Memo

To
Peter de Grave (Deltares) & Alex Roos (RWS) / SITO-PS 2025 WVH02

Date	Our reference		Number of pages
18 December 2025	11211566-012-ZWS-0001		1 of 16
Contact person	**Direct number**	**E-mail**	
Julius Sumihar	+31(0)6 1582 1832	julius.sumihar@deltares.nl	

Subject
Long term storm surge forecasts - Phase 1

1 Project motivation and goal

This memo is part of the results of the "Long-term wind field predictions" project, which is being carried out under the SITO-PS program Strengthening Water Safety Research (in Dutch: Versterking Onderzoek Waterveiligheid, or VOW) that builds on the research that started in 2023. In Deltares (2023), the exploration of available data and techniques regarding long-term (ensemble) expectations, say 10-20 days ahead, of wind speeds and MSL (mean sea level) air pressure is described. A possible use of such long-term ensemble expectations is to predict whether, for example, the coming weeks are suitable for carrying out various operations along the coast, such as maintenance of storm surge barriers or dikes. This can be achieved by using the ensemble meteorological expectations as the driving force of a model that calculates the probability of events in the coastal area.

A traditional way is to use the ensemble meteorological expectations as input for a hydrodynamic model or wave model. However, these models are computationally expensive. In practice, the implementation is often hindered by the required computation time and resources. Therefore, another method is needed.

A weather pattern technique was recommended as an alternative. The idea of the technique is to define a limited set of discrete weather patterns based on historical weather data over a domain (Neal et al, 2018). Additionally, a statistical model is developed that relates each weather pattern to coastal events (target variables). In an operational environment, each weather field from an ensemble meteorological expectation is assigned a weather pattern and thus, based on the statistical model, the probability of a coastal event is determined. A disadvantage of this method is that a lot of relevant information is lost in the step from weather ensemble to patterns. For linear problems, it can be proven that the principal components retain more information than any other method, but in practice (also here) many problems are not linear. A modern extension that also works for non-linear problems is the Variational Auto Encoder (VAE) method. VAE is a Machine Learning method that also works with patterns. Instead of discrete patterns in the input space, the patterns here are represented as a series of numbers in the so-called latent space. The latent variable is continuous and represents the main features of the input data. A neural network is also needed here to relate the patterns in the latent space to the target variables.

1.1 Project goal

The intended ultimate goal is to develop a Machine Learning (ML) model for predicting likelihood of coastal events that are decisive for operational decisions, with an meteorological forecasts ensemble (e.g., ECMWF-ENS) as the driving force.

Note that this outcome is only expected in two years. This memo summarizes the work that has been done in Phase 1 in 2024-2025.

2 Project plan

In the project plan, a number of activities are defined, divided in four phases. The final phase concerns operationalization. The question is whether this phase should take place within the framework of VOW. For completeness, this phase is included in the overall plan. The other three phases are:

Phase 1 – Prototype ML model - Use ERA5 reanalysis and DCSM storm surge at Hoek van Holland to develop a proof of concept
Phase 2 – Development of a definitive machine learning model
Phase 3 – Validation of the machine learning model for probabilistic forecasting of target variables

This memo summarizes the work that has been done in Phase 1.

3 Concepts

In Phase 1 of this project, we aim to develop an ML model, as a proof of concept. The ML model consists of two components and is illustrated in Figure 3.1: (1) for reducing meteorological forecast dataset and (2) for relating the reduced dataset to a coastal target variable. Here, for the first component, a VAE model is developed to extract essential patterns of the meteorological dataset into another (latent) dataset of much smaller dimension than the meteorological data itself. After some first tests and iterations, it was chosen to use an LSTM (Long Short Term Memory) network as the second component to translate the latent environmental conditions determined by the VAE to a coastal target variable. As the target variable, we proposed to use storm surge at some locations along the Dutch coasts. Here, we used the storm surge computed using the DCSM7c (0.5nm), defined as the difference between water level output of DCSM forced by ERA5 and the output of DCSM forced only by astronomical tides, without any meteorological input. The aim of this phase is to check if it is possible to develop such an ML model to reproduce the storm surge calculated by the DCSM model. To start with, we used only one location, i.e. Hoek van Holland.

