ETSIT UPM

POLITÉCNICA

Departamento de Ingeniería Electrónica

Universidad Politécnica de Madrid

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

Student: Jaime Pérez Sánchez

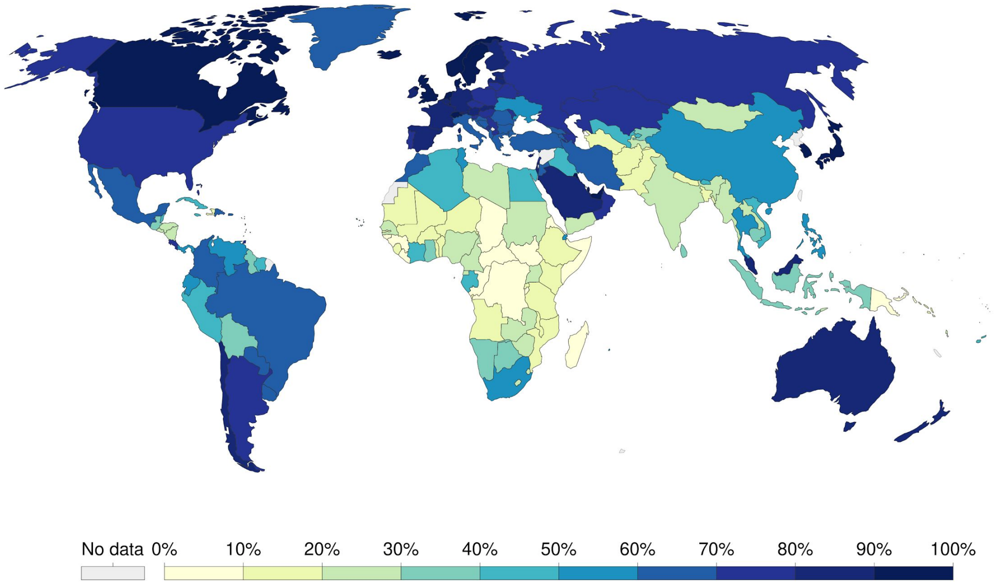
Advisor: Patricia Arroba García

Table of Contents

Digital Economy & Society

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

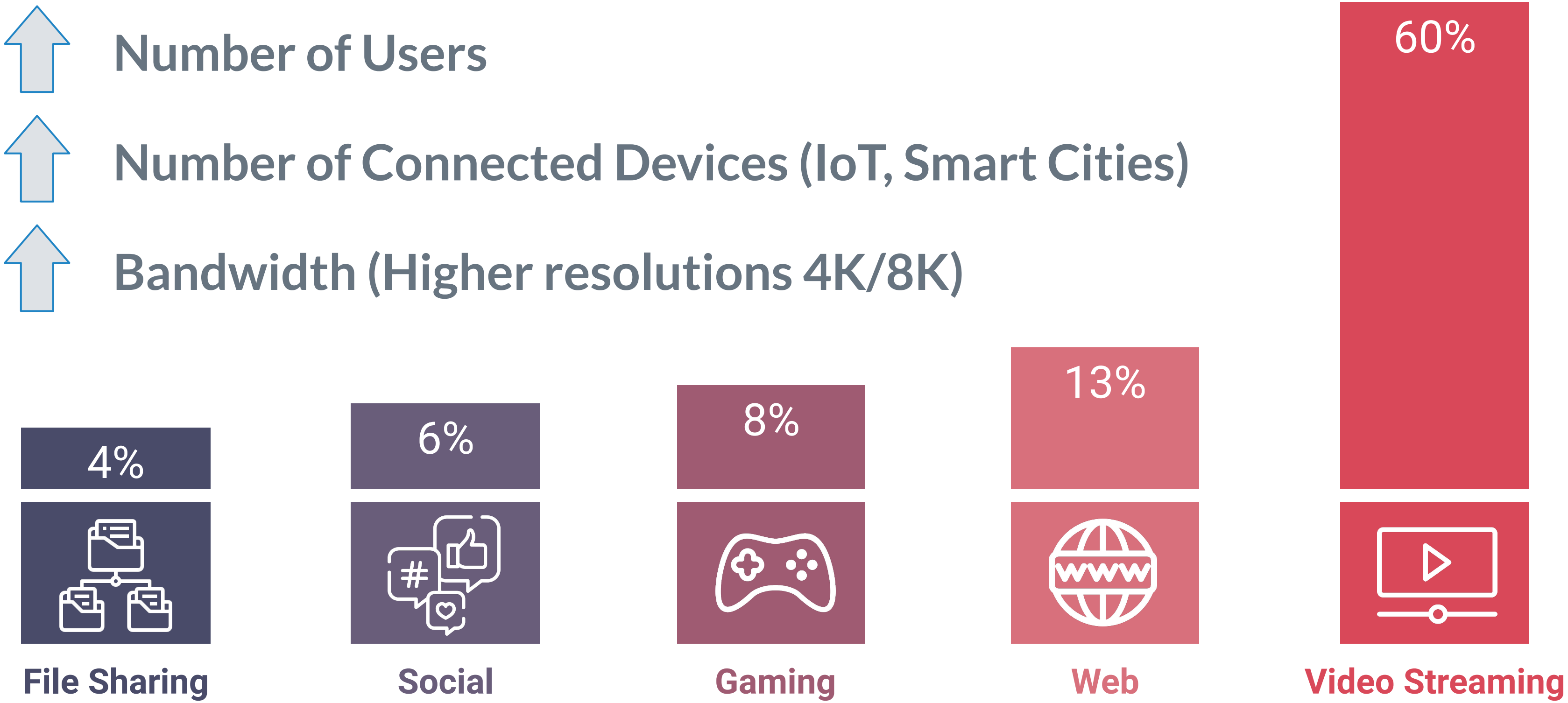
Share of population using the Internet (2017)



Source: World Bank Data (2017)

Internet Traffic Share

- **Number of Users**
- **Number of Connected Devices (IoT, Smart Cities)**
- **Bandwidth (Higher resolutions 4K/8K)**



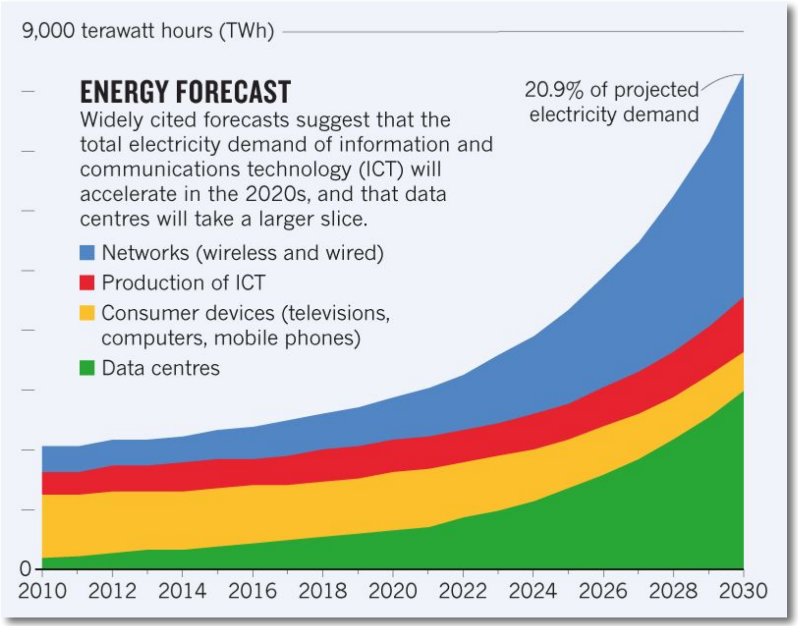
Cloud Computing Paradigm

Compute Instances

- By **550%** and IP traffic by **1,000%** from 2010 to 2018

Energy Consumption

- **1% of Global Energy Consumption** in 2018 (205 TWh)
- **Average ~40% Cooling Energy**



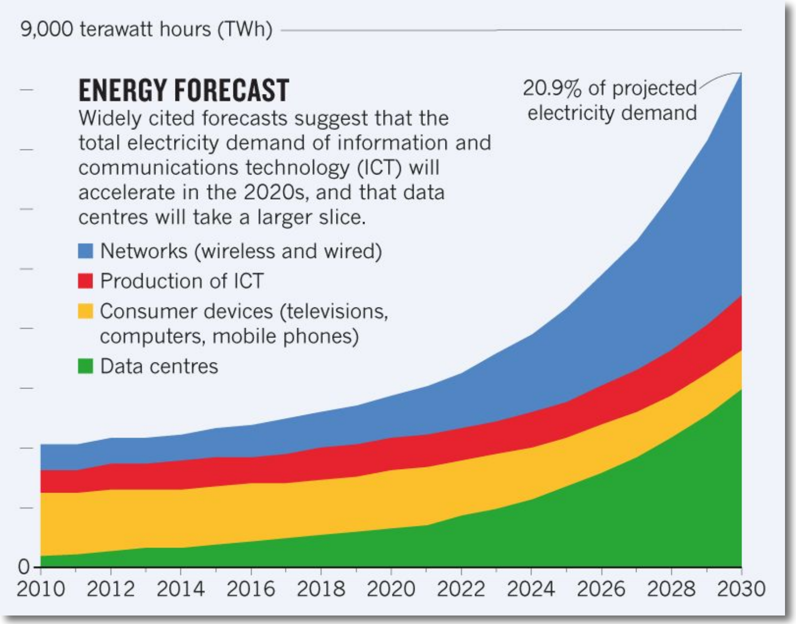
Cloud Computing Paradigm

Compute Instances

- By **550%** and IP traffic by **1,000%** from 2010 to 2018

Energy Consumption

- **1% of Global Energy Consumption** in 2018 (205 TWh)
- **Average ~40% Cooling Energy**



We urgently need better optimization in Data Centers!

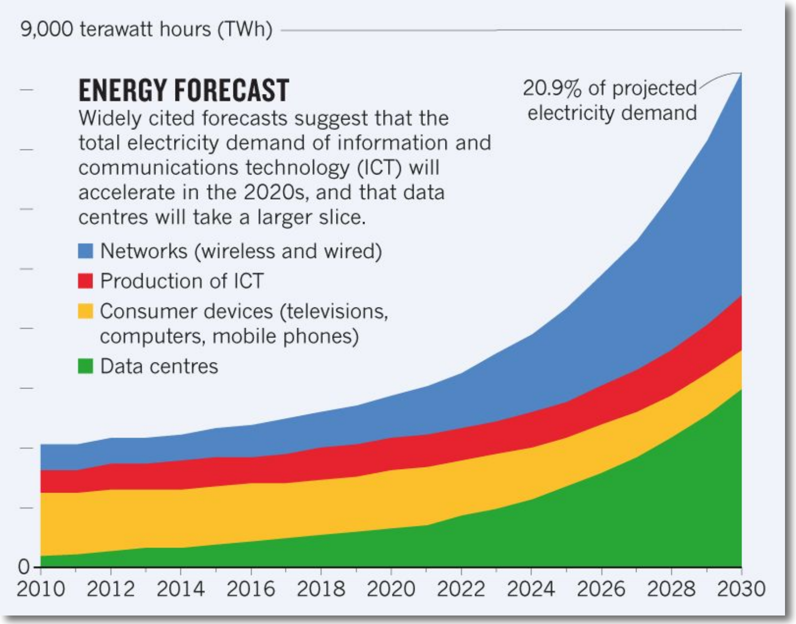
Cloud Computing Paradigm

Compute Instances

- By **550%** and IP traffic by **1,000%** from 2010 to 2018

Energy Consumption

- **1%** of **Global Energy Consumption** in 2018 (205 TWh)
- **Average ~40% Cooling Energy**



We urgently need better optimization in Data Centers!

However, Data Centers are very challenging to optimize...

Data-Driven Optimization

- **Machine Learning** and **Deep Learning** optimization:
  - Hyperscale Data Centers PUE 1.1 or lower (ideal = 1)
- **Why cannot we apply this approach globally?**
  - We need huge amounts of data!
    - **Expensive** and **time-consuming**
    - **Endanger the electronic equipment** integrity
  - Proprietary data sharing is risky
    - Information leakage

* (PUE) Power Usage Effectiveness = $\frac{\textit{Total Energy}}{\textit{IT Equipment Energy}}$

Data-Driven Optimization

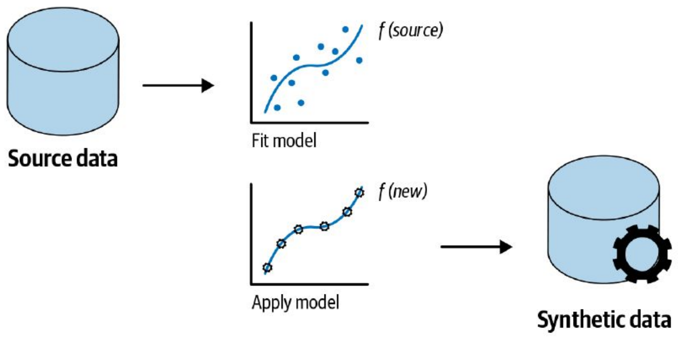
- **Machine Learning** and **Deep Learning** optimization:
  - Hyperscale Data Centers PUE 1.1 or lower (ideal = 1)
- **Why cannot we apply this approach globally?**
  - We need huge amounts of data!
    - **Expensive** and **time-consuming**
    - **Endanger the electronic equipment** integrity
  - Proprietary data sharing is risky
    - Information leakage

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

* (PUE) Power Usage Effectiveness = $\frac{\textit{Total Energy}}{\textit{IT Equipment Energy}}$

MSc Thesis Objective: Synthetic Data Generation

- Objective: Produce **synthetic time-series data**
  - **Generation of realistic scenarios** given a real computing situation
- **Challenges** to apply it to Data Centers field:
  - Data must be **multi-variable** and of different nature (e.g., **categorical data**)
  - Data must be **heterogeneous** (i.e., variability)

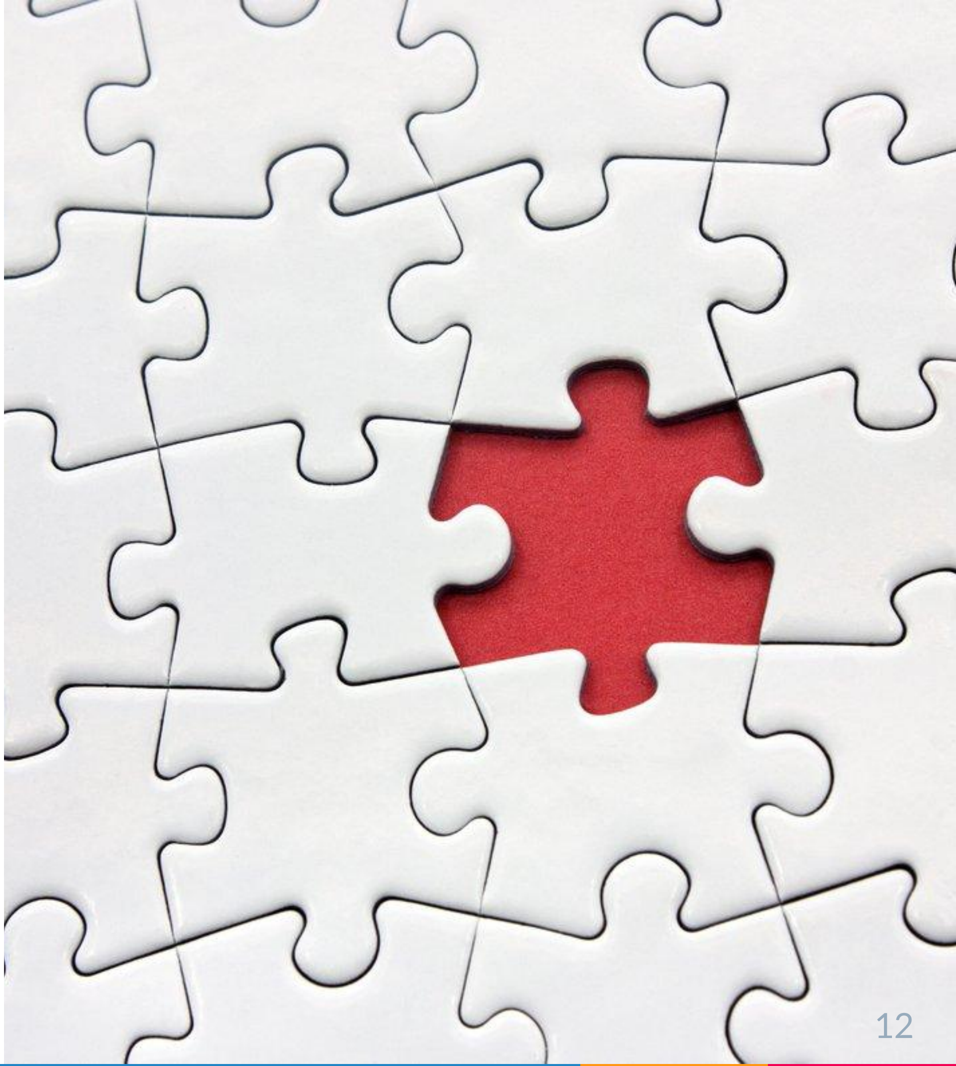


Generative Adversarial Networks (GANs)

