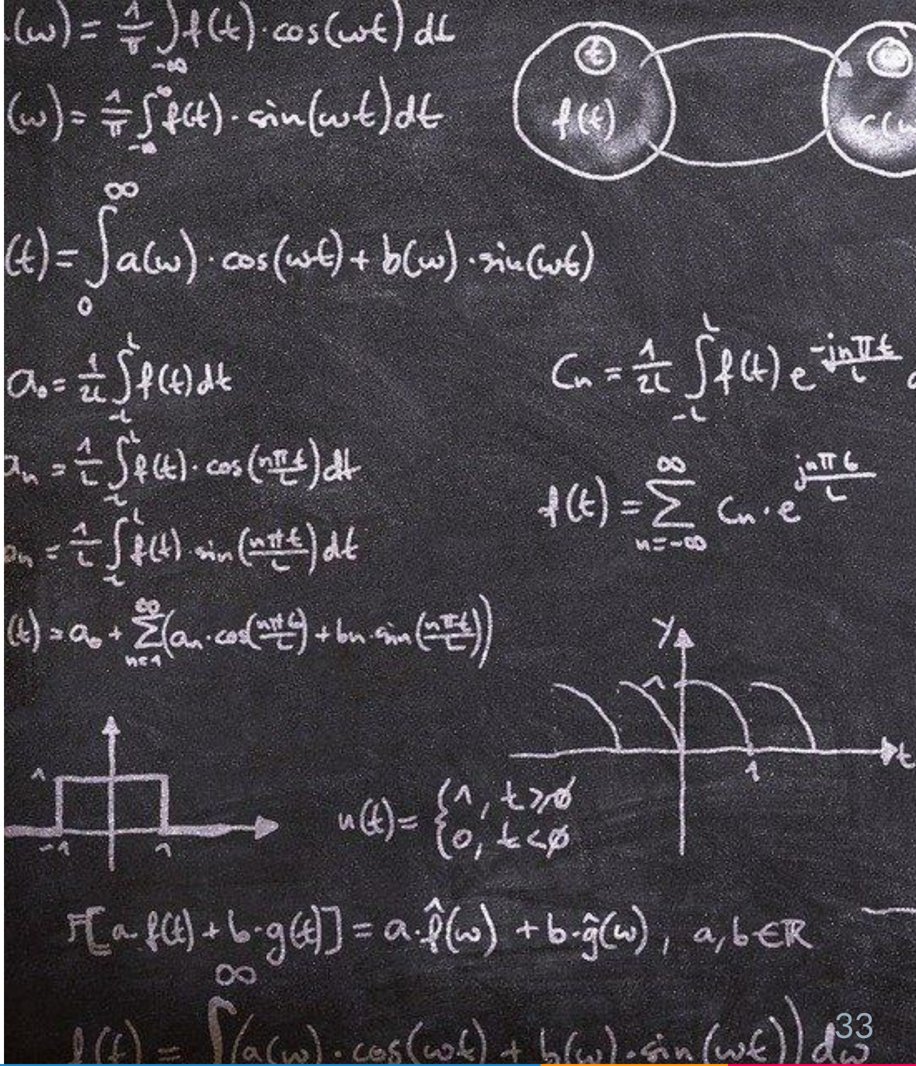
$$\begin{aligned} L_\tau(y,z) &= (y-z)\tau & \text{if } y \geq z \\ &= (z-y)(1-\tau) & \text{if } z > y \end{aligned}$$



- Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Table of Contents



Software Tools

Programming Language	Python 3.6
IDE	*Google Colab*
Deep Learning Framework	*Tensorflow*
Python Libraries for Data Processing	*Pandas*, *Scikit-Learn*, and *Numpy*
Python Libraries for Data Visualization	*Matplotlib* and *Seaborn*

Initial Hyperparameters

Fixed Architectures:

- Generator: Long Short-Term Memory (LSTM) neurons, ~165k parameters
- Discriminator: 1D Convolution neurons, ~735k parameters