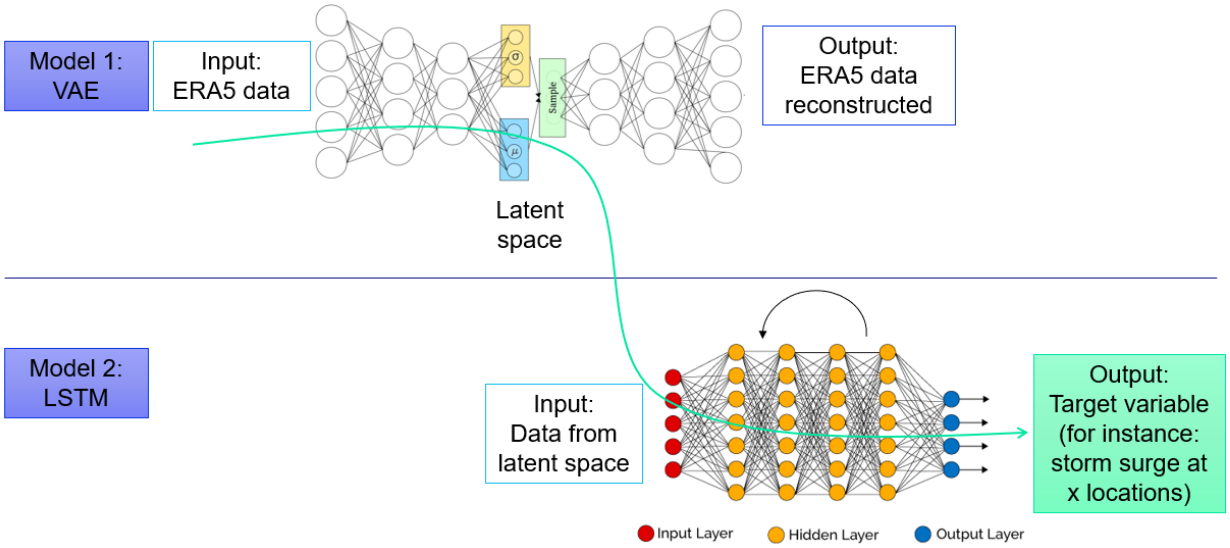

Figure 3.1 Conceptual overview of the ML model

3.1 VAE

3.1.1 Introduction to VAE

An autoencoder (AE) is a type of neural network architecture designed to non-linearly encode (compress) data down to its essential features. This property makes it a valuable tool in geoscience applications that often seek the consistent structures within patterns in considerable volumes of data. It has found various applications (Behrens et al., 2022; Growth and Chavez, 2024; Hsieh and Wu, 2024; Mooers et al., 2020; Naeini and Snaiki, 2024; Olivera et al., 2021).

An AE consists of two components: an encoder and a decoder. The encoder extracts essential features from high-dimensional input variables to low-dimensional latent variables. The decoder uses the latent variables as input to reconstruct the high-dimensional input variables back as output.

In an AE, the latent space is determined to minimize the reconstruction loss. As a result, the latent space may not be continuous and does not allow for interpolation nor for physical interpretation, as the gaps between the points representing the input data in the latent space are undefined.

To solve this issue, Kingma and Welling (2014, 2019) proposed the concept of Variational Auto-Encoder (VAE). Instead of a deterministic latent space, the VAE's encoder allows stochastic mappings between the input space and latent space. The VAE's encoder generates parameters for a predefined distribution within the latent space. The training of a VAE is designed not only to minimize the reconstruction loss, but also a regularization loss, which measures how much the learned latent distribution deviates from a prior distribution (based on Kullback-Leibler divergence). The KL divergence loss ensures smooth latent space, encouraging similar inputs to have similar latent representations. This allows also an extra feature of VAE for generative modelling: by aligning the latent space to a known distribution, it is possible to sample a new data point in the latent space as input to the decoder.

3.1.2 VAE model setup

In this study, as input to the VAE, we use air pressure, u- and v-wind velocity of ERA5 data, defined over a spatial grid of 201 by 103, covering 10 year period: 01-01-2012 00:00 until 31-12-2021 23:00. The data is divided randomly in time into three sets: training dataset (70%), validation dataset (15%, used during training for validating the VAE model), and test dataset (15%, used for testing the trained VAE model). The validation dataset is used by the algorithm during training a.o. to check if the model performs well too to unseen data and to stop the training if the performance doesn't improve. The test dataset is used to test the final model, after the training, using data not used in the training. Prior to using them as input to a VAE model, all input data are converted to dimensionless and scaled data in the range of [0,1].

A suitable architecture of a VAE that fits best our purpose should be searched by trial and error. A number of VAE model structures have been explored, including varying the latent space dimension and the weighting factor of the background term. Here we report only two of them: one that we obtained in the end of 2024 and another one that we obtained recently, referred to as VAE2024 and VAE2025, respectively. They differ in the neural-network layers used as building blocks of the VAE architecture, as described in Appendix A. The VAE2024 model uses a number of dense layers, while VAE2025 a combination of 2D-convolutional, batch normalization and dense layers. The dense layer basically exploits the relation between data at one point with data at all other points, without taking the relative locations from each other into consideration. On the other hand, the 2D-convolutional layer exploits the relative spatial locations to preserve local information. Both VAE models are designed to reduce the

dimension of the meteorological data from (3 * 201 * 103) = 62109 variables in physical space to 32 variables in latent space, for a single time level. The latent space dimension of 32 was chosen as a starting point. Intuitively, it is the nearest power-of-two multiple to the number of weather patterns used at UK Met Office (30 patterns; Neal et al, 2018). A number of tests have been done with smaller latent dimensions using the VAE2024 model. However, they led to issues of numerical instability (KL loss = NaN) or posterior collapse (KL loss = 0).

