# Stay Report

To: Geosphere Austria, Vienna, Austria
Period: April 20th – May 18th, 2025
Topic: Work on Machine Learning Methods for Radiation Post-processing
Supervisors: Mag. Alexander Kann and Mag. Dr. Irene Schicker
Researcher: Iris Odak (IrOd)

# Machine Learning Methods for Radiation Point-based Post-processing

## 1. Introduction

I stayed at GeoSphere Austria for four weeks, learning and working on Machine Learning (ML) methods that can be used for the post-processing of several different meteorological parameters. Currently, there are only a few post-processing methods operationally used at DHMZ (e.g., neighborhood, analogs). Several other methods that were drafted during the last stay are also being tested, especially LSTM and RF (random forest).

With the growing demand for accurate solar energy predictions, there's a need for advanced ML techniques that can analyze and predict weather variables with high precision. These predictions are crucial for optimizing renewable energy production, especially for solar power generation, which heavily depends on weather conditions such as temperature, cloud cover, and solar radiation.

Even though former methods such as basic implementation of LSTM did show certain potential, especially after the fraction of daily/night hours is included as additional predictor, the result was sonly comparable to operational ALADIN-HR deterministic model. Thus, at this point, no post-processing technique is implemented at DHMZ to provide a better forecast for solar radiation or solar power prediction. Additionally, the increasing demand for post-processed visibility parameter is also becoming a factor to go for a more complex model and use more than station data. The natural next step is to examine current achievements in literature and consider available satellite data in order to develop a reasonable nowcasting model.

## 2. Literature overview

There are many available papers on the subject, some of which I found particularly interesting.

Many older or general-purpose solar energy simulation studies relied on standard atmosphere models (e.g. mid-latitude summer/winter profiles), synthetic weather generators or TRYs (test reference years) and idealized or constant turbidity (e.g., T = 2), showing certain limitations.

| Model | Main idea | Weakness |
|---|---|---|
| **Liu & Jordan** | Simple linear fit | Overestimates Id in clear skies |
| **Erbs** | Piecewise linear (3 segments) | Discontinuous, over-smooth |
| **Reindl** | Polynomial regression + clearness index + sun altitude | More accurate |
| **Hay & Davies** | Separates diffuse into isotropic + circumsolar parts | Needs beam/diffuse ratio (AI) |

**Gassel** emphasizes that these assumptions do not reflect real conditions, especially in cloud-prone, aerosol-heavy, or low-winter-sun regions like central Europe. Gassel used measured turbidity factors from the German Weather Service (e.g., Chemnitz 1994 dataset); analysed temporal variations, seasonal variation in turbidity (lower in winter, higher in summer), daily cycles (T lower in morning, rising through day), and compared synthetic vs. measured radiation components (global, direct, diffuse). On a clear day the results show inversion error typically < 3% and on mixed sky error ~5–8% but still much better than assuming typical K_T.

**Bird–Hulstrom** Model Equations move toward a more physical model, combining atmospheric transmission (scattering, absorption) and sun angle geometry. Next, custom-built statistical models that relate measured solar radiation (global and diffuse) to locally available meteorological parameters, using **multiple linear regression** seems to be very accurate and flexible solution, but needs a decent meteorological data set.

Next breakthrough seems to be considering satellite-derived solar radiation for intra-hour and intra-day applications. The idea is to evaluate how accurate satellite-based models are at estimating surface solar radiation (SSR) for short-term forecasting applications. Solar Short Term Ensemble Prediction System (**SolarSTEPS;** Carpentieri et al., 2023) introduces a probabilistic ensemble forecast based on satellite-derived clear-sky index (CSI) maps using Cloud motion vectors (CMVs) from Lucas-Kanade optical flow, spatial scale decomposition via Fourier transform, cascade-specific autoregressive (AR) models to simulate cloud evolution at different spatial scales and noise generation that respects spatial autocorrelation and local variability. The advantage of such applications is having forecasts for entire satellite images (not just points) that outperform advection-only models by 9.3% improvement in CRPS and ~45-minute longer forecast horizon at the same skill level. This paper uses real satellite data (**HelioMont**) and approach is tested in Alpine conditions. The paper shows that this approach combines deterministic physical modelling (via optical flow, AR(1)) with stochastic perturbations that respect spatial and temporal structure, providing a robust baseline for probabilistic solar forecasting.