Feature Scaling	Min-Max Scaler [-1, 1]
Loss Function	Wasserstein-Loss with Gradient Penalty
Batch Normalization in Generator	✓
Spectral Normalization	✓
Networks Size Ration (Discriminator / Generator)	~4.5
Gaussian Noise Dimension	8
Embedding Layer Dimension	8
Batch Size	64

Experiments

Experiments Methodology: Random scenario generation of 24 time-steps duration, from a validation set

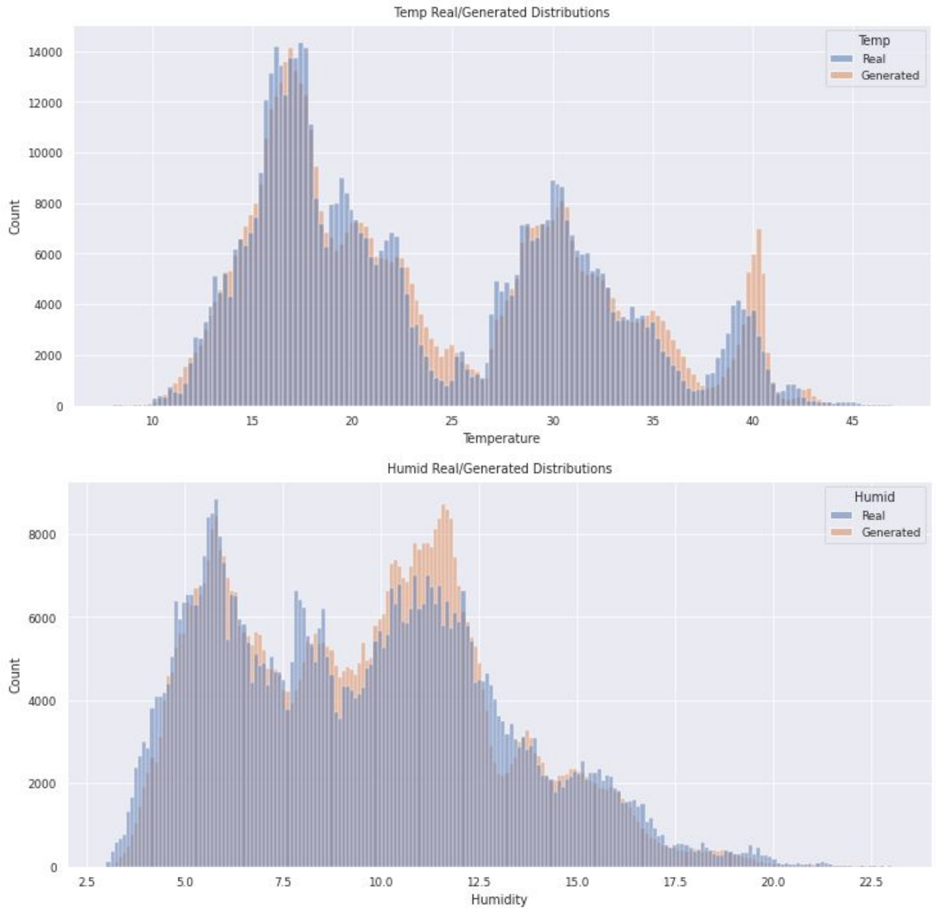
Tuned Hyperparameters:

- Optimizer: Adam / Adabelief
- Skip-Connection Architecture: ✓ or ❌
- Output Activation in Generator: linear / tanh
- TTUR: ✓ or ❌
- Dropout: ✓ or ❌

Best Results									
Hyperparameter					Metrics				
Optimizer	Skip-Connection Architecture	Output Activation	TTUR	Dropout	KL Divergence [bits]	Pinball Loss		MSE	
						Temp.	Humid.	Temp.	Humid.
AdaBelief	❌	linear	✓	✓	**1.432**	**0.488**	**0.219**	**0.977**	**0.438**