As common to any data reduction, we expect to lose certain detail features of the input data in the reconstruction output of the VAE model. Our aim is to reduce the dimension as much as possible while retaining the features crucial for the underlying process of the target variables.

3.2 LSTM

3.2.1 Introduction to LSTM

Recurrent Neural Networks (RNNs) are a class of neural networks designed for processing sequential data. Unlike traditional feedforward neural networks, RNNs have connections that form cycles, allowing them to maintain a 'memory' of previous inputs. This makes them particularly suitable for tasks where the context of previous data points is important, such as time series prediction, natural language processing, and speech recognition.
Long Short-Term Memory (LSTM) networks are a further developed type of RNN designed to avoid typical problems that can occur with traditional RNNs. LSTMs are particularly effective for tasks involving sequential data, such as time series prediction, because they can maintain and update information over long sequences. The method was first proposed by Hochreiter and Schmidhuber (1997) and has found many applications a.o. in hydrological forecasting (e.g. Kratzert et al, 2018; Gauch et al, 2021), in sea level forecasting (e.g. Yang et al, 2020; Accarino et al, 2021) and storm surge forecasting (Tiggeloven at al, 2021; Hermans et al, 2025).

Key Features of LSTMs:

- Memory Cells: LSTMs have special units called memory cells that can store information for long periods.
- Gates: They use three types of gates (input, forget, and output gates) to control the flow of information:
    - Input Gate: Decides what new information to store in the cell.
    - Forget Gate: Decides what information to discard from the cell.
    - Output Gate: Decides what information to output from the cell.

Why LSTMs are Commonly Used in Time Series Prediction:

- Handling Long-Term Dependencies: LSTMs can remember information for long periods, making them suitable for time series data where past events influence future events.
- Avoiding Vanishing Gradient Problem: The architecture of LSTMs helps mitigate the vanishing gradient problem, allowing them to learn from long sequences more effectively.
- Flexibility: They can be used for various time series tasks, including forecasting, anomaly detection, and sequence classification.

LSTMs are powerful tools for modelling time-dependent data, making them a popular choice for time series prediction tasks.

3.2.2 Model set-up

As input data the latent meteorological information, see Paragraph 3.1, is used. The latent meteo consists of 32 values per time step (z0 – z31 in the sample in Figure 3.2). In addition, the historical storm surge at Hoek van Holland is given as input (s in the sample in

Figure 3.2). Note that if the storm surge at more locations would be input (s0 – sx), the model can be trained to forecast at those locations as well.

In this phase of the project it was chosen to use 3 days of historical data to predict 5 days with hourly time-step (120 timesteps in total) in the future. It was hypothesized with 3 days of history the model has enough information about the weather patterns, e.g. influence of nearby storms and surge at that time to predict the future. In the next phase it will be determined which input will be used for the model; it will determined the eventual forecast length. It is recommended to investigate the influence of a shorter and longer historical period to improve the prediction.