Finally, in 2025, Carpentieri et al. demonstrate that **SHADECast** is, to the best of their knowledge, the first uncertainty-aware, physics-inspired, satellite-based regional-scale forecast model for intraday SSI forecasts. They show that SHADECast produces skilful, sharp and reliable solar forecasts without blurring under variable weather conditions due to innovative, physics motivated splitting of the task at hand, also providing a comparison to SolarSTEPS (Figure 1). This paper was a main reason to choose SHADEcast as the most perspective one to try to understand and implement during my stay.

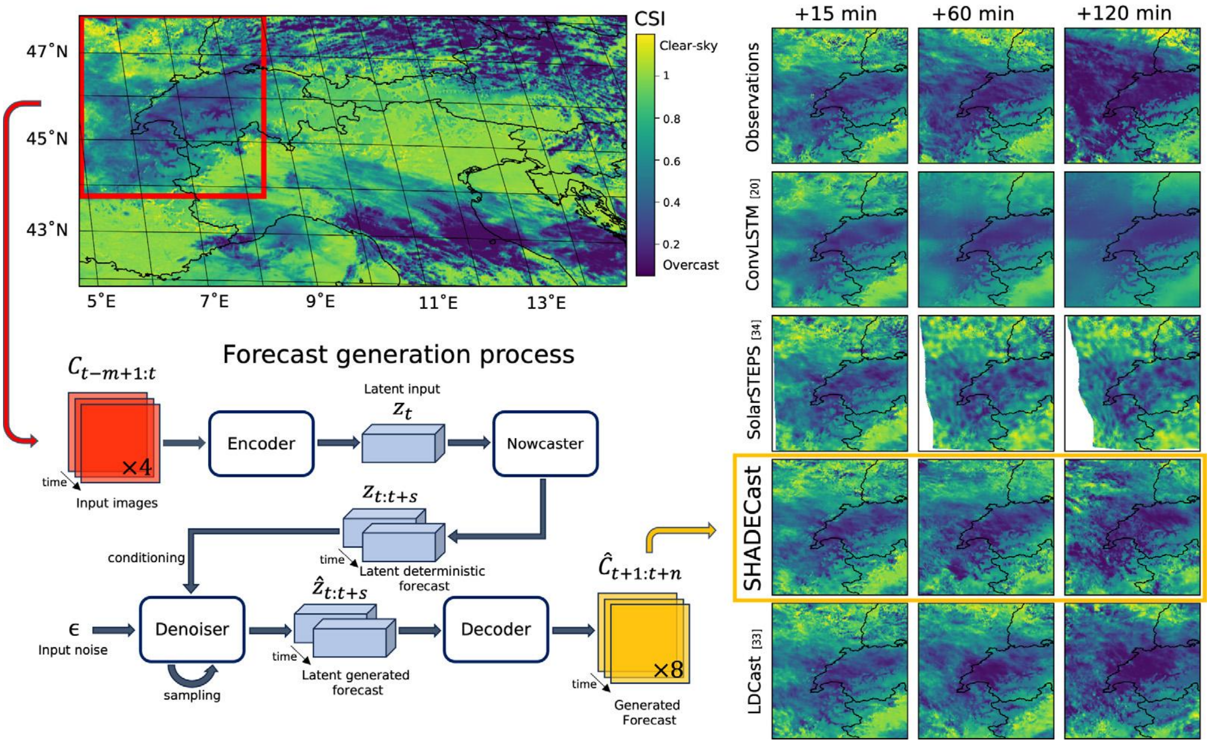

*Fig. 1. From Carpentieri et al. (2025): SHADECast forecast generation pipeline. Upper left panel: Example CSI field of 24 Feb. 2016 at 11:45 UTC. The red box highlights the region forecasted in the right panel. Lower left panel: SHADECast forecast generation pipeline. The input CSI fields ($C_{t-m+1:t}$) are fed to the encoder, which projects the image sequence to the latent space, obtaining $z_t$. Then, the deterministic nowcaster forecasts the future latent representation of the CSI fields ($z_{t+1:t+s}$), where $s$ is the lead time in the latent space, which can differ from $n$ due to data compression. The latent forecast is, then, fed to the denoiser together with Gaussian noise $\epsilon$. The pseudo linear multi-step (PLMS) sampler employs the denoiser to generate an ensemble member. The decoder finally decompresses the latent ensemble forecast, obtaining $C_{t+1:t+n}$. Right panel: Forecasts made by SHADECast (yellow box) and benchmark models for lead times up to 120 min. For SHADECast, LDCast and SolarSTEPS the ensemble member chosen is the one with the lowest average root mean squared error (RMSE). The first row shows the satellite-derived CSI fields.*

## 3. SHADECast model

SHADECast is structured into the following key modules, which are implemented as PyTorch modules and combined into a training and inference pipeline:

- Variational Autoencoder (VAE): Compresses CSI input into latent space.
- AFNO Nowcaster: Deterministic forecast module operating in latent space.
- Latent Diffusion Model: Stochastic generator producing ensemble forecasts in latent space.
- Decoder: Transforms latent forecasts back into full-resolution CSI maps.

The goal is to perform regional, short-term, probabilistic forecasting of surface solar irradiance, expressed via the clear-sky index (CSI).

**Code Structure Overview**

```