Real Images **Synthetic Images**

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

- ❏ **Poor work** on **multi-variate data** with **non-linear** relationships
- ❏ Cannot handle data from different natures (i.e., categorical)
- ❏ Require expertise knowledge

State Of the Art: Synthetic Time-Series Data

Generation Based on **GANs**:

PROS

- **Handle multi-variable data** with **non-linear** relationships, and from different natures (i.e., **categorical**)
- Flexible tune generation (i.e. **variability**)
- Can be applied in conjunction with traditional data augmentation methods
- Outstanding empirical results

CONS

- Unstable training (partially solved)
- It needs relatively large amounts of data
- Difficult to interpret (i.e., black-box model)

Contributions

		Data Augmentation	Scenario Generation					*Ours*
		[63]	[66]	[67]	[68]	[69]	[70]	
Obj. 1	**Scenario** Generation	❌	✓	✓	✓	✓	✓	✓
Obj. 2	**Multi-variable** Generation	✓	❌	❌	❌	❌	❌	✓
Obj. 3	Introduce **Categorical** Variables	✓	❌	❌	❌	✓	❌	✓
Obj. 4	Generate on-demand **anomalous** situations (increase variability)	❌	❌	❌	❌	❌	❌	✓

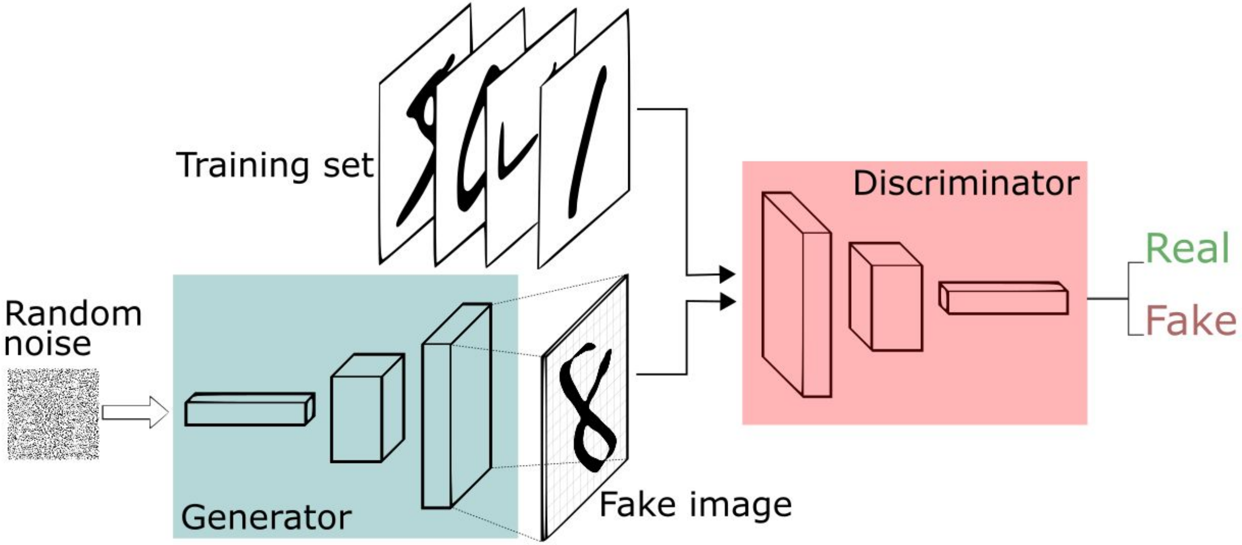
Table of Contents

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)

Incredibly fast evolution!

"Generative Adversarial Networks [...] and the variations that are now being proposed is the most interesting idea in the last ten years in Machine Learning"

- Yann LeCun, Director of AI research at Facebook

Table of Contents

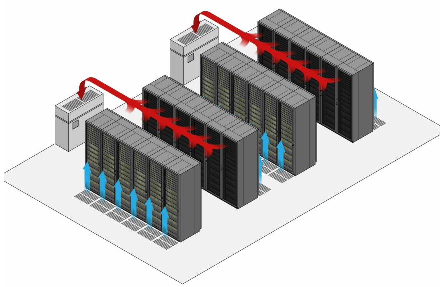
Cooling Data Centers Optimization

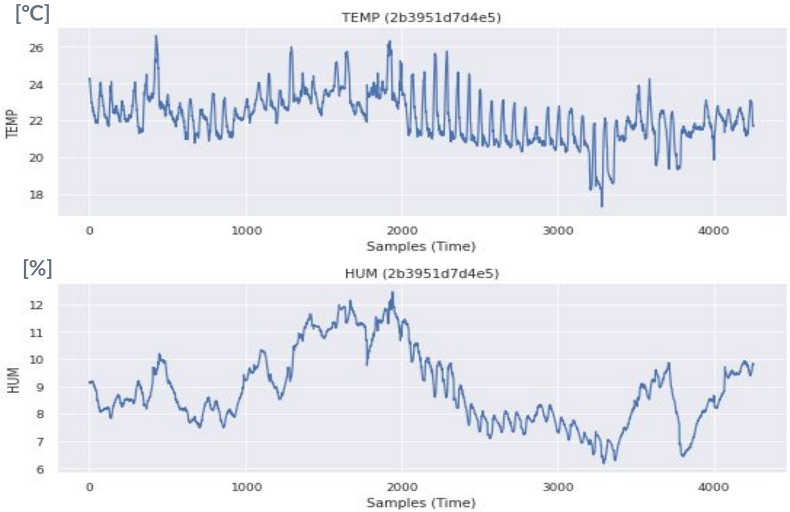
Real data gathered from an operating Data Center:

- 35 sensors → **Temperature** and **Humidity** collected **every 10 minutes**
- **Scenario Generation → Synthetic Datasets → Improve Cooling Optimization Models**