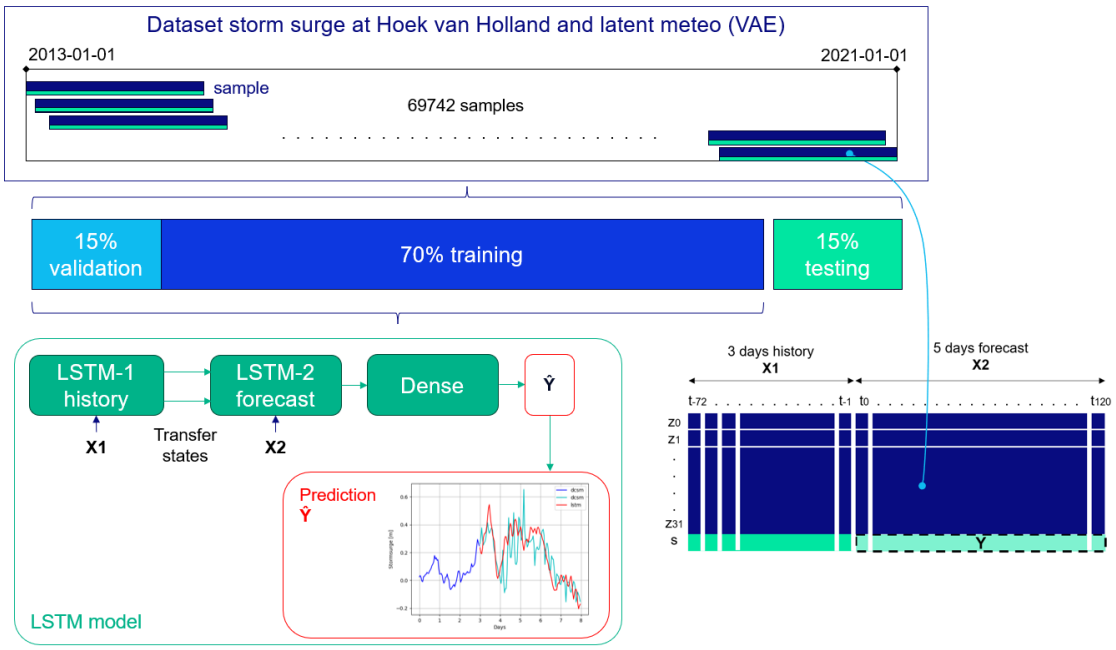

Figure 3.2 Architecture of the dataset and LSTM model

The currently tested LSTM model set-up is based on the work of Deng (2023), specifically ‘Model 2’. This model is constructed with two LSTMs arranged in cascade configuration. LSTM-1 (history) processes data from the past 3 days sequentially. The final cell state and hidden state of LSTM-1 are passed through a fully connected layer (FC), and the resulting “transferred states” are used as the initial cell state and hidden state for LSTM-2 (forecast). LSTM-2 processes meteorological data of the future 5 days, and returns the full sequence of outputs. The outputs of LSTM-2 are then fed into the Dense layer, which produces the final output of 120 predictions simultaneously.

4 Results

4.1 VAE – encoding meteorological data

The VAE models are evaluated here visually by comparing the reconstructed data to the original data from the test dataset. Figure 4.1 - Figure 4.3 show result examples of VAE2024 and VAE2025 at one particular time level. The results show that VAE2025 performs much better than VAE2024.

The results of VAE2025 show general patterns that are consistent with the original data. However, VAE results are rather blurry, miss certain details, and may not have the right magnitude. This is most pronounced in wind speed data. The question is whether they contain sufficient information about the target variables. To answer this question, we took a pragmatic

approach by using the latent variables generated by VAE2025 as input to the LSTM model and evaluating the results (next section).

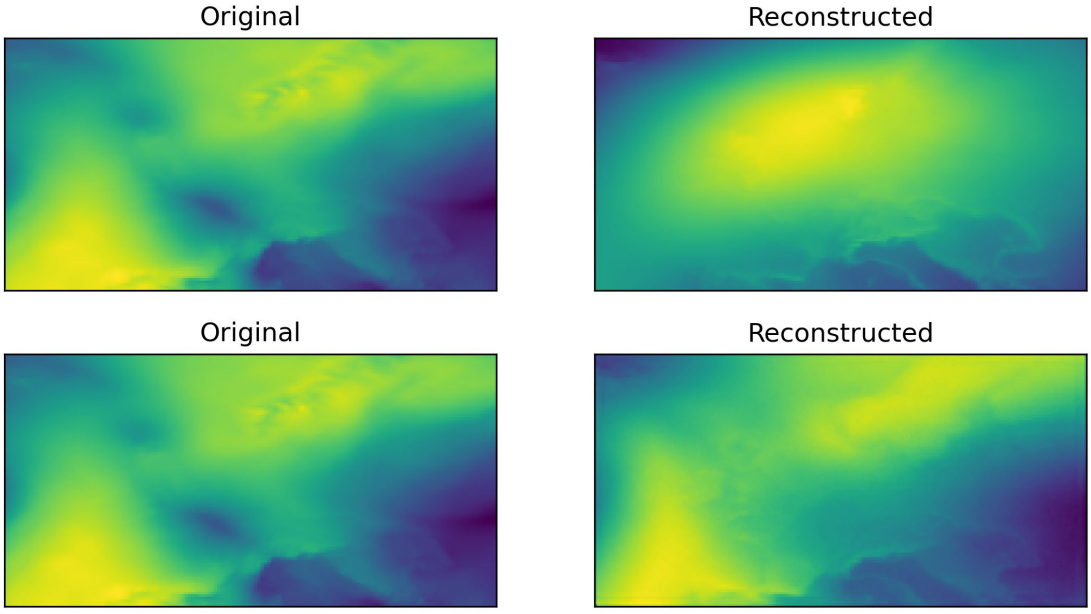

Figure 4.1 Original vs reconstructed (scaled) air pressure data at a certain time level generated by VAE2024 (top) and VAE2025 (bottom)