models/
|— vae.py                    VAE with encode(), decode(), reparameterize()
|— afno_nowcaster.py         Nowcaster with AFNO blocks and temporal transformer
|— latent_diffusion.py       U-Net denoiser with AFNO cross attention
datasets/
|— csi_dataset.py            Loads CSI tensors from SARAH-3 files
```

| | |
|---|---|
| train.py | Trains VAE + nowcaster or denoiser |
| sample.py | Generates ensemble forecasts from trained model |
| configs/ | YAML configs for architecture and training control |
| utils/ | Metrics: CRPS, PINAW, PICP, etc. |

## 4. Data

SSR (Surface Solar Radiation), also called Global horizontal irradiance (GHI), Surface incoming shortwave radiation, Shortwave downward radiation – it is: Total solar energy flux received on a horizontal surface from the sky (both direct + diffuse), under all-sky conditions. SSR is the primary input for PV systems. The variability of SSR, due to clouds, aerosols, and terrain, leads to intermittency in PV output.

Satellite data measures Spectral channels, e.g., 3 visible (VIS), 8 thermal infrared (IR), 1 high-resolution VIS. In other words, it measures top-of-atmosphere (TOA) reflectance, brightness temperature (in IR) and radiance in visible channels, latter indicating cloud albedo. However, this data does not contain radiation at the ground level — that's what must be modelled using physics. This is where models like Heliosat, HelioMont, libRadtran, and cloud detection algorithms come in.

For example, libRadtran is an open-source radiative transfer software library used to simulate solar radiation interaction with the atmosphere. It is a widely used library for clear-sky irradiance estimation.

In both SHADECast (Carpentieri et al., 2023) and SolarSTEPS (Carpentieri et al., 2025), the authors used clear-sky index (CSI) maps derived from Meteosat Second Generation (MSG) SEVIRI imagery, processed using the HelioMont algorithm developed by MeteoSwiss. This approach is based on the Heliosat family of methods and provides high-resolution CSI data tailored to complex terrain, especially in Alpine regions. The SARAH (Surface Solar Radiation Data Set – Heliosat) datasets are satellite-derived solar irradiance products developed by CM SAF (EUMETSAT). Since dataset used in paper was not easaly accessed, we decided to focus on SARAH 3 data, which is the updated version of SARAH-2. Newer MSG reprocessing was reasoning to use SARAH 3 datasets, it shows much more potential than, for example, SARAH 2.