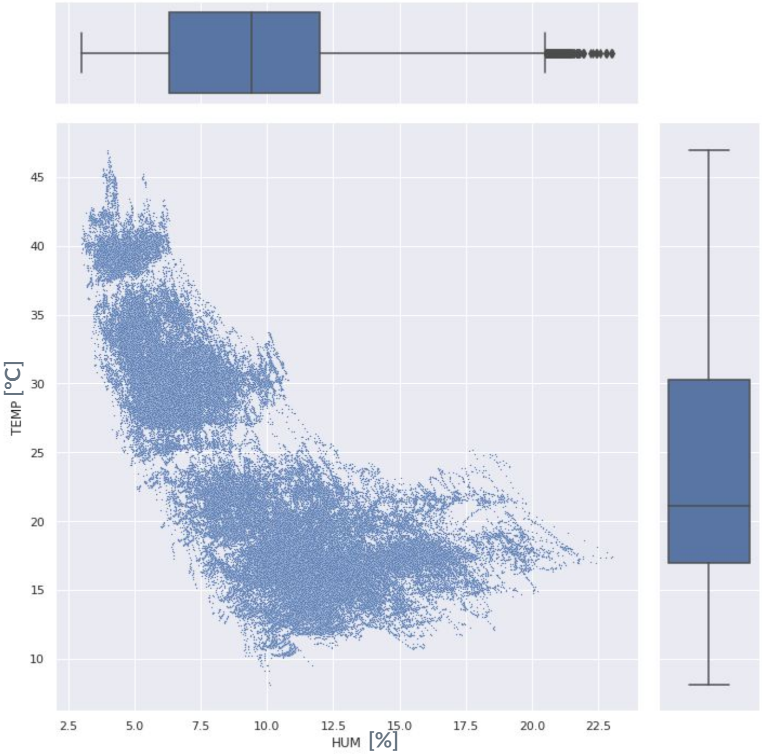
Adam Your datahome

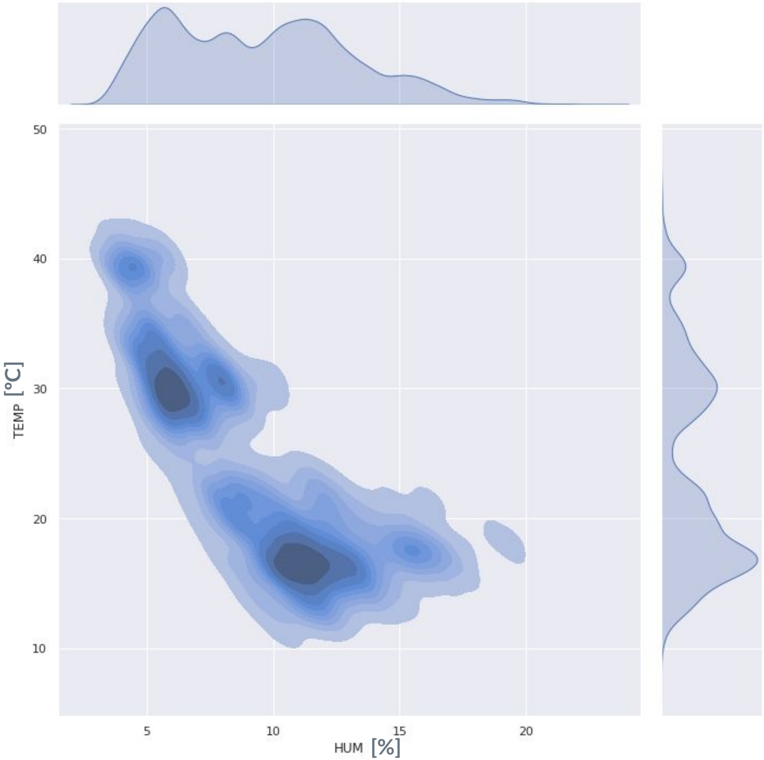
TycheTools



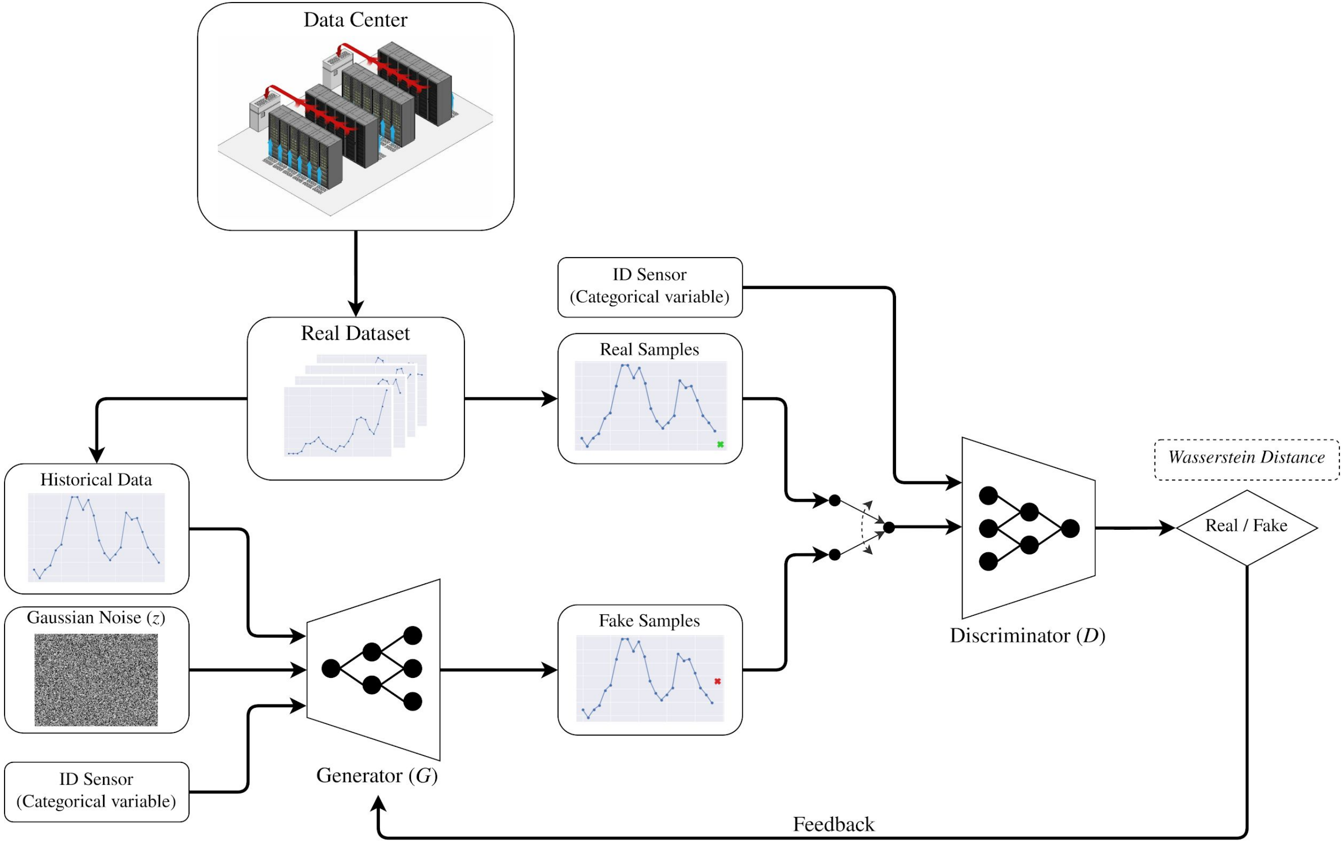


Exploratory Data Analysis: Correlations



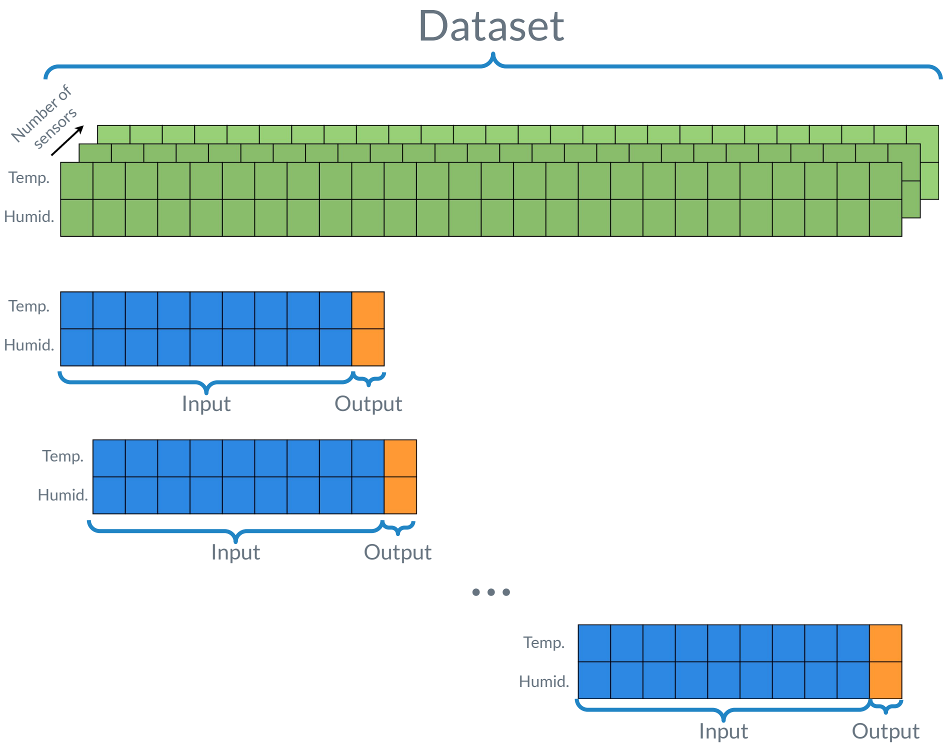


Methodology

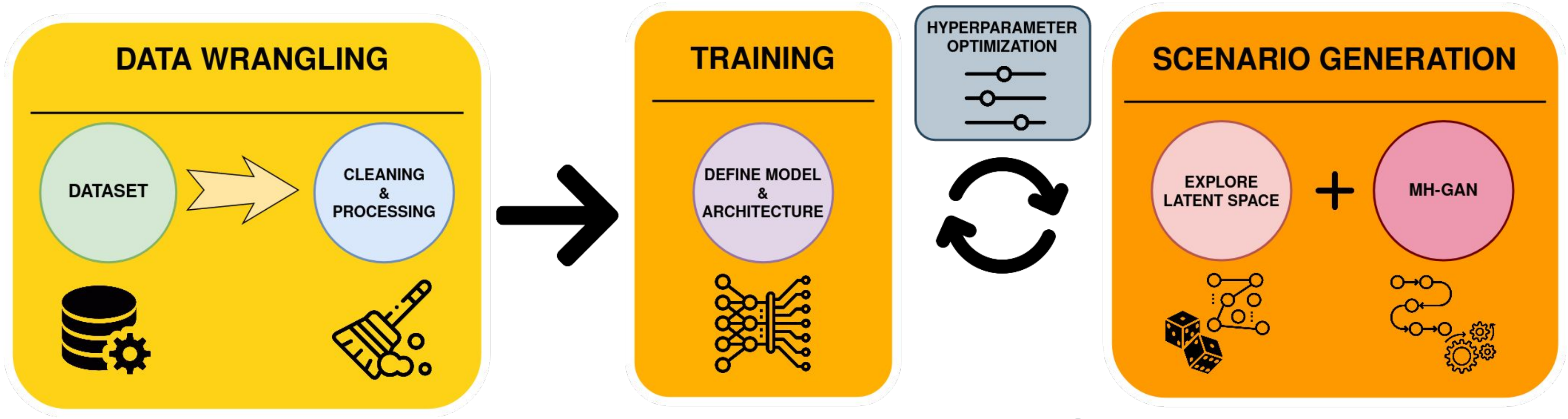


Model In/Out

- **Model Inputs:**
  - Multi-variable 24 time steps (4 hours)
  - Categorical variable (ID sensor)
  - Random Gaussian Noise
- **Model Output:**
  - Multi-variable 1 time step (10 min.)
  - We concatenate predictions to obtain forecasts farther into the future



Methodology Improvements



Train Improvements:

- ▷ Wasserstein-Loss with Gradient Penalty (WGAN-GP)
- ▷ Spectral Normalization
- ▷ Two Time-Scale Update Rule (TTUR)

Output Improvements:

- ▷ Metropolis-Hastings GAN (MH-GAN): Sampling method based on Markov Chain Monte Carlo (MCMC)

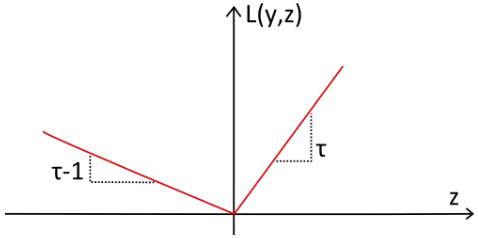
Evaluation Metrics

- 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

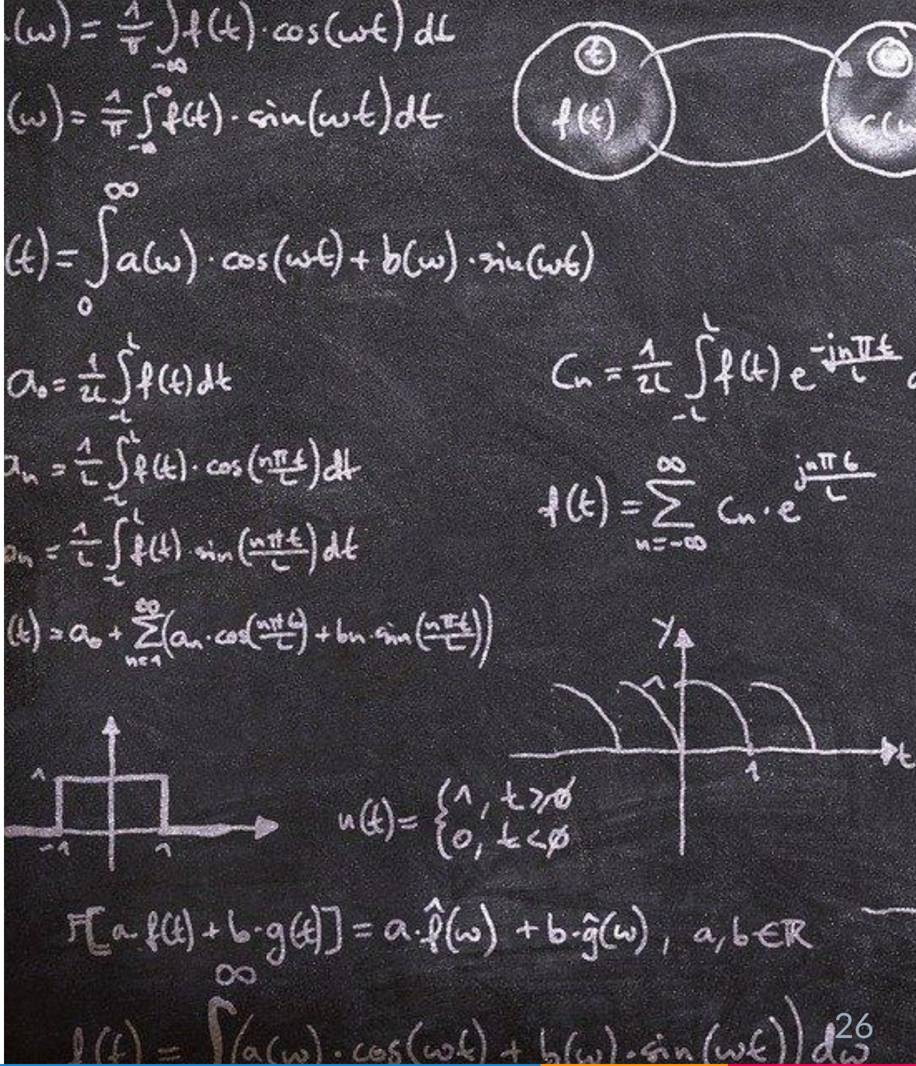
$$\begin{aligned} L_\tau(y, z) &= (y - z)\tau && \text{if } y \geq z \\ &= (z - y)(1 - \tau) && \text{if } z > y \end{aligned}$$



- Mean Squared Error

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

Table of Contents



Initial Hyperparameters

Fixed Network Architectures:

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

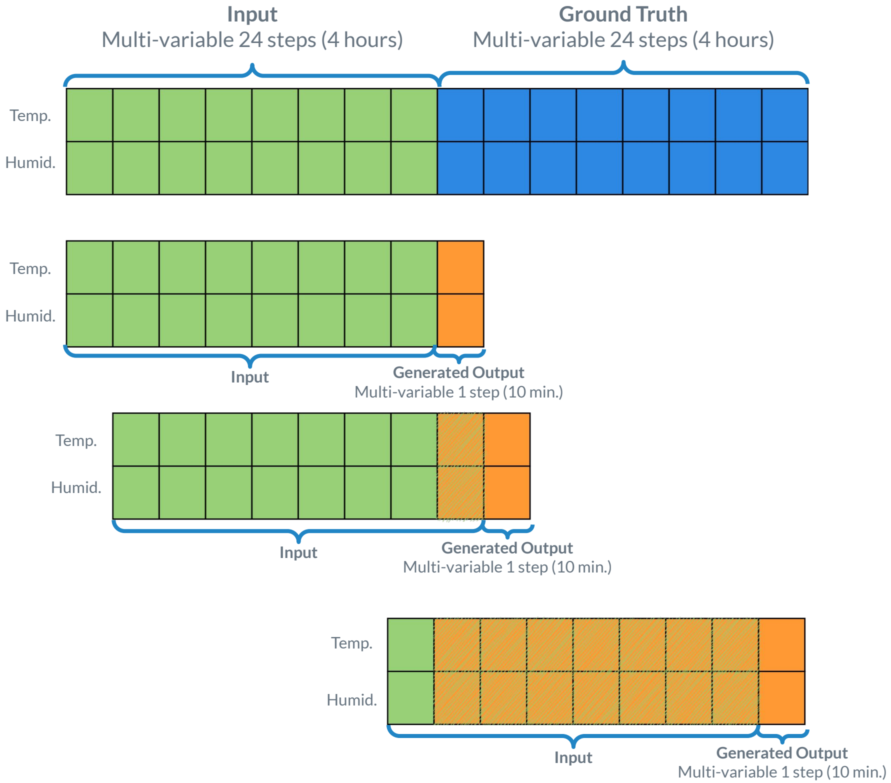
Tuned Hyperparameters:

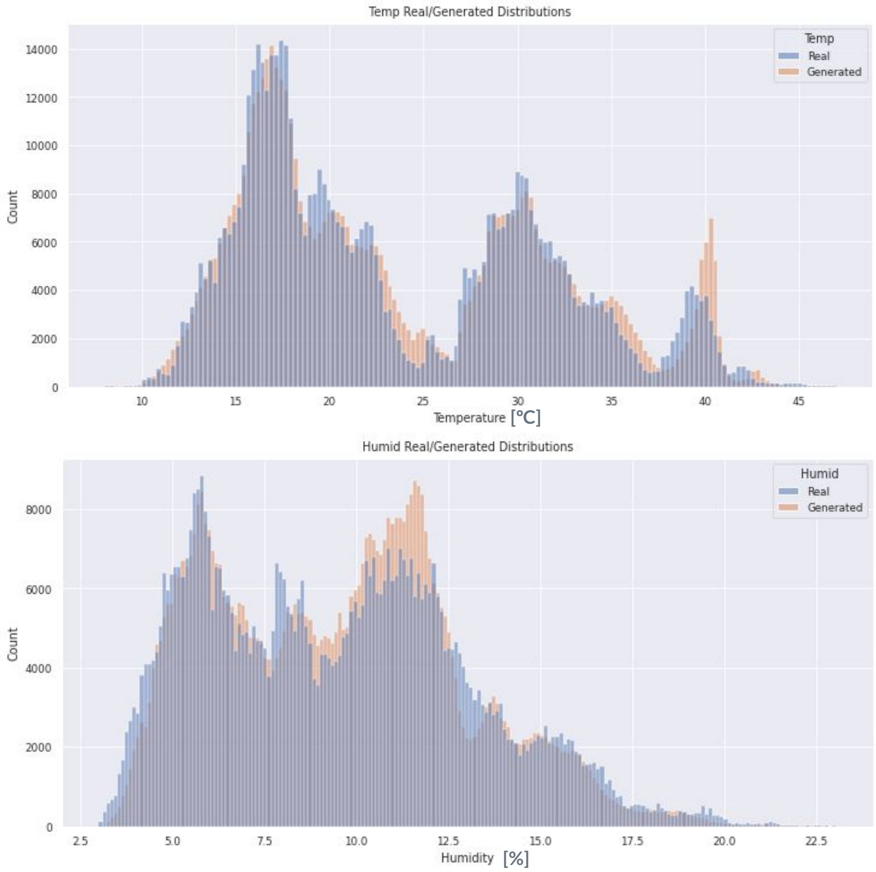
- Optimizer: Adam / Adabelief
- Output Activation in Generator: linear / tanh
- Skip-Connection Architecture, TTUR, Dropout: ✓ or ❌

Best Results on Validation Set - 24 steps predictions (4 hours)

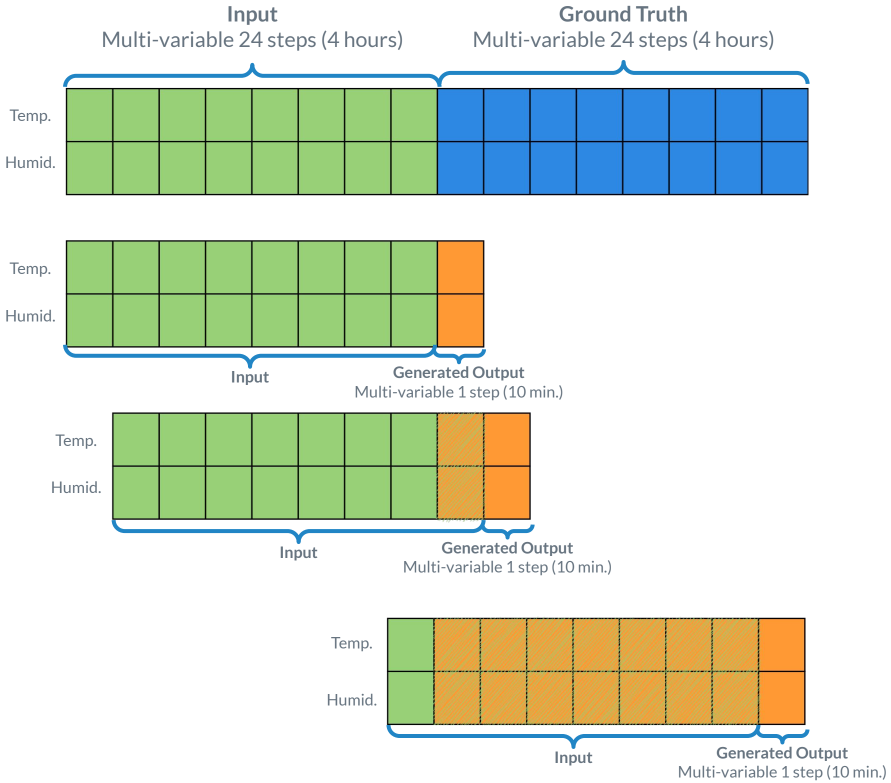
Hyperparameter					Metrics				
						Pinball Loss		RMSE	
Optimizer	Skip-Connection Architecture	Output Activation	TTUR	Dropout	KL Divergence [bits]	Temp.	Humid.	Temp. [°C]	Humid. [%]
AdaBelief	❌	linear	✓	✓	**1.432**	**0.488**	**0.219**	**0.988**	**0.661**