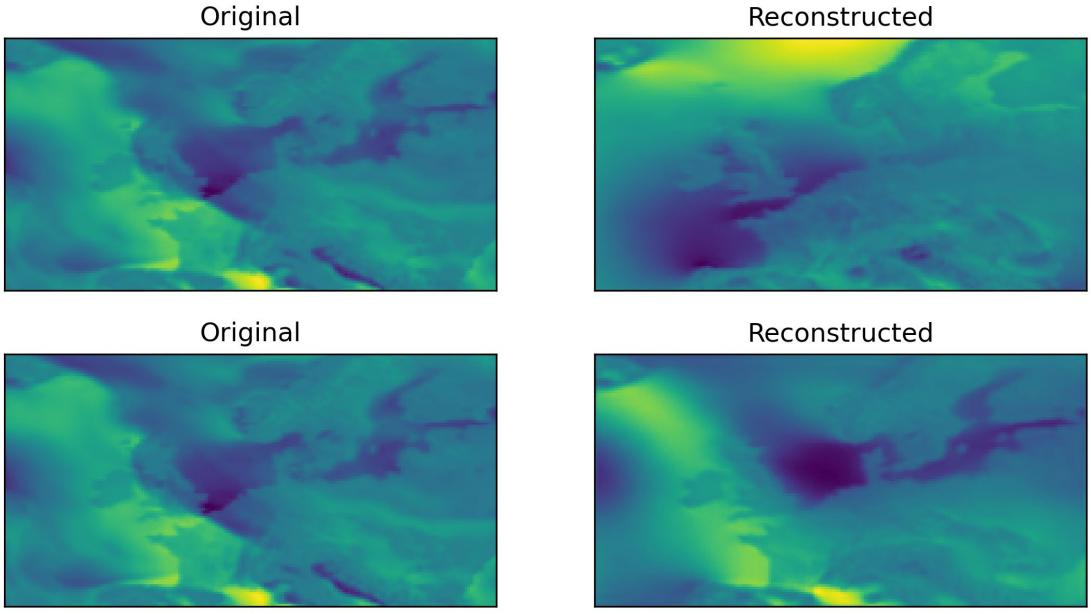

Figure 4.2 Original vs reconstructed (scaled) eastward wind speed at the same time level as Figure 1, generated by VAE2024 (top) and VAE2025 (bottom)

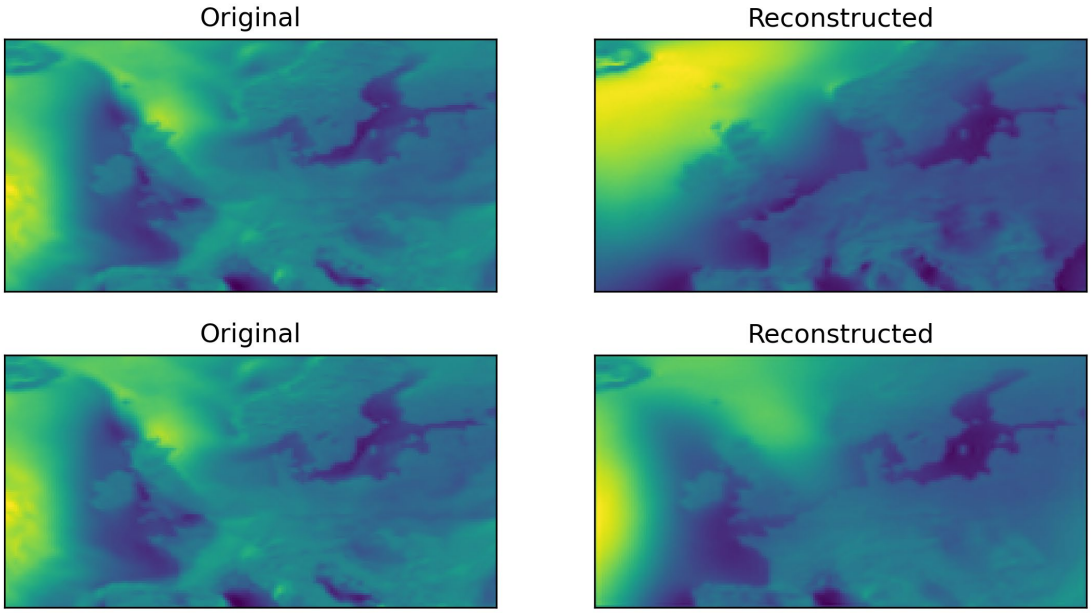

Figure 4.3 Original vs reconstructed (scaled) northward wind speed at the same time level as Figure 1, generated by VAE2024 (top) and VAE2025 (bottom)

4.2 LSTM – predicting storm surge

The LSTM model setup contains many calibration possibilities. In this proof of concept phase a couple of variations have been tested. In this paragraph the results of the (at this point) best performing model is shown to proof the concept. It is noted that the results can be further improved, see the recommendations in Chapter 5. Given the proof of concept phase the current performance is considered sufficient to move to the next phase.