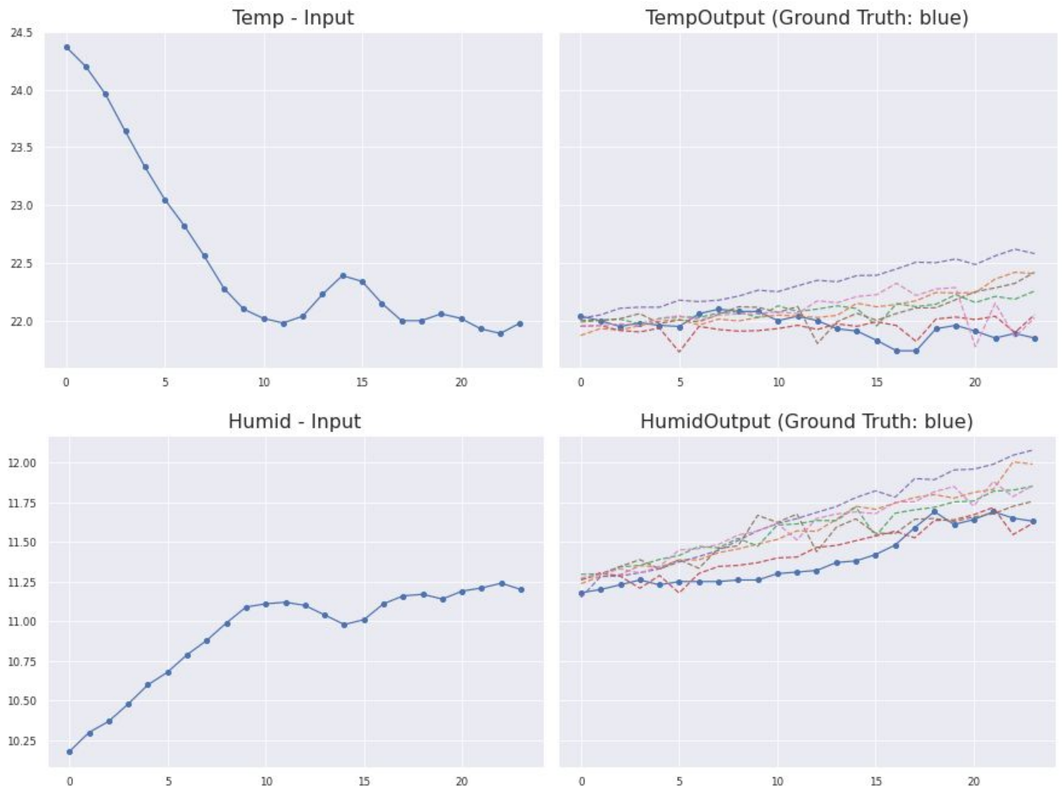
Experiments

Experiments Methodology: Random scenario generation of 24 time-steps duration, from a validation set