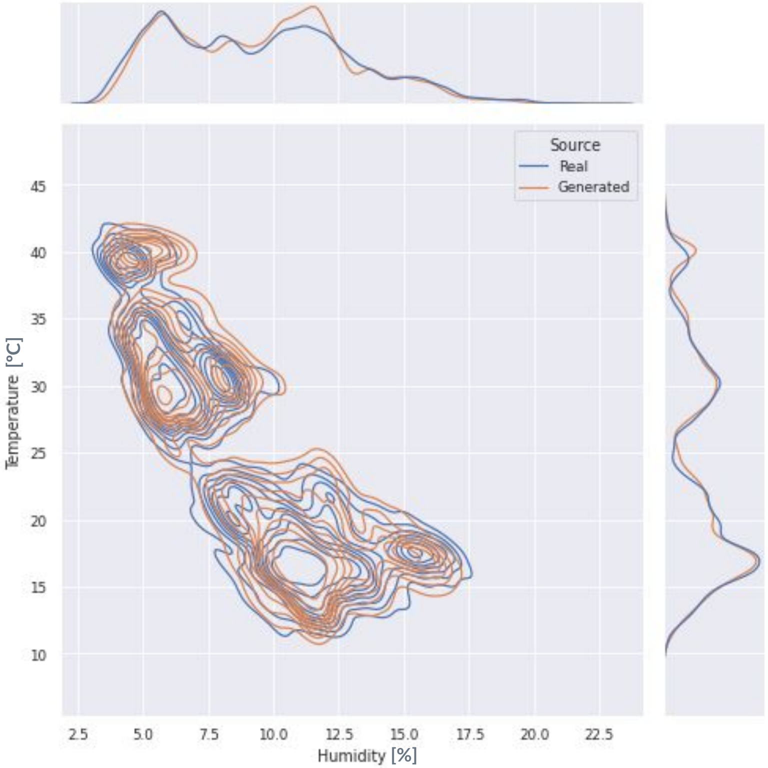
Results on Validation Set





Results on Validation Set



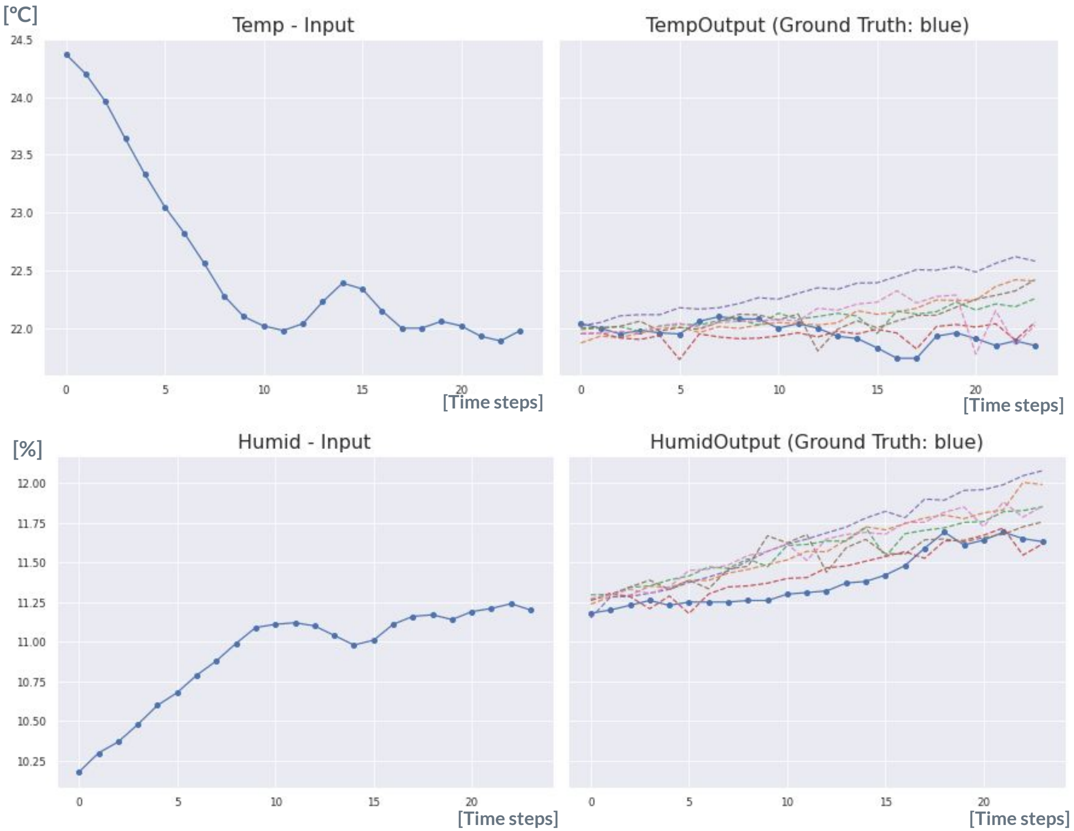


Scenario Generation: Random Exploration

- 6 scenarios

Each color represent a different scenario

- 24 steps predictions concatenated (4 hours)



Random Exploration results on Test set:

KL Divergence = 1.133 bits

RMSE Temperature = 1.033 °C

RMSE Humidity = 0.673 %

Scenario Generation: Random Exploration

- 6 scenarios

Each color represent a different scenario

- 24 steps predictions concatenated (4 hours)

<u>Random Exploration results on Test set:</u>
KL Divergence = 1.133 bits
RMSE Temperature = 1.033 °C
RMSE Humidity = 0.673 %

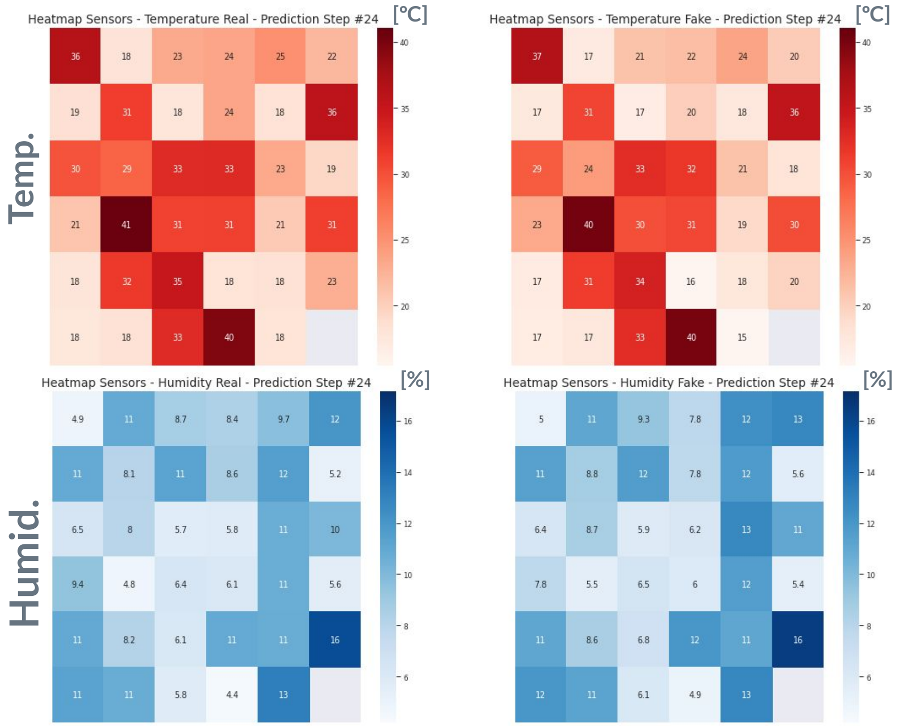
Scenario Generation: Random Exploration

- **Scenario Heatmap**

Each square represent a different sensor

- Predictions on the 24th step (4 hours)
- **The predicted scenarios are consistent!**

Real Data **Synthetic Data**



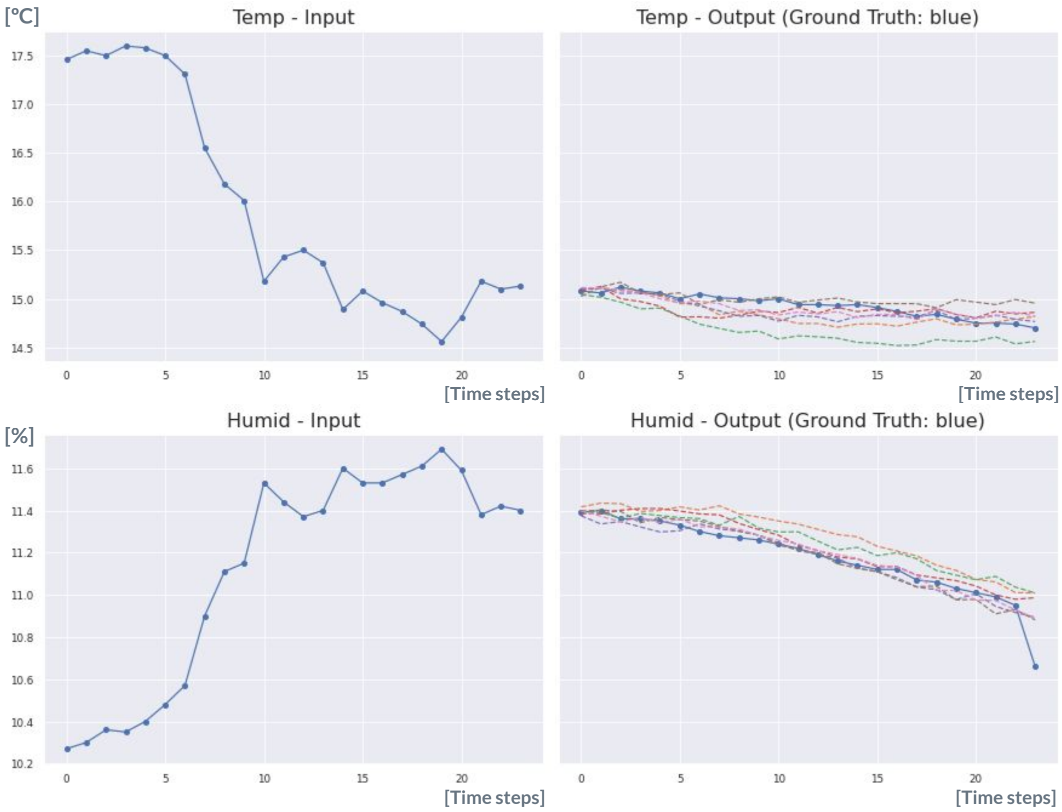
Scenario Generation: MH-GAN

- K = 20 (loops on each sample)
- 6 scenarios

Each color represent a different scenario

- 24 steps predictions concatenated (4 hours)

MH-GAN results on test set:
KL Divergence = 1.059 bits
RMSE Temperature = 1.001°C
RMSE Humidity = 0.661%

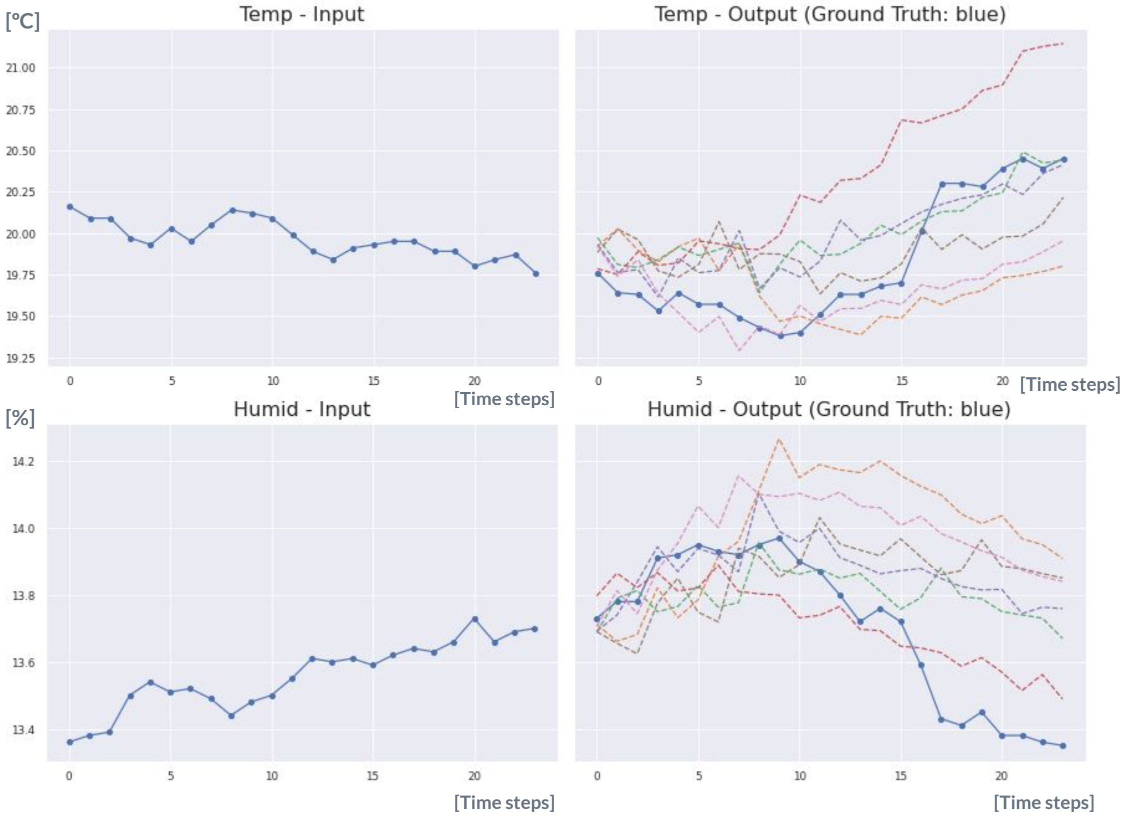


Scenario Generation: MH-GAN

- K = 20 (loops on each sample)
- 6 scenarios

Each color represent a different scenario

- 24 steps predictions concatenated (4 hours)



MH-GAN results on test set:
KL Divergence = 1.059 bits
RMSE Temperature = 1.001°C
RMSE Humidity = 0.661%

Scenario Generation: MH-GAN

▷ **Scenario Heatmaps**

Each square represent a different sensor

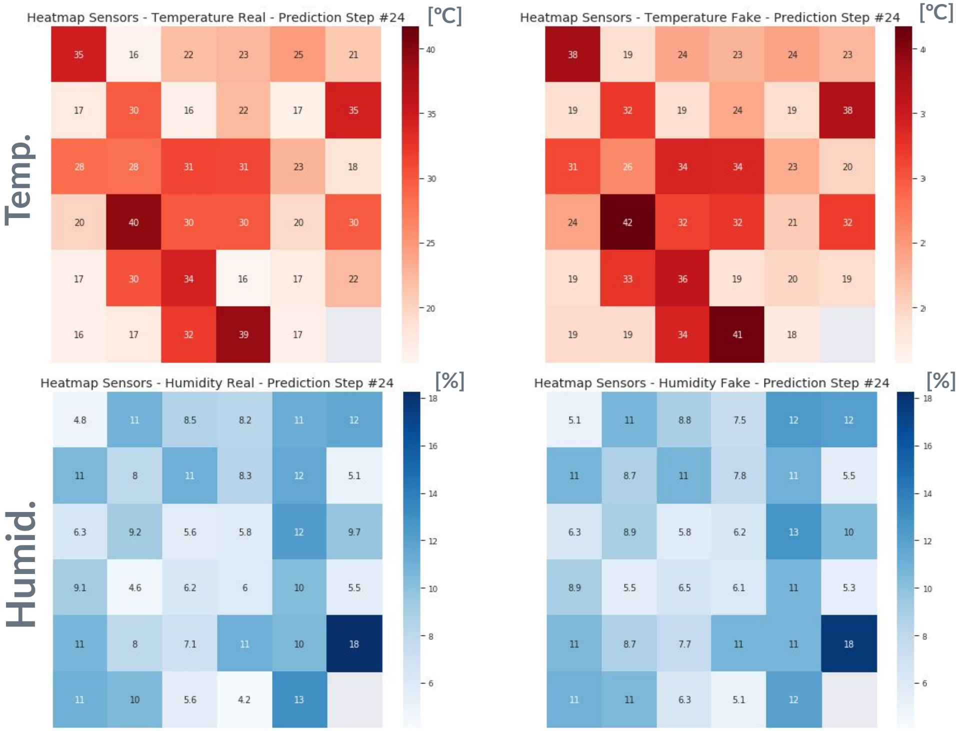
▷ Predictions on the 24th step (4 hours)