To illustrate the performance of the model, first timeseries of DCSM ('true' storm surge) are compared with LSTM. Figure 4.4 shows the test dataset and indicates which periods are selected for further analysis. The analysis periods are selected based on high storm surge peaks, which are most relevant to predict accurately. Figure 4.5 and Figure 4.6 show the timeseries of the DCSM model storm surge and the LSTM model output.

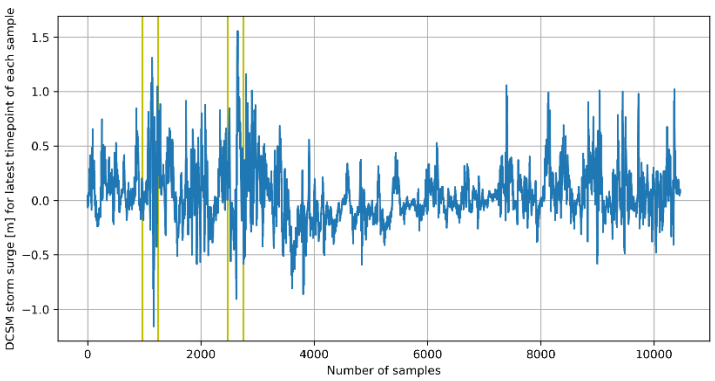

Figure 4.4 Impression of the test dataset. The vertical lines indicate the further analysed periods.

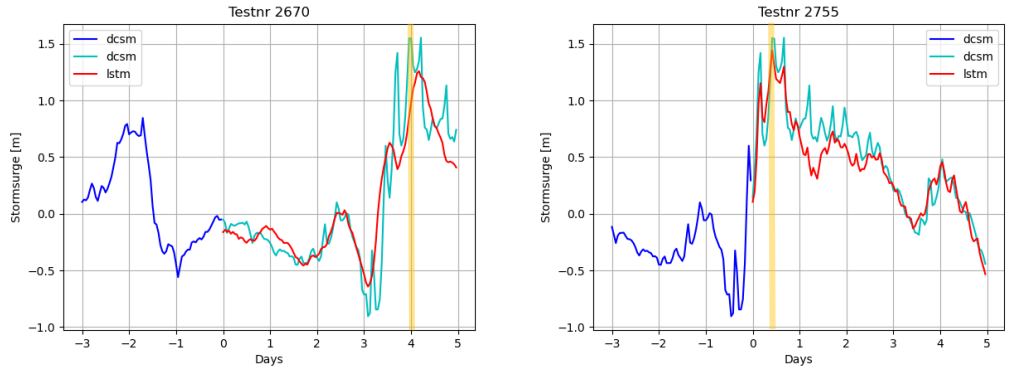

Figure 4.5 DCSM storm surge history in blue and forecast in turquoise and the LSTM forecasted storm surge in red. The panels show a storm surge event approaching Hoek van Holland. The difference between both panels is approximately 3,5 days. The yellow stripes indicate the position of the storm surge peak.

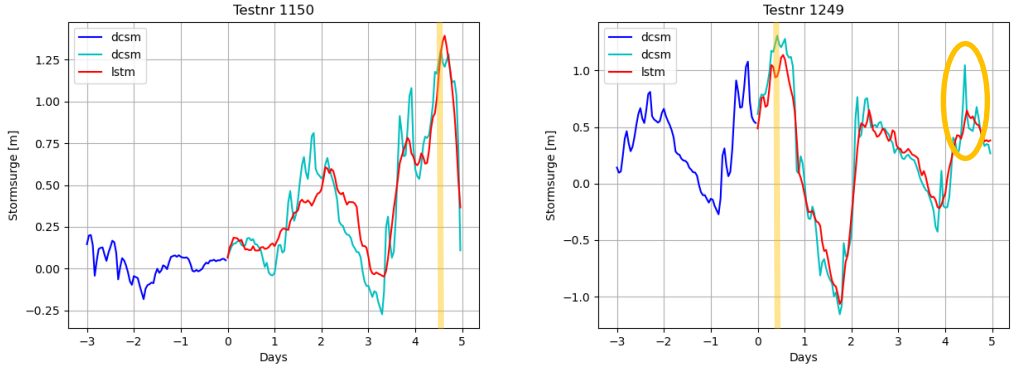

Figure 4.6 DCSM storm surge history in blue and forecast in turquoise and the by LSTM forecasted storm surge in red. The panels show a storm surge event approaching Hoek van Holland. The difference between both panels is approximately 4 days. The yellow stripes indicate the position of the storm surge peak.