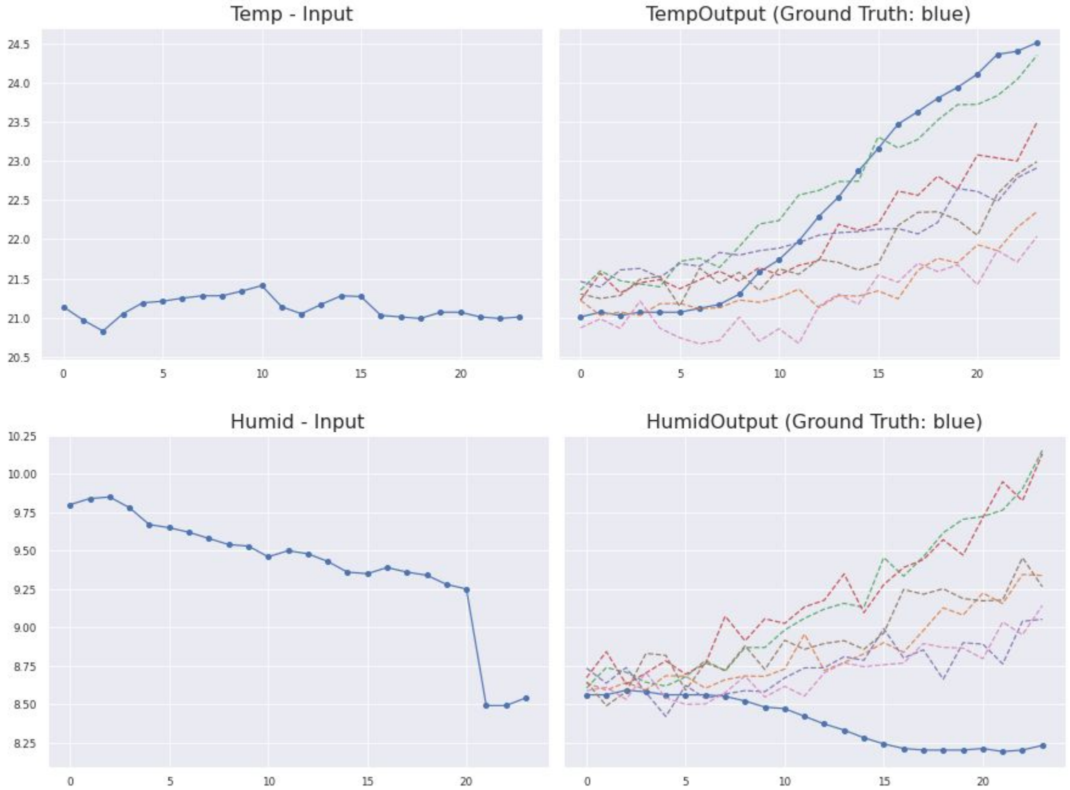
Scenario Generation: Exploring Latent Space

- Low uncertainty example



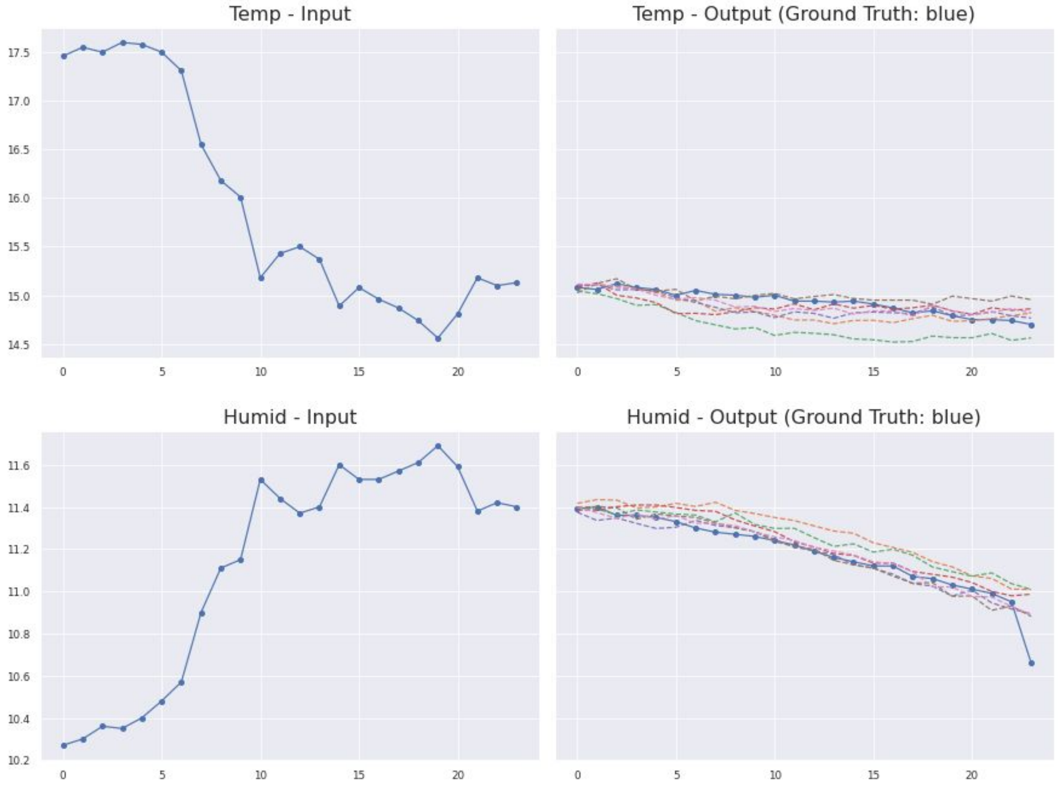
Scenario Generation: Exploring Latent Space