▷ **The predicted scenarios are consistent!**

MH-GAN:
Higher computational cost!
Seconds → Days

Real Data

Synthetic Data



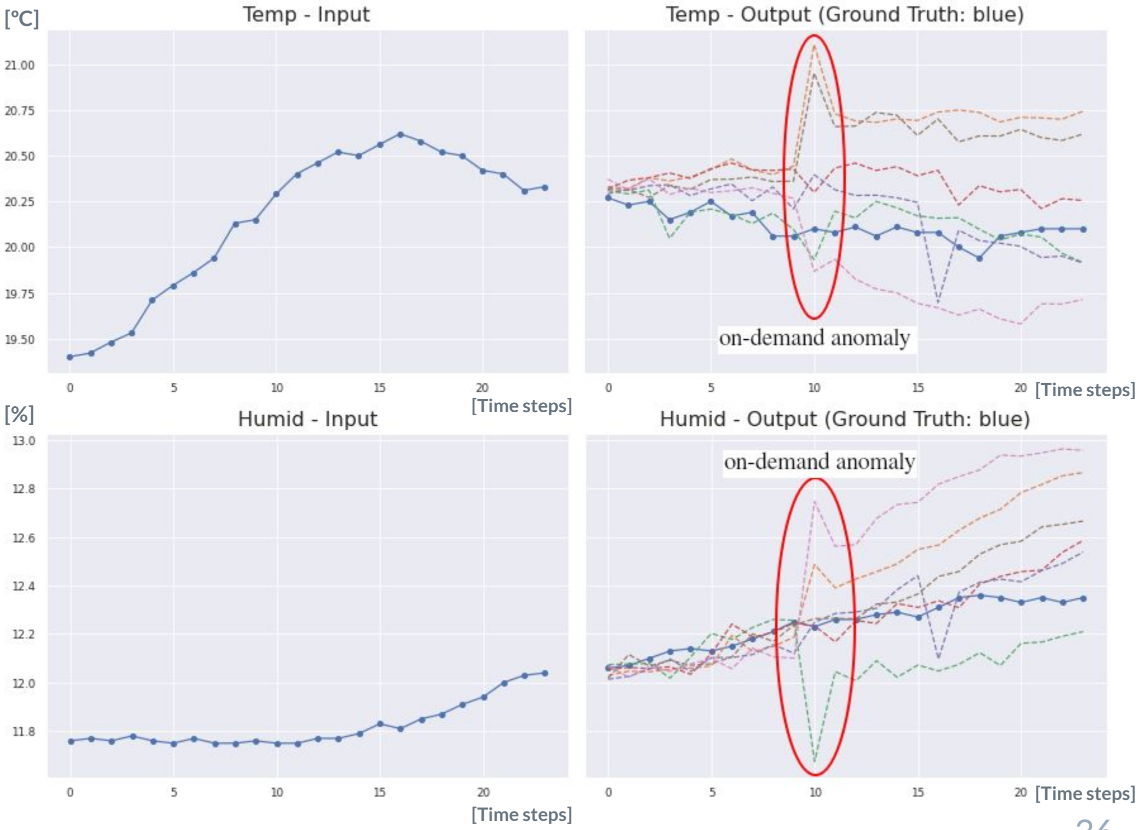
On-Demand Anomaly Generation

- 6 Scenarios

Each color represent a different scenario

- 24 steps predictions concatenated (4 hours)
- Introduce anomaly on the 10th step

Anomaly Generation Method:
Increase Standard Deviation of Gaussian latent space



On-Demand Anomaly Generation

- **Scenario Heatmaps**

Each square represent a different sensor

- Predictions on the 24th step (4 hours)
- Introducing an anomaly in the 10th step of the sensor #18
- The **correlation** between variables **is maintained!**

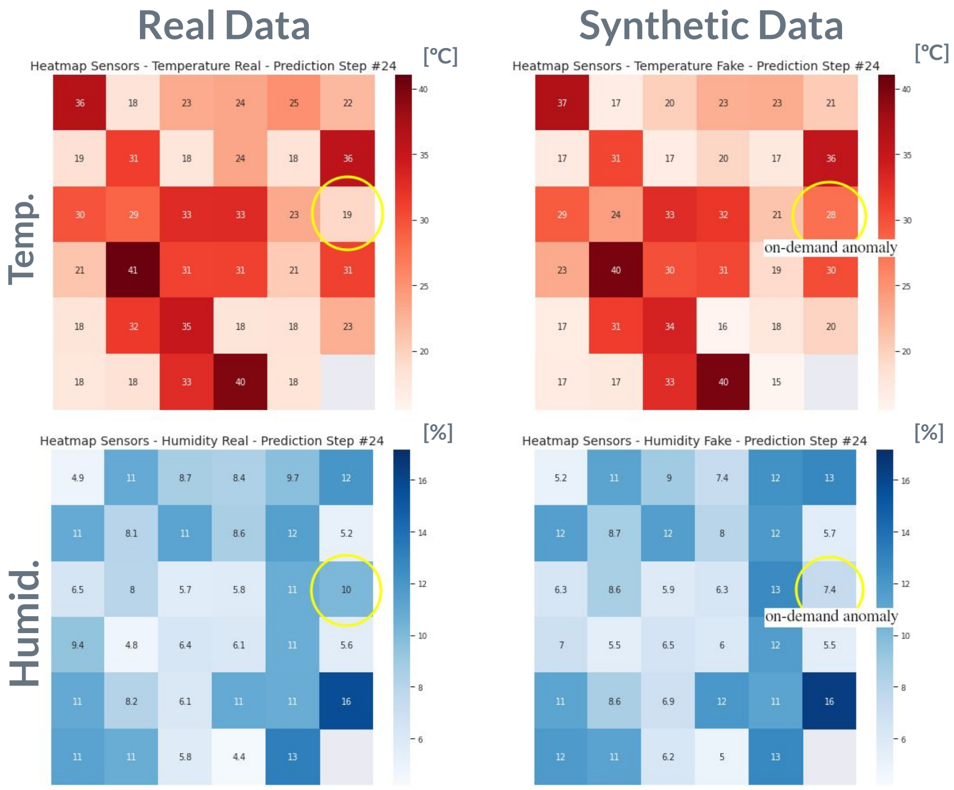
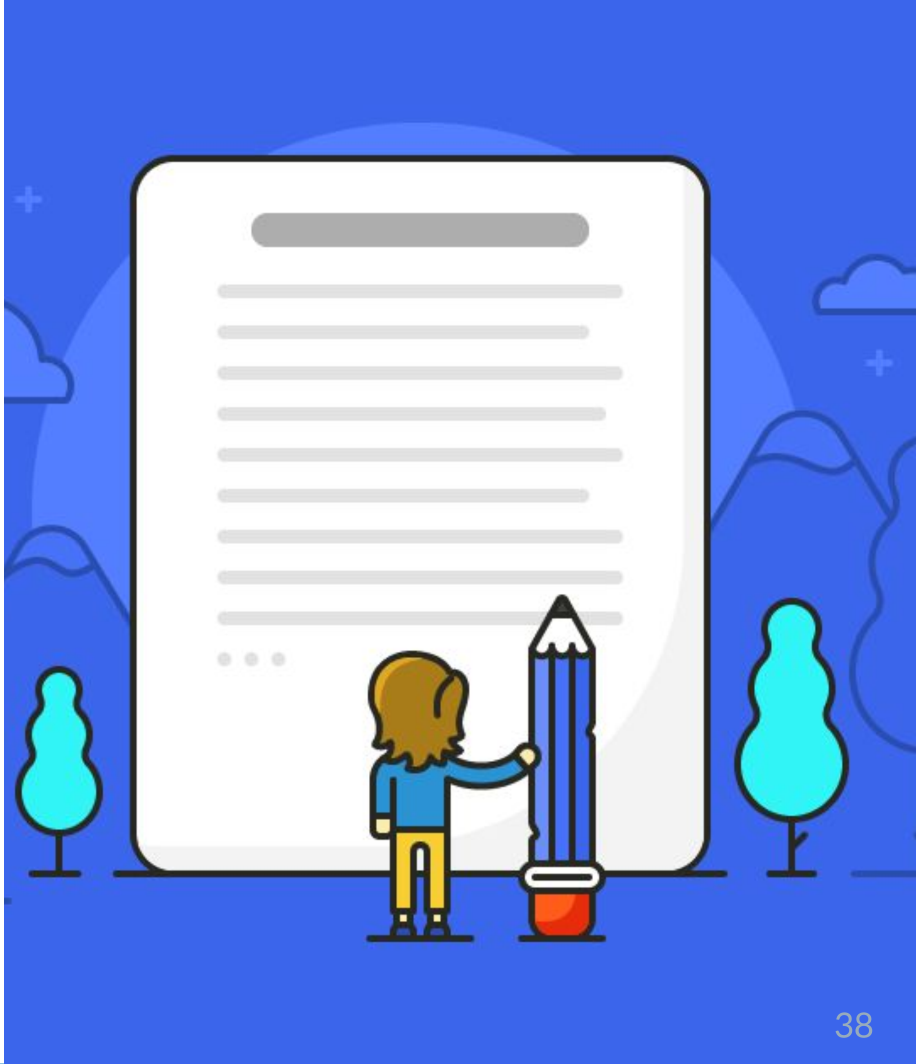
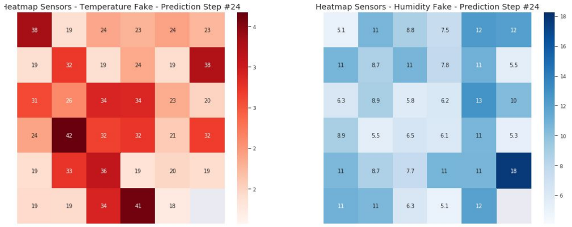


Table of Contents



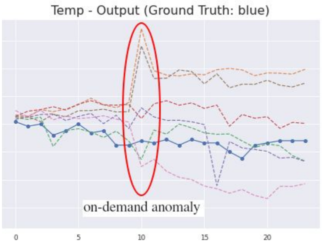
Conclusions

- **(Obj. 1) Scenario Generation**
  - (Obj. 2) Multi-variable scenario generation (Temperature & Humidity)
  - (Obj. 3) Introduce Categorical Variables → ID sensor
    - Test set:
      - KL Divergence = 1.133 bits
      - RMSE Temperature = 1.033 ºC
      - RMSE Humidity = 0.673 %



  - **MH-GAN**
    - Slightly better results, yet at a very high computational costs → Not necessary

- **(Obj. 4) Generate on-demand anomalous situations**
  - Data **variability** without additional effort
  - Without endangering electronic equipment integrity



- Enable **better optimization of Data Centers** → **More sustainable and greener future**

Future Research Work

Analyze Scalability of Multi-Variables Datasets

-

Further Study on the Usefulness of the Results

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

-

Hyperparameter Optimization

Add Supplementary Information at the Inputs

E.g., Freq., ARIMA, Date 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?

Jaime Pérez Sánchez

jaime.perez.sanchez@alumnos.upm.es

www.linkedin.com/in/jaime-perez-sanchez/

Thanks!

Any questions?

Jaime Pérez Sánchez

jaime.perez.sanchez@alumnos.upm.es

www.linkedin.com/in/jaime-perez-sanchez/

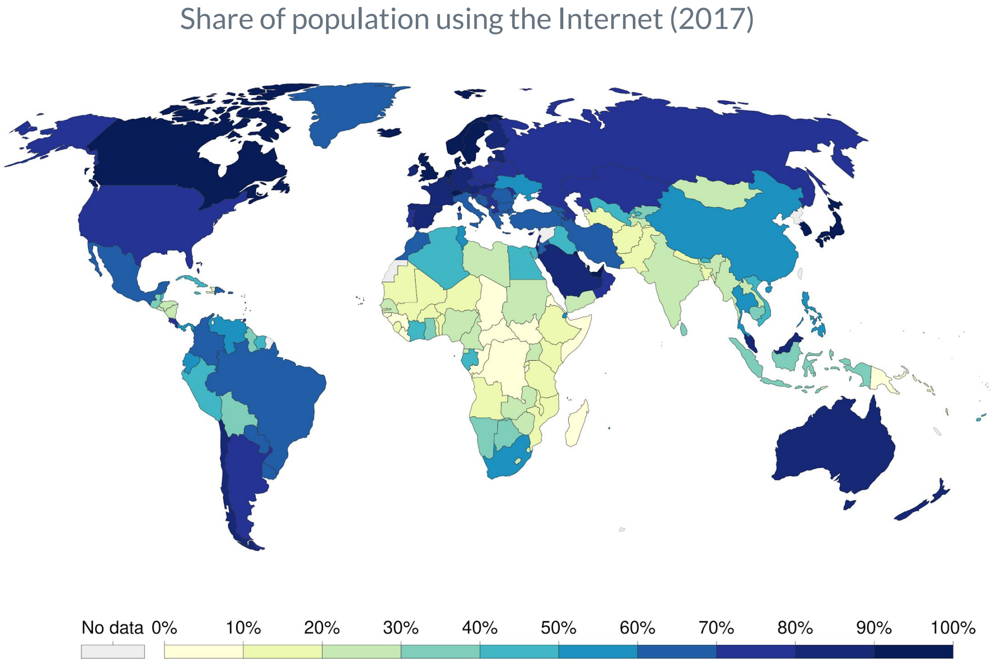
BACKUP

Digital Economy & Society

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

European Citizens Using Internet Daily	
2014	48%
2018	73%

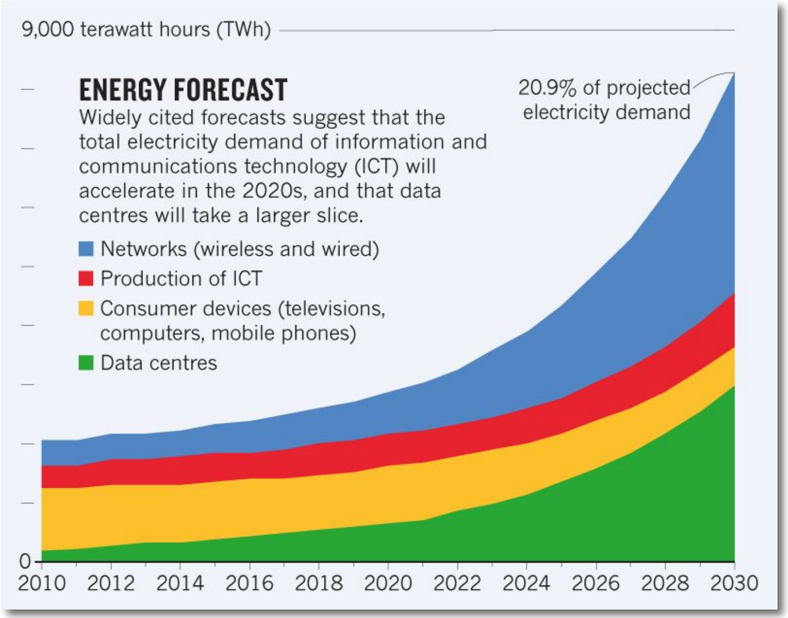
Daily Hours Spent on Digital Media by U.S. Citizens	
2013	<5h
2018	>6h