## 4. SARAH-3 Adaptation

The following work is done using minimum amount of SARAH-3 data in order to technically make the SHADECast model technically work, prior to producing meaningful results. Thus, the data specifications and potential result values are considered irrelevant at this point. The main focus is on code structure and needed modifications.

### *Preprocessing*

We decided to focus on SARAH 3 data that, in contrast to the SARAH-2 dataset, does not offer gridded surface radiation products including SIS (surface incoming shortwave radiation) and CSI as ready-to-use variables. Therefore, I computed the clear-sky irradiance (SIS_CLEAR) using the Ineichen model via **pvlib**, based on geographic coordinates, elevation, and Linke turbidity, and subsequently derived CSI as the ratio between SIS and SIS_CLEAR.

All the codes and implementation have to be adapted accordingly, to provide working SHADECast model. To use SHADECast with SARAH-3 data, the following modifications need to be performed:

- Preprocess SARAH-3 SIS fields into CSI using pvlib and save as NetCDF.
- Replace HelioMont loader with SARAH-3-compatible loader in datasets/csi_dataset.py
- Adjust spatial resolution and subdomain in YAML configs.
- Update sampling interval to 30 minutes.
- Train models

Thus, the first step in adapting SARAH-3 data for CSI-based modeling was to generate clear-sky irradiance estimates using the Ineichen clear-sky model implemented in `pvlib`, combined with spatially and temporally resolved Linke turbidity values. This procedure was applied for each latitude–longitude–timestamp tuple in the SARAH-3 dataset. To achieve this, the workflow involved several preparatory stages: parsing the original SARAH-3 filenames to extract exact acquisition timestamps, subsetting the dataset to the spatial domain of interest, and retrieving site-specific elevation data to feed into the Ineichen model. For each tuple, the solar position (zenith and azimuth angles) was computed using standard solar geometry calculations, ensuring physical consistency across diurnal and seasonal cycles. These elements allowed the calculation of direct normal and global horizontal clear-sky irradiance values, which were then combined with the observed SARAH-3 surface incoming shortwave (SIS) values to compute the Clear-Sky Index (CSI) as the ratio `CSI = SIS / SIS_clear`. This CSI representation serves as a normalized, dimensionless input for the generative nowcasting architecture, decoupling the prediction task from absolute radiative values and focusing the learning on deviations from clear-sky conditions.

Example of how SIS_CLEAR is computed using pvlib for SARAH-3 input:

```
solar_pos = pvlib.solarposition.get_solarposition(time=time_utc,
latitude=lat, longitude=lon)
zenith = float(solar_pos["zenith"].values[0])
rel_airmass = pvlib.atmosphere.get_relative_airmass(zenith)
abs_airmass = pvlib.atmosphere.get_absolute_airmass(rel_airmass)
tl = pvlib.clearsky.lookup_linke_turbidity(pd.DatetimeIndex([time_utc]),
lat, lon).iloc[0]
alt = location.lookup_altitude(lat, lon)
clearsky = pvlib.clearsky.ineichen(zenith, abs_airmass, linke_turbidity=tl,
altitude=alt)
sis_clear[t_idx, i, j] = clearsky["ghi"]
```

Final CSI is computed and clipped as: `csi = np.clip(sis.values / sis_clear, 0, 2)`