- Increasing uncertainty example



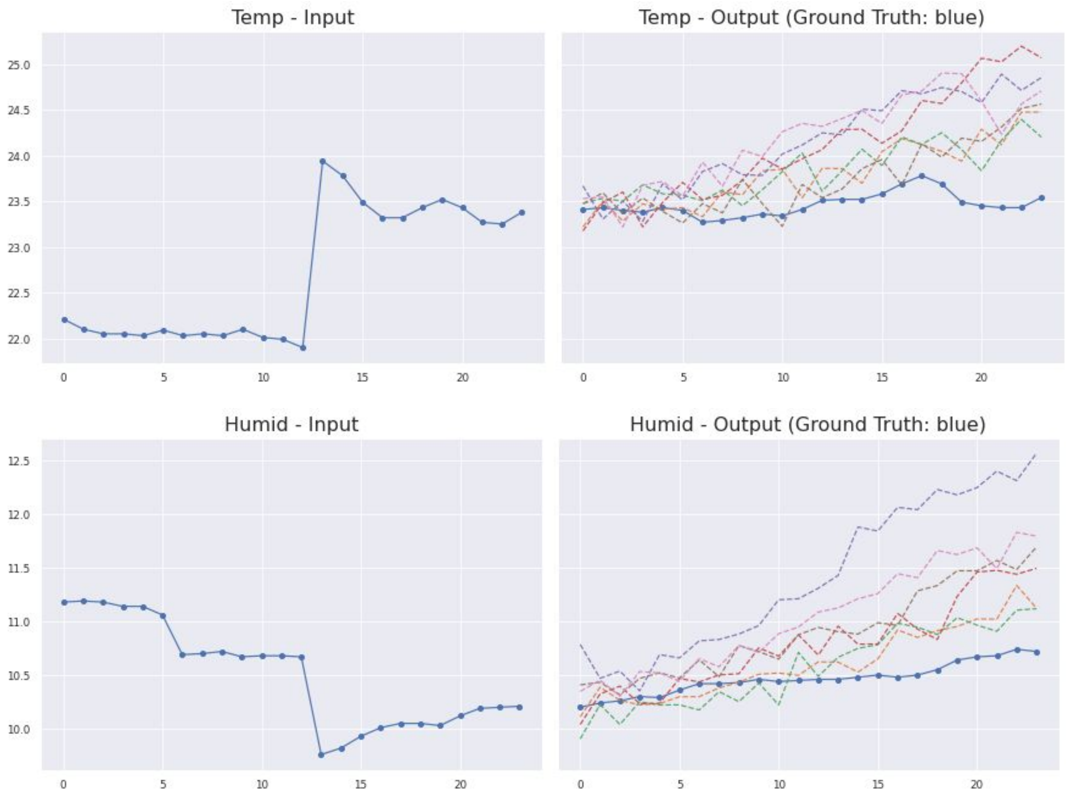
Scenario Generation: MH-GAN

- Low uncertainty example



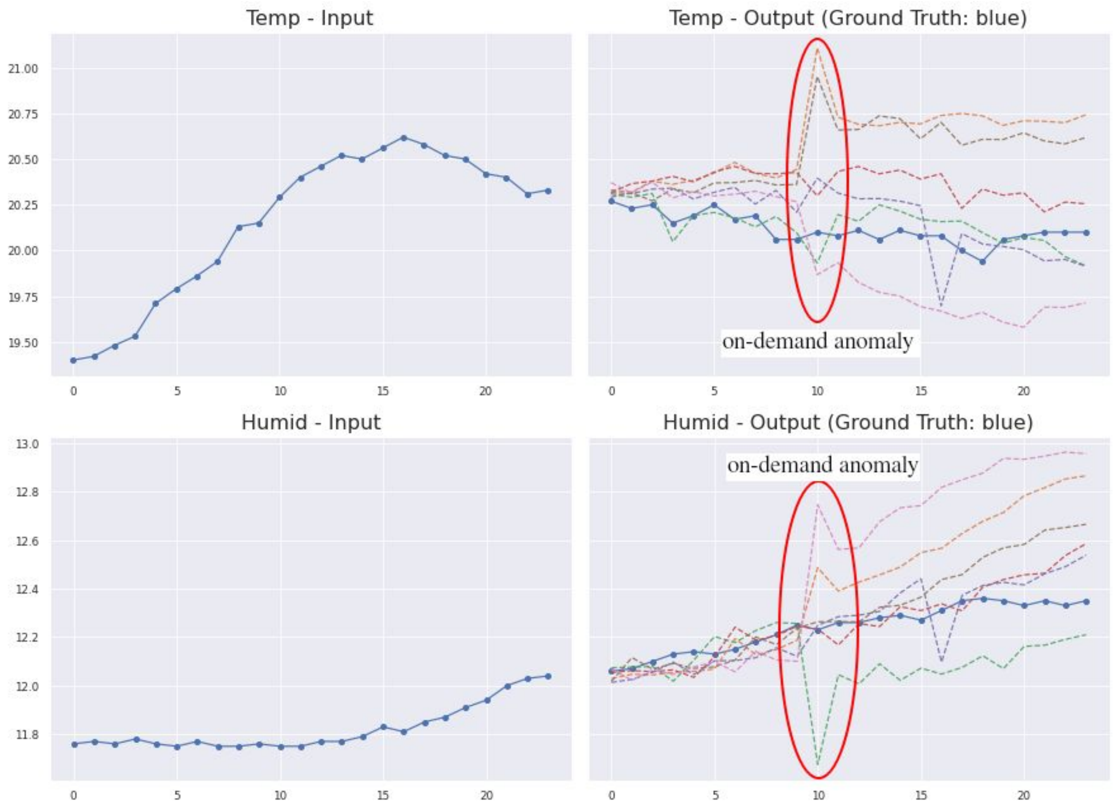
Scenario Generation: MH-GAN

- Increasing uncertainty example



On-Demand Anomaly Generation

**Anomaly Generation Methodology:
Increase standard deviation of Gaussian latent space on specific time instants**



On-Demand Anomaly Generation

Anomaly Generation Methodology:
Increase standard deviation of Gaussian latent space on specific time instants