Source: World Bank Data (2017)

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)



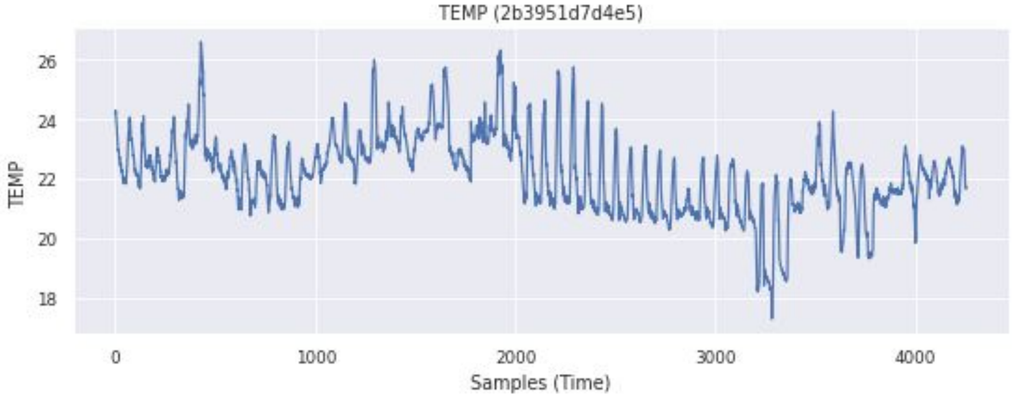
Cloud Computing Paradigm

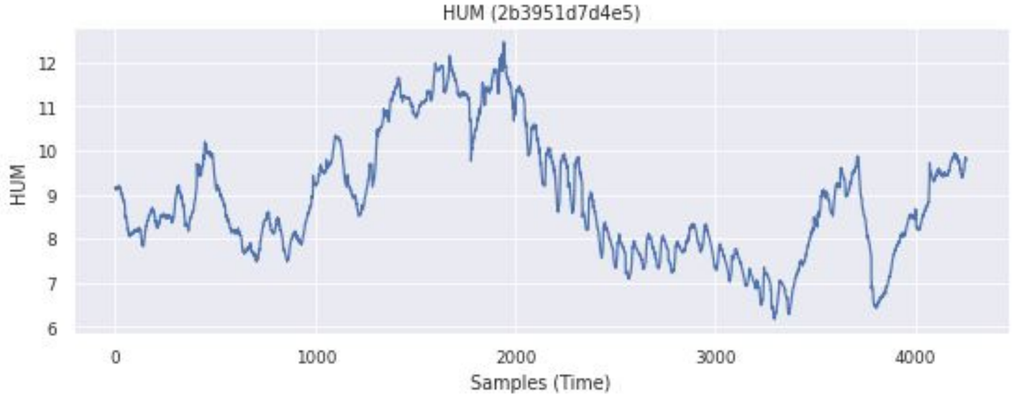
- Progress on Data Center **energy efficiency**
  - Energy use per computation has dropped ~400%
- **Is it enough?**
  - Global Power Usage Effectiveness of **1.67** (ideal = 1) in 2019
  - **Cooling energy** expenses represent, on average, up to **40%** of the total bill [2]
  - In the next **3 or 4 years** the number of compute instances will **double** again

We urgently need better optimization in Data Centers!

However, Data Centers are very challenging to optimize...

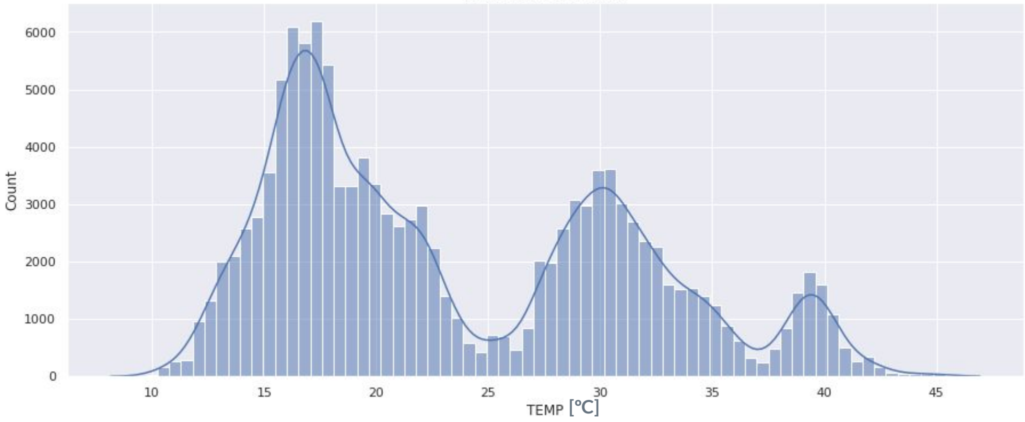
Example of Data Gathered by One Sensor



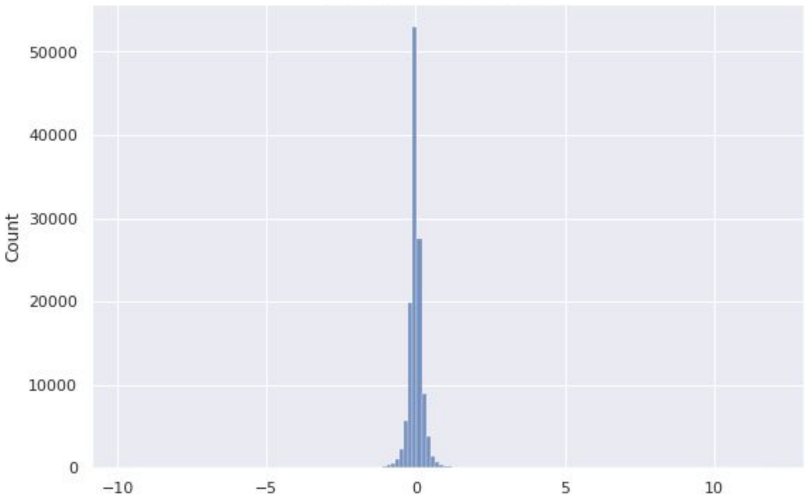


Exploratory Data Analysis: Temperature

Temperature Histogram



Temperature Derivative Histogram

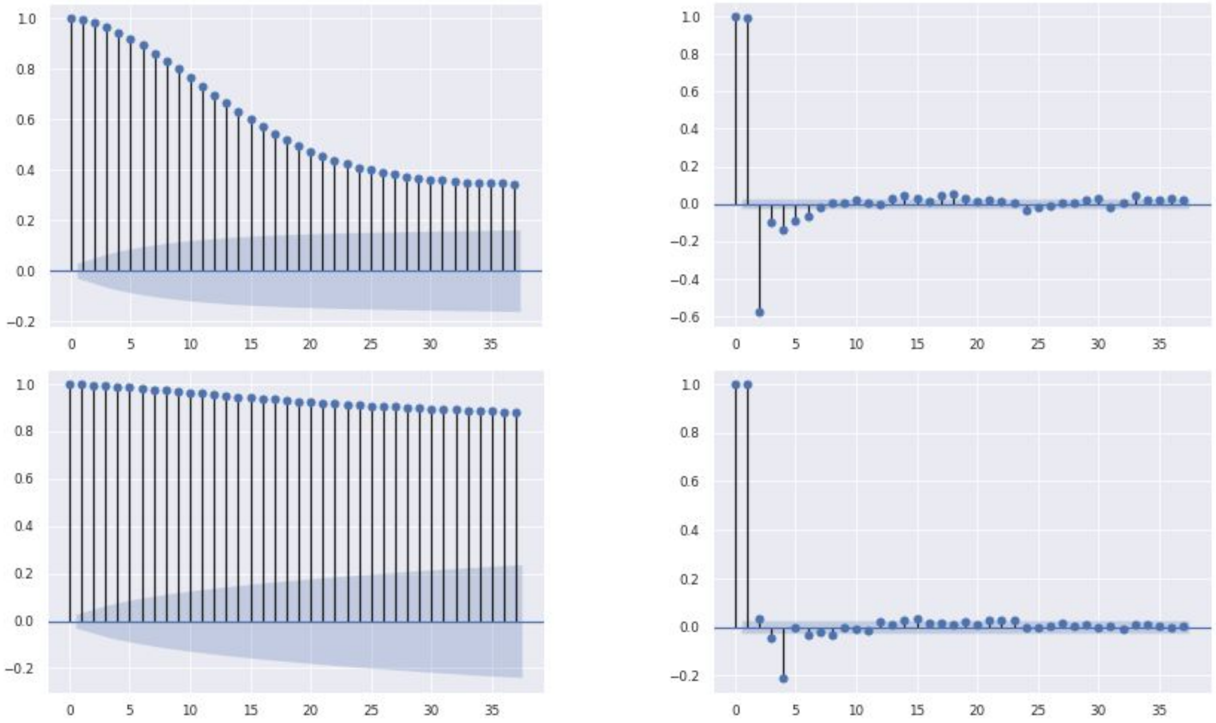


Max. Increase.: 11.91°C
Max. Decrease: -9.79°C

Exploratory Data Analysis: Temperature

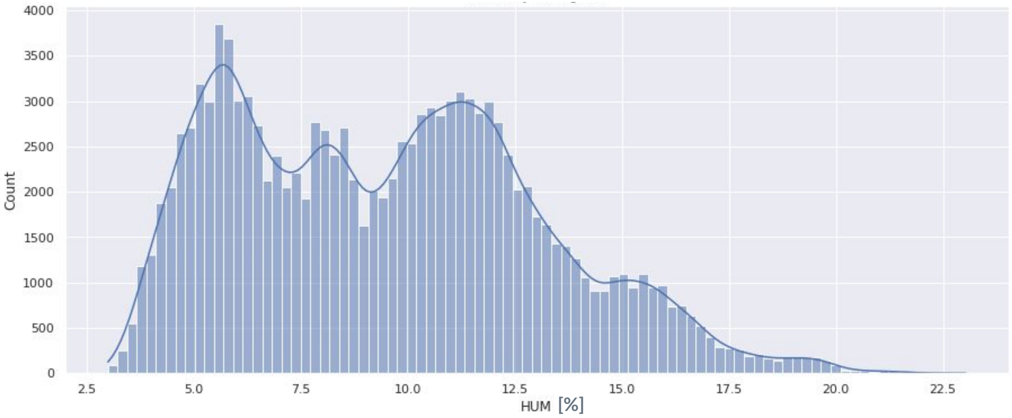
Autocorrelation

Partial Autocorrelation

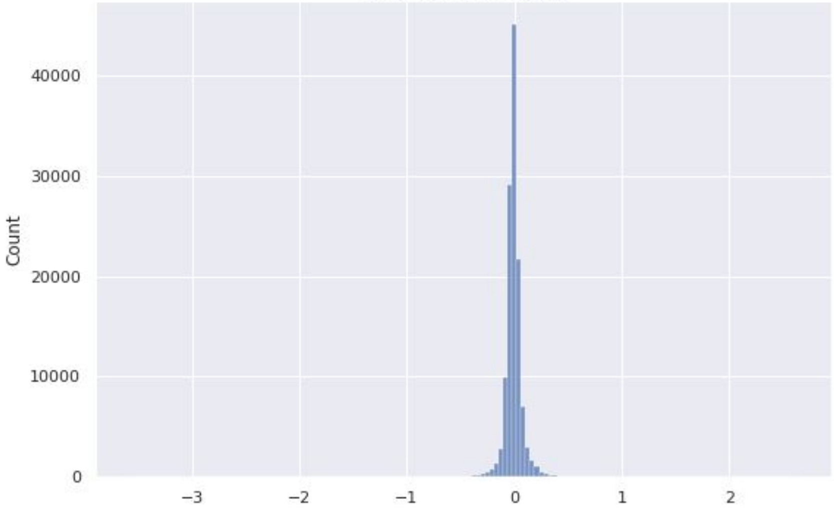


Exploratory Data Analysis: Relative Humidity

Relative Humidity Histogram

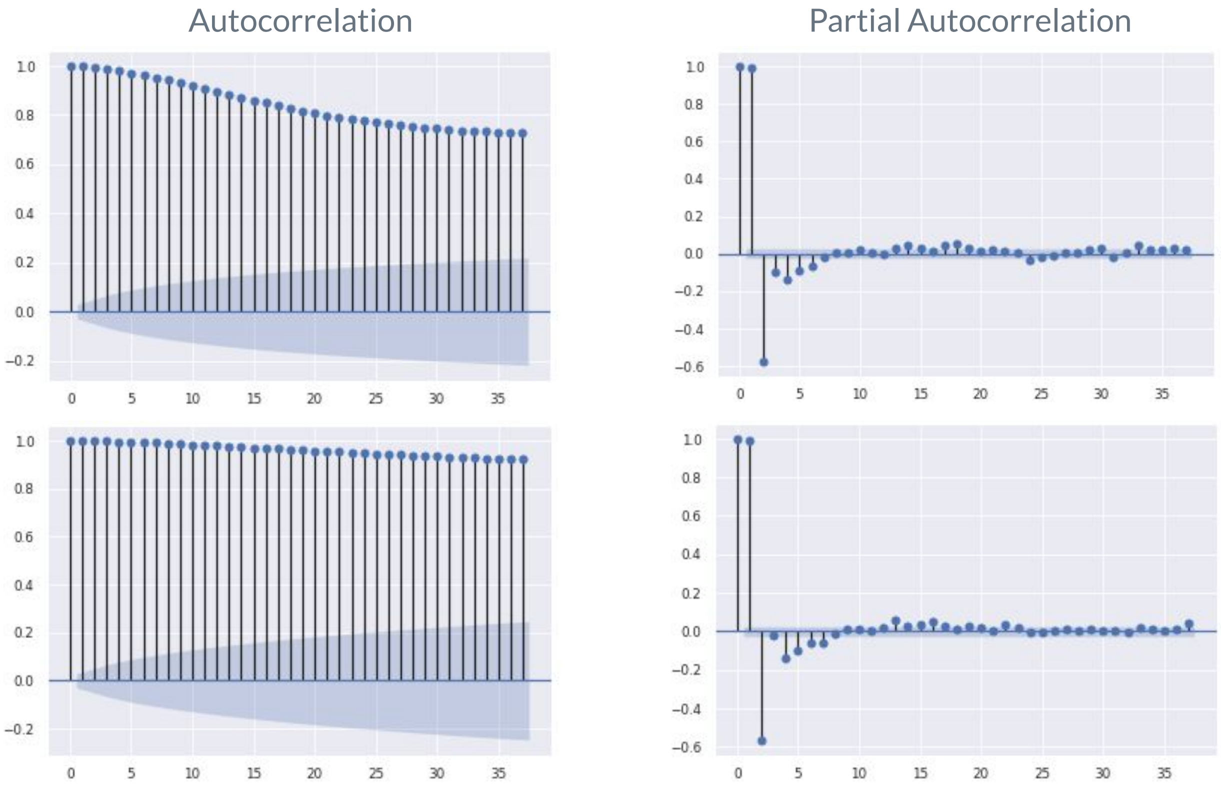


Relative Humidity Derivative Histogram



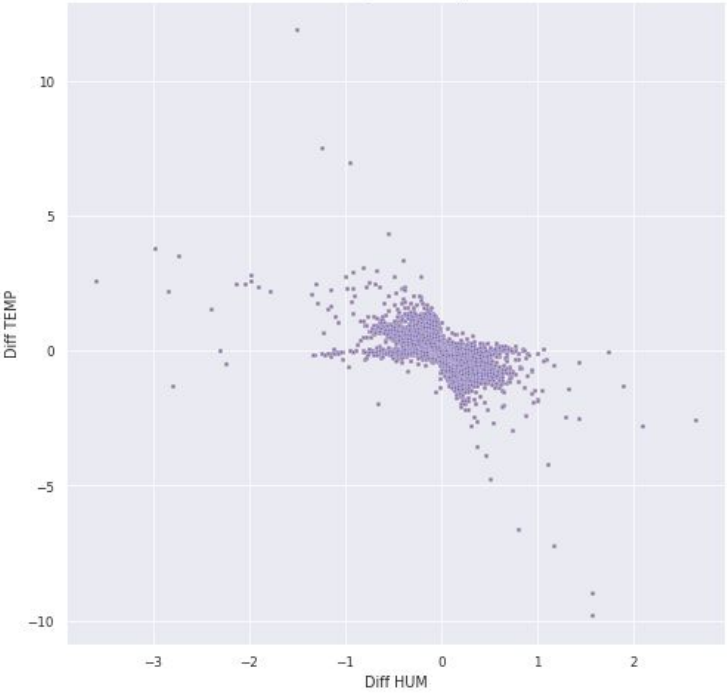
Max. Increase.: 2.64%
Max. Decrease: -3.59%