To integrate SARAH-3 CSI data into the SHADECast framework, several preprocessing adjustments were required within the CSIDataset. The original input images had spatial dimensions of 20×40, which, after passing through the VAE encoder (downsampling by a factor of $2^2$), resulted in latent representations of size 5×10. The subsequent UNet downsampling step further reduced these dimensions to 3×5, which proved incompatible with the Nowcaster component due to missing spatial context entries, leading to a KeyError (3, 5). To resolve this structural mismatch, the preprocessing pipeline was revised so that the input images are padded directly within CSIDataset to spatial dimensions that are powers of two, specifically 32×48. This ensures consistent and compatible spatial resolutions across the VAE, UNet, and Nowcaster stages. In addition, minor adjustments to the normalization procedure were introduced and explicitly annotated in the dataset code. With these changes, the SARAH-3 data preprocessing was successfully completed and fully aligned with the SHADECast model architecture.

### *VAE model*

The Variational Autoencoder (VAE) is used in SHADECast to compress input CSI sequences into a latent space, enabling efficient denoising via a diffusion model. Architecturally, it consists of a 3D convolutional `Encoder`, a bottleneck transformation into latent moments (mean and log variance), and a mirrored `Decoder` that reconstructs the full CSI sequence from the sampled latent representation. The model is trained using a weighted combination of reconstruction loss and KL divergence from the standard normal distribution.

In our implementation, a key modification was introduced to adapt the VAE to SARAH-3 CSI data. The original VAE `_loss()` function expects a single input tensor `x`, whereas the `CSIDataset` returns a tuple `(input_seq, target_seq, timestep)`. To resolve this mismatch, a new `prepare_batch()` method was implemented inside the VAE class. This method concatenates `input_seq` and `target_seq` along the time dimension, producing a single sequence compatible with the VAE's input expectations. This change required minor rewrites in `training_step()` and `val_test_step()` to ensure the batch is processed through `prepare_batch()` before loss computation. All other architectural details remain faithful to the original SHADECast design.

### *VAE training*

To adapt the VAE training to SARAH-3 CSI data, several key changes were introduced in both the training script and configuration. The most critical adjustment was adapting the dataloader logic to the new `CSIDataset` format, which provides `(input_seq, target_seq, timestep)` rather than a single tensor. As a result, the original `get_dataloader()` function was replaced by a new version that builds dataloaders directly from `CSIDataset`, accepting parameters such as `root_dir`, `sequence_length`, `input_length`, and `output_length`, and omitting the original normalization arguments (now handled internally in the dataset). Corresponding changes were made in the instantiation of `train_dataloader` and `val_dataloader`, ensuring that total sequence length was correctly calculated as `input_length + output_length`. Additionally, the configuration file (`testVAEtrainingconf.yml`) was rewritten to reflect these parameters, and a new training script (`adaptedVAETraining_pl.py`) was implemented to support these dataset and configuration changes. These adaptations made it possible to seamlessly integrate SARAH-3 CSI as the input source for the VAE module, while preserving all core logic of SHADECast's original VAE training framework.

### Nowcaster model

The Nowcaster module in the SHADECast architecture is designed for short-term spatiotemporal prediction of cloudiness fields represented by the Clear-Sky Index (CSI). It leverages the latent representation produced by a VAE encoder and processes it using a stack of AFNO blocks, with an optional temporal transformer to map between the input and output time dimensions. In the original implementation, the model treated time implicitly—using generated sequences such as `torch.arange`—which assumed uniform temporal spacing and lacked connection to real-world timestamps.

In this project, the Nowcaster was adapted to work with SARAH-3 CSI data, where time is an explicit, physical dimension. The most significant modification was the consistent introduction of the real timestep tensor as part of the input data, passed directly from the `CSIDataset`. The `Nowcaster` LightningModule was updated so that the `_loss()` method now unpacks batches as `(x, y, t)` instead of the original `(x, y)`, and the `timestep` is used throughout the forward pass. The `forward()` method in both the module and the underlying `AFNONowcastNet` network was modified to accept this `timestep` input and propagate it through all components.