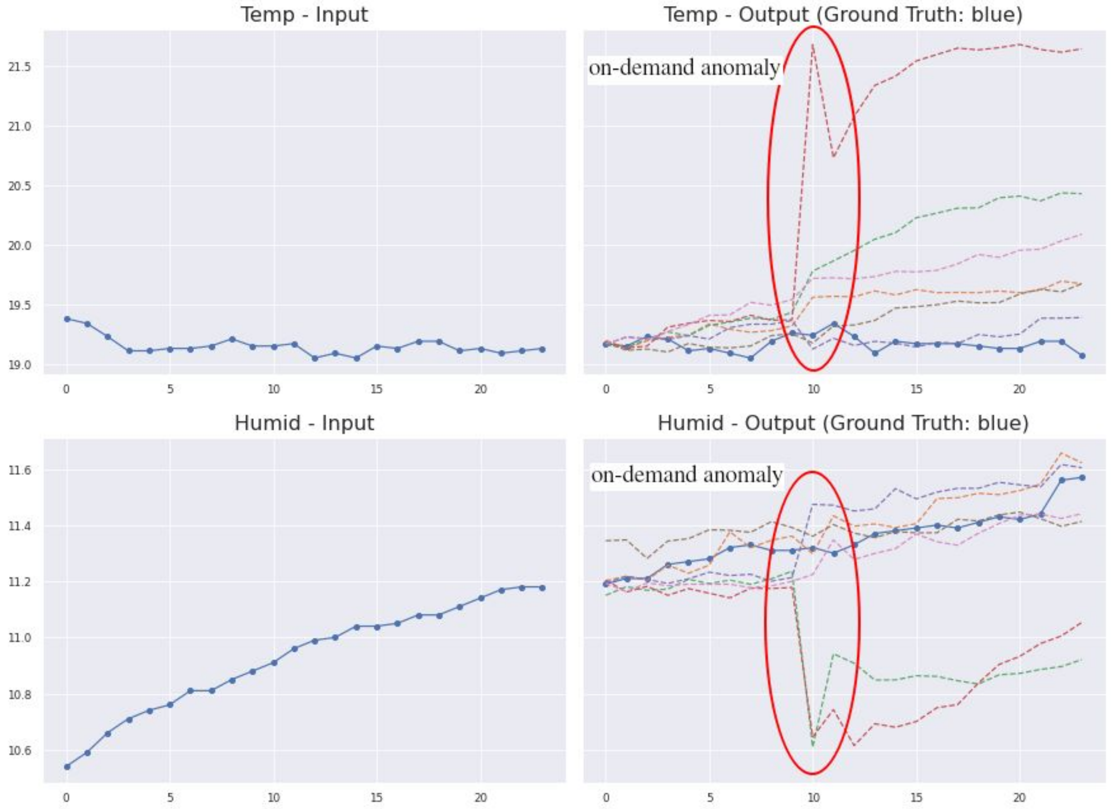


Table of Contents

Conclusions

The results obtained in this work establish a methodology that extends **synthetic time-series data** applications, enabling to use **categorical variables** and **multi-variable scenario generation**.

On-demand anomaly generation introduces significant data **variability** without additional effort or endangering electronic equipment integrity.

The proposed methodology can be employed in any similar time-series-like problems in **other fields of application**.

Our research will help to **apply synthetic data generation to different real-world time-series** problems. Through the proposed use case, enabling **better optimization of Data Centers**, and thus, **a more sustainable and greener future**.

Open Issues

- **Scarce Research on Time-Series GANs**

  Leads to unstable training and the need for intensive hyperparameter search.

- **Generated Data Bias**

  Generated data variability is limited and biased by the available data.

  We also need better metrics to measure “realism”.

- **Human Supervision is Needed**

  The fact that the training process is not perfect and the accessible data is limited, implies that some generated samples do not correspond to real situations.

- **Relatively Large Amounts of Data are Needed**

  GANs require “large” amounts of data for stable training. Still, this amount is tiny compared to that needed to achieve state-of-the-art results in Deep Learning models

Future Work

Hyperparameter Optimization

Analyze Scalability of Multi-Variables Datasets

Further Study on the Usefulness of the Results

E.g., Train on Synthetic, Test on Real

Disentangled Latent Space

[54]

Add Supplementary Conditional Information

E.g., Freq., ARIMA, Time Information…

Explore Alternatives Evaluation Metrics

E.g., Divergence of Freq. Spectrum

Explore Further GAN Architectures and Improvements

E.g., StyleGAN, Model Weight Averaging…

Thanks!

Any questions?

BACKUP

XXXX