Exploratory Data Analysis: Relative Humidity

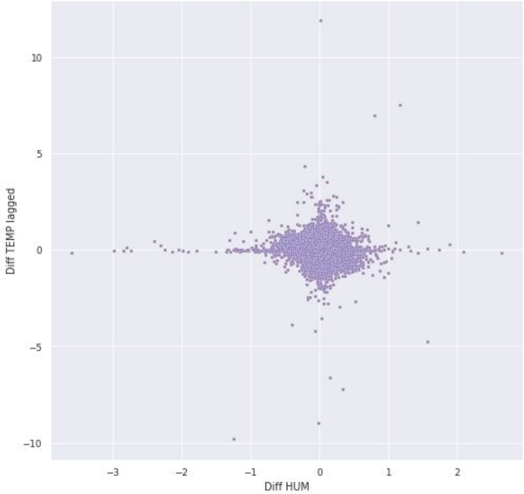


Exploratory Data Analysis: Correlations

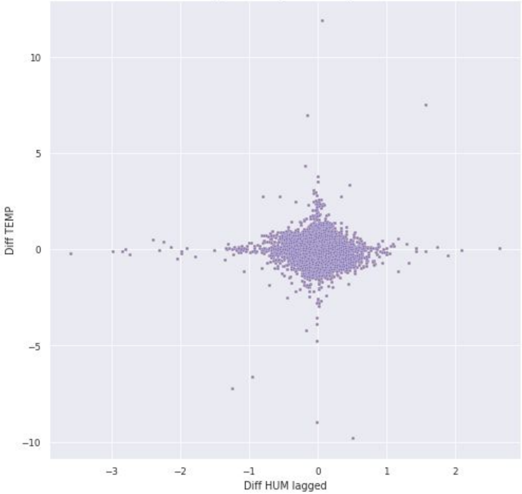
Rel. Humidity Derivative vs. Temperature Derivative



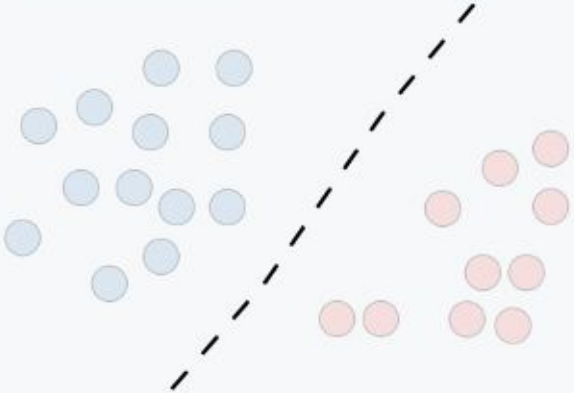
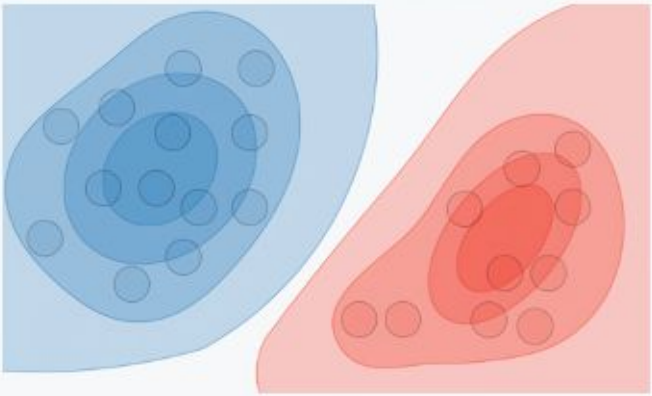
Rel. Humidity Derivative vs. Temperature Derivative
With lagged Temperature



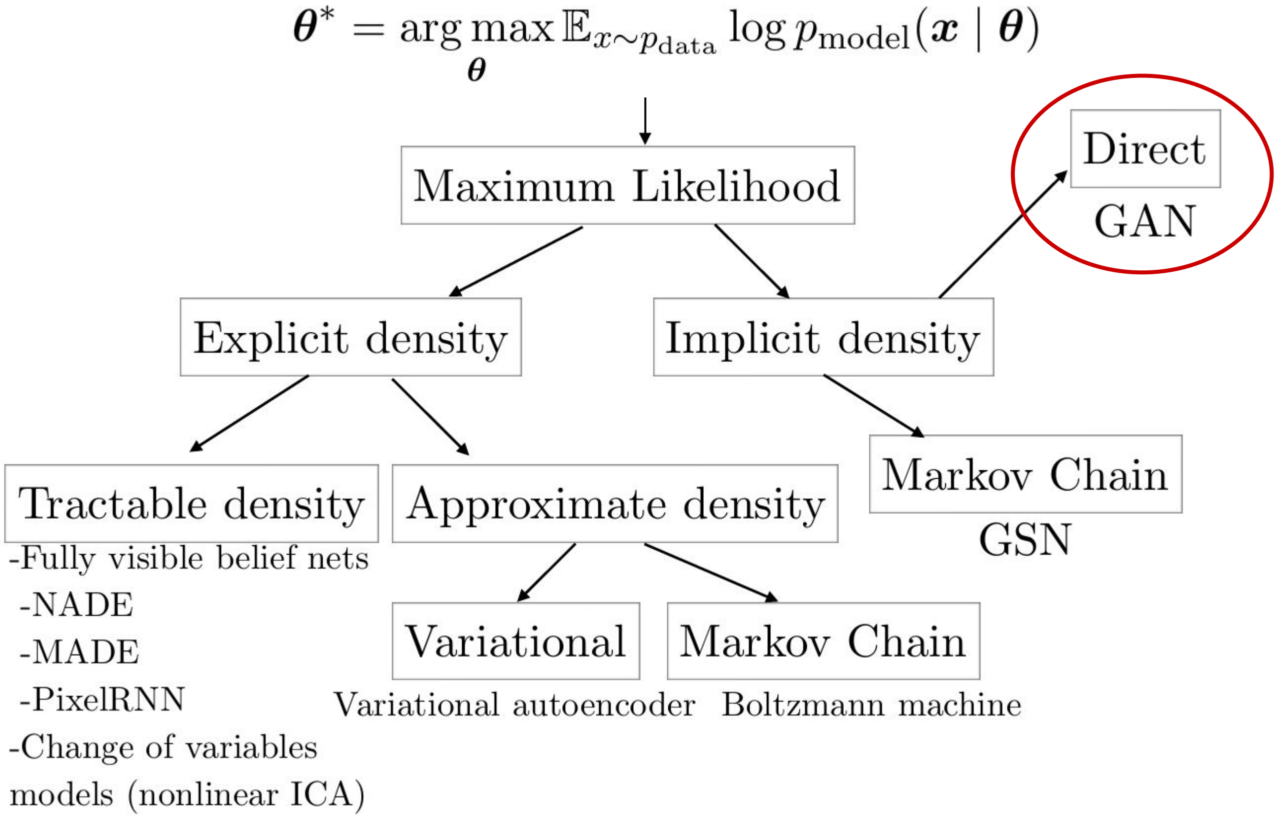
Rel. Humidity Derivative vs. Temperature Derivative
With lagged Humidity



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

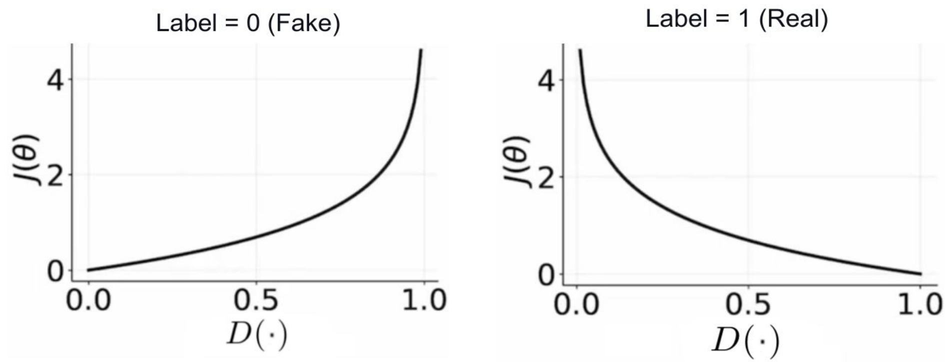


GAN Training

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

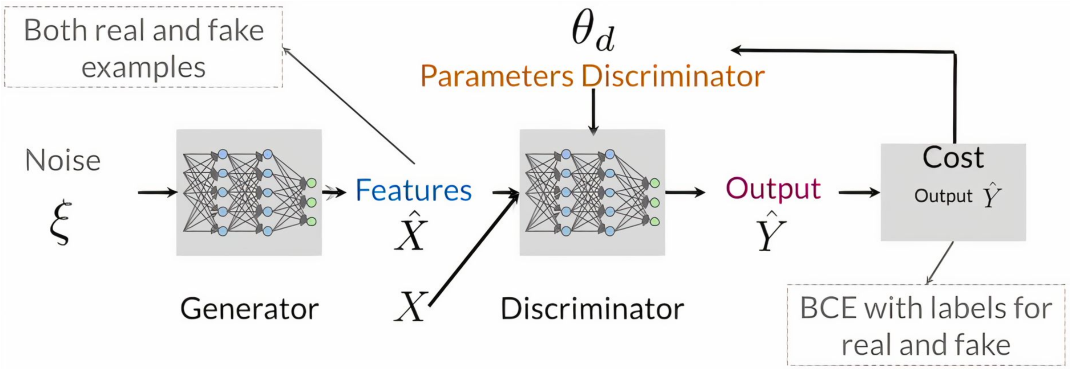
$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log \left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log D\left(G(\boldsymbol{z})\right)$$

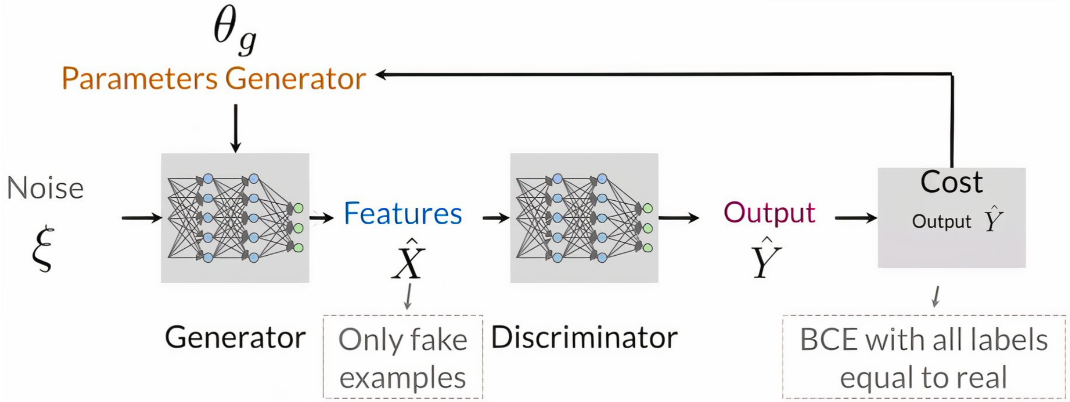


GAN Training

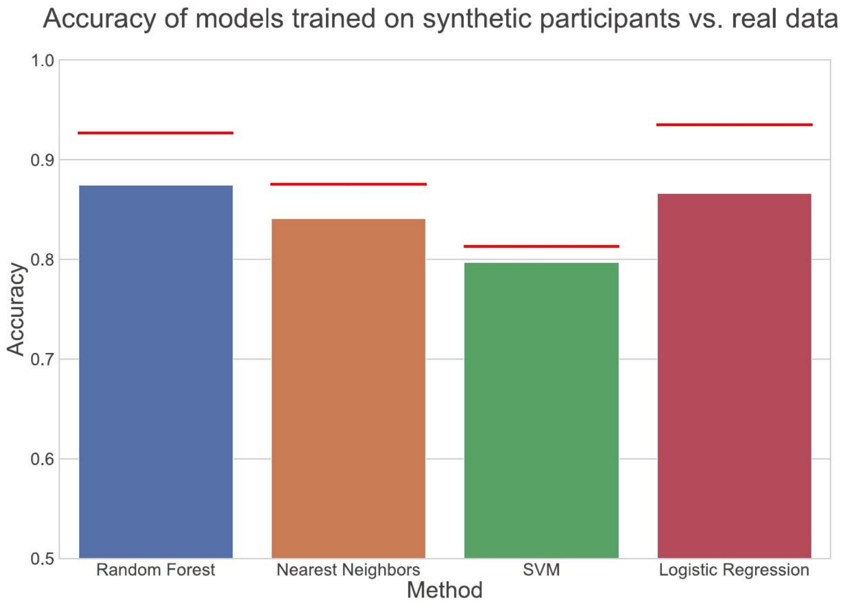
- Training Discriminator



- Training Generator



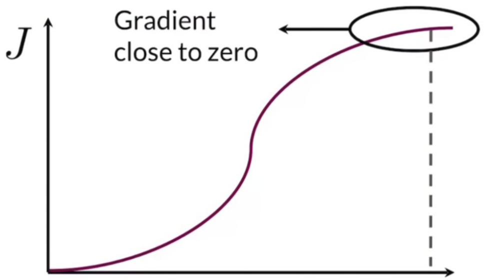
Synthetic GAN Data



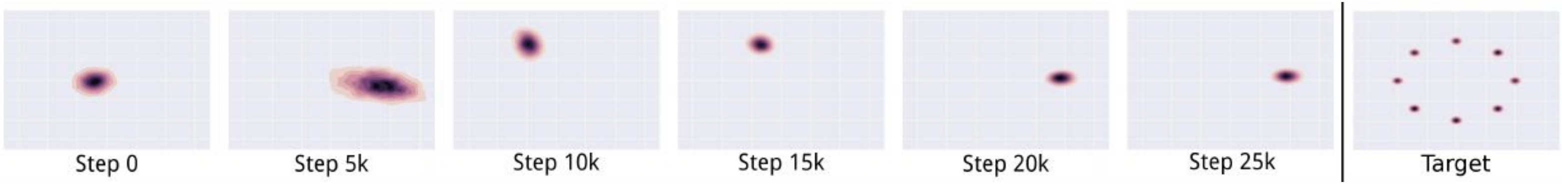
Accuracy of models trained on synthetic GAN data vs. real data.