Within the core `AFNONowcastNetBase` model, positional encodings are now computed using the actual timestamps, not synthetic indices. Additional logic was added to handle situations where the autoencoder compresses the time dimension, ensuring that the positional encodings remain aligned with the latent temporal resolution. Furthermore, when using the temporal transformer, separate encodings are now computed for both the input (past) and output (future) time vectors. The transformer uses these to query future latent states in a way that reflects real temporal context.

These changes enabled the Nowcaster to interpret SARAH-3 data sequences as temporally anchored inputs, without altering the original architecture's integrity. The model remains fully SHADECast-compatible but is now semantically consistent with real satellite-derived forecast data.

### Nowcaster training

The Nowcasting training pipeline was adapted to operate directly on SARAH-3-derived CSI data, using the latent representation provided by a pretrained or trainable VAE. The main structural adjustment involved aligning the `train_dataloader` and `val_dataloader` logic with the `CSIDataset`, which provides temporally structured CSI sequences in the form `(input_seq, target_seq, timestep)`. The training script was updated to define `sequence_length`, `input_length`, and `output_length` explicitly, avoiding assumptions from previous KI-based data formats. Another key change was enforcing compatibility between the VAE latent dimension and the Nowcaster's projection layer: the `embed_dim_out` parameter was set to match `vae.hidden_width`, ensuring the model correctly maps latent-encoded CSI sequences through the AFNO-based nowcasting network. These changes were carefully annotated in the code, and the modified configuration file `testNowcastertrainingconf.yml` reflects the required dataset, optimizer, and model parameter adjustments necessary for full SHADECast compatibility with SARAH-3.

### Current status and future work

The `SHADECastTraining.py` script was modified analogously to the VAE and Nowcaster training routines, adapting the data loading logic to work with the `CSIDataset` structure and the SARAH-3 CSI data format. These adjustments ensure that both `train_dataloader` and `val_dataloader` correctly parse input, target, and timestep dimensions based on the expected sequence configuration.

Beyond these data-handling changes, the core architecture of the `UNetModel` and `LatentDiffusion` components remained mostly unchanged. Their original design—based on the SHADECast paper—was preserved to maintain architectural fidelity. The `UNetModel` acts as the denoiser in the latent space, and the `LatentDiffusion` module wraps the full denoising and noise scheduling logic conditioned on spatial-temporal context embeddings provided by the Nowcaster cascade.

At this stage, the training setup is almost complete and structurally aligned with the original SHADECast framework but adapted to SARAH-3. Future work includes testing (it is possible that some minor technical issues remained) after preparing a reasonably large dataset instead of minimal working example, hopefully during the next stay in order to discuss results and relevant issue while supervised. The next logical step would be to prepare an experiment and then perform evaluation and asses spatial and temporal consistency of these predictions. Once the pipeline is verified on larger input sequences and multiple conditions, the model output could be expanded towards producing deterministic or probabilistic EPS (ensemble) predictions. Additionally, introducing NWP-based inputs—such as cloud cover or temperature from coarse-resolution forecasts—could be explored to improve spatial context sensitivity, although this would belong to a longer-term, secondary phase of development.

**Relevant papers:**

A. Carpentieri, D. Folini, D. Nerini, S. Pulkkinen, M. Wild, A. Meyer,
*Intraday probabilistic forecasts of surface solar radiation with cloud scale-dependent autoregressive advection*, Applied Energy, Volume 351, 2023, 121775, ISSN 0306-2619,
https://doi.org/10.1016/j.apenergy.2023.121775.

A. Carpentieri, D. Folini, J. Leinonen, A. Meyer,
*Extending intraday solar forecast horizons with deep generative models,*
Applied Energy, Volume 377, Part A, 2025, 124186, ISSN 0306-2619,
https://doi.org/10.1016/j.apenergy.2024.124186.

## Acknowledgments

The authors would like to thank our colleagues in the ACCORD and RC-LACE consortiums for their support of our work.