Figure 4.5 shows that the LSTM prediction of the storm surge peak can improve when it gets closer to t=0, so it seems to properly make use of the past 3 days of data (X1). However, also examples of the opposite are found, the storm surge peak in Figure 4.6 was predicted less well closer to t=0.

This figure also shows that not all narrow and high peaks are predicted well, see the yellow circle in the right panel at 4.2 days forecast. It is hypothesized that these narrow peaks of 1 to 2 hours (due to local effects that are not included in the input yet?) can be predicted less well than broad(er) (meteo driven?) peaks. This is further investigated by plotting the scatter including the derivative of the DCSM model, Hoek van Holland storm surge. Figure 4.7 shows the scatter plots of the performance of LSTM relative to DCSM. Note that the scatter plots were made using all LSTM forecasts for all forecast lead times; the vertical stripes occur from the fact that each DCSM output is forecasted multiple times by LSTM. The right panel also shows the influence of storm surge peakiness by plotting the scatter in the colour of the derivate of DCSM (storm surge at t_i minus storm surge at t_{i-1}, see blue line in Figure appendix B.1). A large derivative (blue or red) means a large peakiness or in other words a large difference with the previous timepoint, a small derivative (yellow) means a small difference with the previous timepoint. It can be seen that many of the outer points (furthest away from the 1-1 line, perfect match) have a high peakiness.

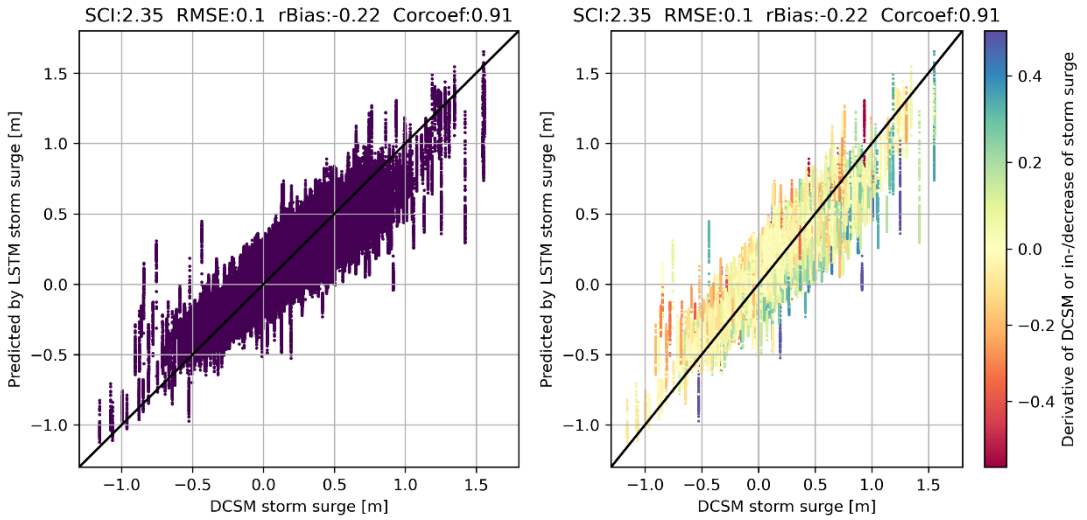

Figure 4.7 Scatter plots of DCSM ('true' storm surge) and the LSTM model. The titles show the statistical performance with SCI the scatter index, RMSE the root mean square error, rBias the relative bias and Corcoef the correlation coefficient.

Figure 4.8 further illustrates the performance of LSTM via a boxplot. The mean storm surge (orange horizontal line) is predicted quite well as the difference between LSTM and DCSM is close to zero. The 5 and 95 percentiles are visualised by black whiskers (horizontal lines above and below the box). The 5 and 95 percentiles of the differences between LSTM and DCSM are approximately -0.17 and 0.16 meter. The blue dots show the differences for one timepoint outside the 5 and 95 percentiles. It can be seen that the difference between the LSTM model prediction and DCSM model can go up to 1 meter, which is quite large when compared to the absolute storm surge values. The absolute maximum and minimum storm surge at Hoek van Holland of the whole dataset lies between -1.16 and 1.55 meter. On average, the amount of storm surge is smaller, the 5 and 95 percentiles of the DCSM storm surge are approximately -0.3 and 0.5 meter. (Note that in this proof of concept ensembles are not considered and therefore their uncertainty is not yet included. Here, the ERA5 reanalysis wind product and DCSM storm surge output are used to train the VAE and LSTM model).