GAN Training Problems

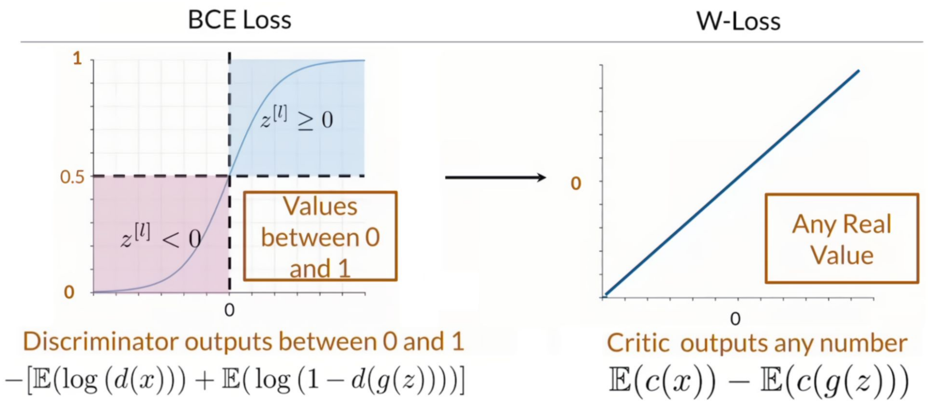
- Vanishing Gradients problem



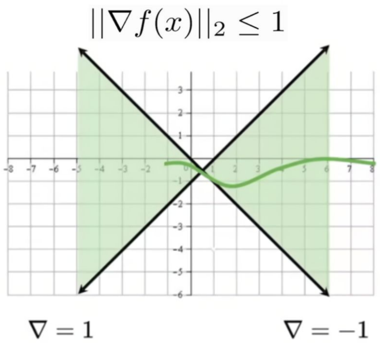
- Model Collapse problem



GAN Training Improvements: WGAN-GP



- 1-Lipschitz continuity condition



- Cost Function

$$L_d = \underbrace{\mathbb{E}\left[D(x)\right] - \mathbb{E}[D(G(z))]}_{\text{Original critic loss}} + \underbrace{\lambda \cdot \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}}(D(\hat{x})\|_2 - 1)^2]}_{\text{Gradient penalty}}$$

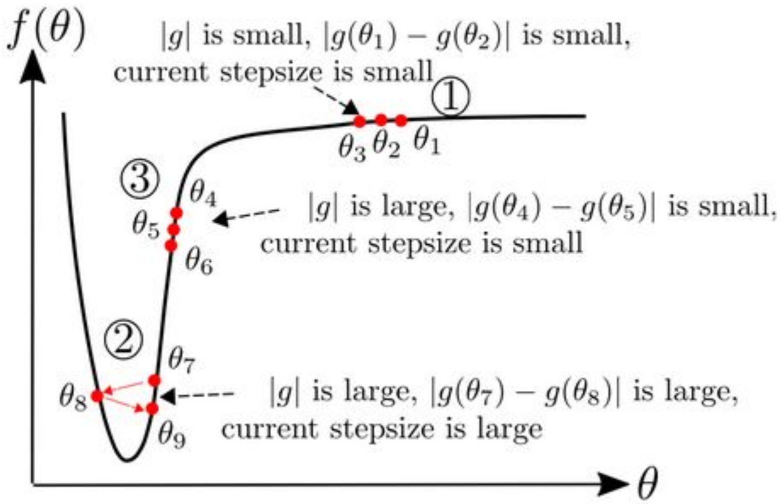
$$\text{with } \hat{x} = \epsilon x + (1-\epsilon)G(z)$$

$$\text{and } \epsilon \sim U[0,1]$$

GAN Training Optimizer: AdaBelief

Why AdaBelief is better?

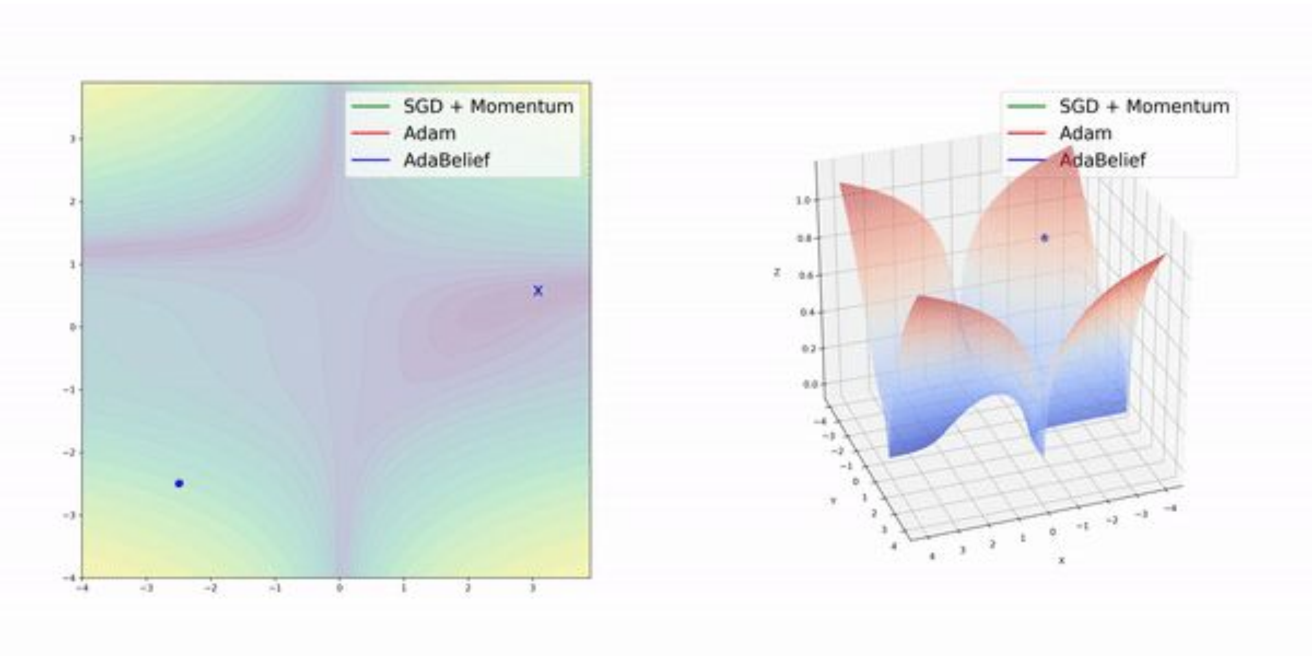
AdaBelief considers the curvature of loss function



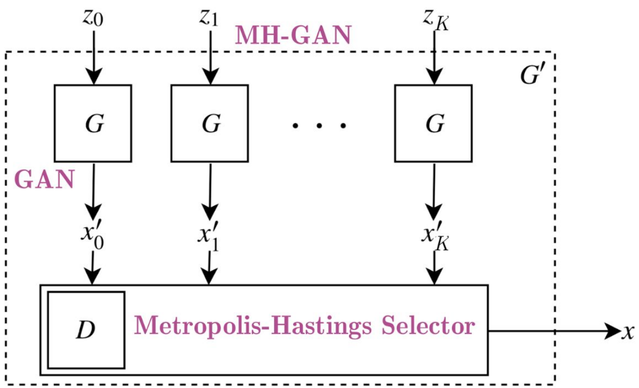
An ideal optimizer considers curva- ture of the loss function, instead of taking a large (small) step where the gradient is large (small). In region 3 , we demonstrate AdaBelief's advantage over Adam in the "large gradient, small curvature" case. In this case, $|g_t|$ and v_t are large, but $|g_t - g_{t-1}|$ and $|s_t|$ are small; this could happen because of a small learning rate α. In this case, an ideal optimizer should increase its stepsize. SGD uses a large stepsize (~ $\alpha|g_t|$); in Adam, the denominator $\sqrt{v_t}$ is large, hence the stepsize is small; in AdaBelief, denominator $\sqrt{s_t}$ is small, hence the stepsize is large as in an ideal optimizer.

AdaBelief considers the sign of gradient in denominator

GAN Training Optimizer: AdaBelief



Data Generation Improvement: MH-GAN



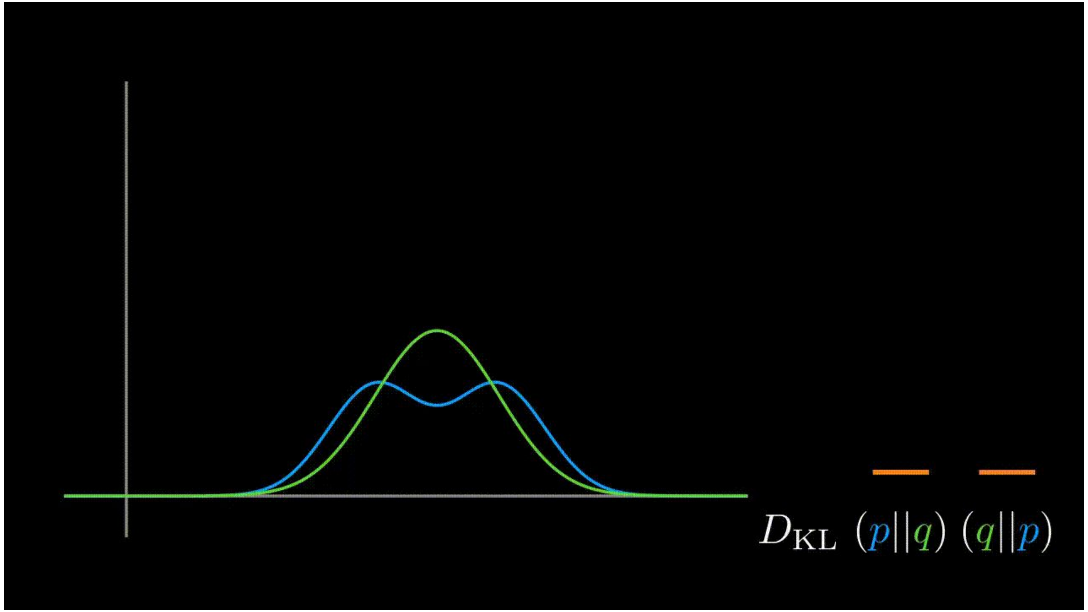
The key feature of MH-GAN is that the acceptance probability can be computed just with the ratio of probability densities p_D/p_G, which is readily available from the output of the GAN's discriminator! Starting with x_k as the current sample, a new sample x' is accepted over the current sample x_k with probability α:

$$\frac{p_D}{p_G} = \frac{1}{D^{-1} - 1} \implies \alpha(\mathbf{x}', \mathbf{x}_k) = \min\left(1, \frac{D(\mathbf{x}_k)^{-1} - 1}{D(\mathbf{x}')^{-1} - 1}\right)$$

where D is the discriminator score $D(\mathbf{x}) = \frac{p_D(\mathbf{x})}{p_D(\mathbf{x}) + p_G(\mathbf{x})}$.

K is a hyperparameter and can be chosen based on speed/fidelity trade-offs. It can be shown that for a perfect discriminator and as $K \to \infty$, this recovers the real data distribution.

Kullback-Leibler Divergence



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 Network 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

Tuned Hyperparameters:

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

Best Results on Validation Set									
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**

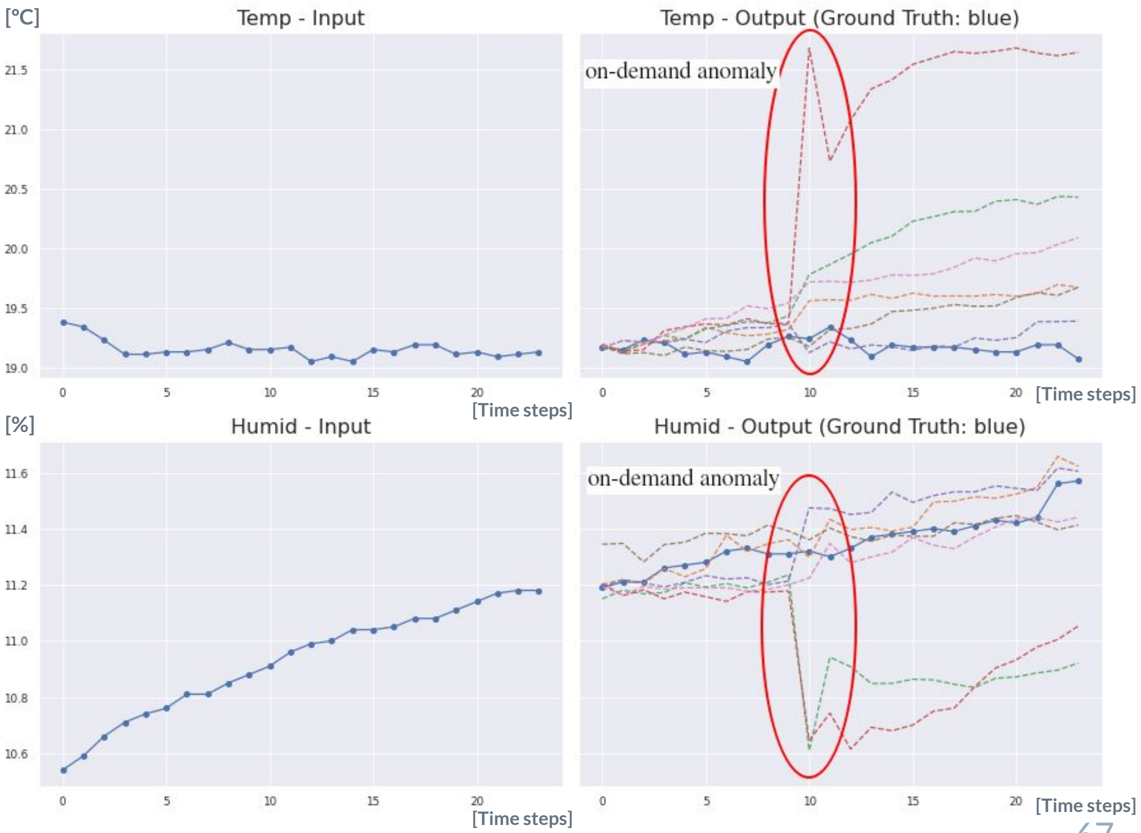
On-Demand Anomaly Generation

- 6 Scenarios

Each color represent a different scenario

- 24 steps predictions concatenated (4 hours)
- Introduce anomaly on the 10th step

Anomaly Generation Method: Increase Standard Deviation of Gaussian latent space



Conclusions

This work extends **synthetic time-series data** applications, enabling to use **categorical variables** and **multi-variable scenario generation**.

Methodology **validated with real data** gathered from a Data Center in operation. Obtaining a **KL divergence** of **1.432 bits** and an **RMSE** accuracy error of **0.988ºC** for temperature and **0.661%** for humidity, on a validation set of data.

The **MH-GAN** sampling method produce slightly better results, yet at a very high computational cost. We do not consider it necessary in our architecture.

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

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 need better metrics to measure “realism” of the synthetic data.

- **Human Supervision is Needed**

  The limitations on the training process and the accessible data, 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.

Backup

Backup