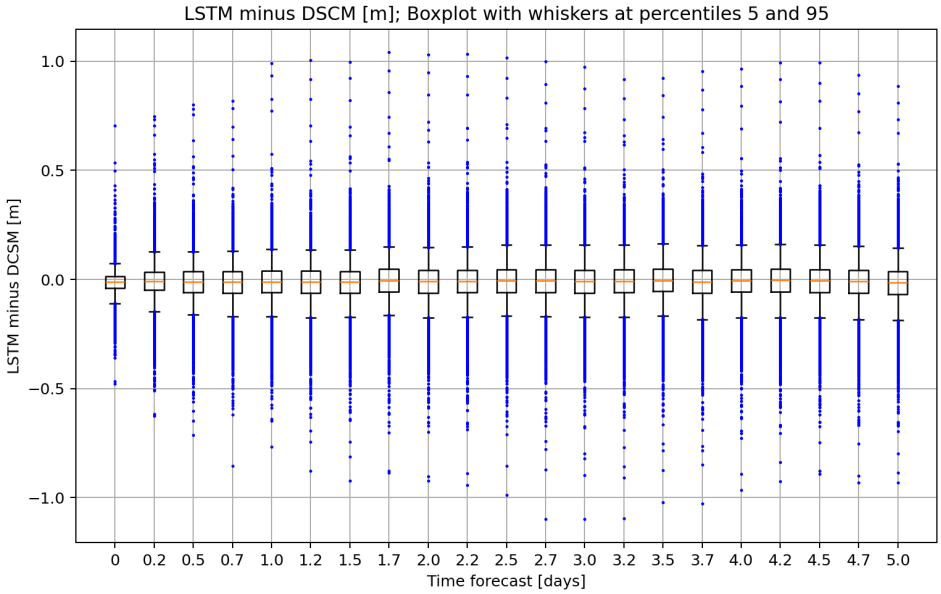

Figure 4.8 Boxplot (in black) of the differences between the LSTM model results and the DCSM storm surge. The blue points show the differences outside the 5 and 95 percentile.

5 Recommendations

Now that the concept is proven for the ERA5 reanalysis, the next step is to consider datasets that enable forecasts for a time span of at least 15-20 days ahead and that include ensembles. Possible examples are the ECMWF-ENS or ECMWF sub-seasonal-range data sets. It is recommended to first consider which dataset suites best as input for the AI train set up here. Secondly the VAE and LSTM should be retrained with the selected dataset. The present study already leads to several recommendations that can be considered once the new dataset is available. Various choices have been made regarding the settings of the two models. Further validation and optimization is recommended, preferably with the new input dataset. Various settings in the LSTM model should be (re)calibrated anyway, given new input datasets.

Thirdly, it is noted that in this proof of concept ensembles are not considered and it is therefore recommended to investigate the best method to include this (new) information. For example, it is possible to include all ensembles in the input (X1) and obtain 1 storm surge forecast (with a to be determined uncertainty), but it is also possible to calculate the storm surge per ensemble (similar to presented proof of concept) and obtain n forecasts. These forecasts can then be post-processed to one forecast with uncertainty bands.

Last, the application of the present model focusses at the water level at Hoek van Holland. To increase its application it is recommended to increase the number of locations with locations relevant for operational purposes or even consider 2D water level fields.

VAE:

- To check the quality of the VAE model, it will be insightful to use the reconstructed meteo data as input to DCSM and check how similar the results are to the ones driven by the original ERA5 dataset. This check should give insights if the reconstructed data contains most essential information for driving storm surge processes despite the lack of detail features. This check can also be useful for interpreting the output of the LSTM model, e.g. whether deviations in storm surge output from DCSM can be attributed to the quality of the input data or to the model itself.
- Check the continuity of latent space. Until now, we only evaluated the VAE model visually by checking the input and reconstructed meteo. A more fundamental check is required to see if the latent space is continuous.
- Examine latent space, explore how we can interpret them, whether a simple relation exists between the latent and physical spaces, e.g. if there are leading dimensions that we can use to study and explain the meteo data as well as the target variables.
- The architecture of VAE2025 is rather simple: it basically contains only two 2D-convolution layers. It is recommended to explore if other architectures can give better results, for example the use of Pooling layers Moreover, it is recommended to explore various the hyperparameters to obtain a better performing VAE configuration.

LSTM:

- Investigate the use of a simple 1-LSTM approach without the Dense layer at the end. The 2-LSTM approach used in this study was developed for hydrological forecasting (Deng, 2023), where observations are available for running the first LSTM model (hindcast). It is however not the case for storm surge forecasting. Hence, a simple 1-LSTM approach may be sufficient. The Dense layer at the end connects the LSTM output at all time levels, hence it makes the model more complicated. Investigate if it is possible to simplify the model by removing it / replacing it by separate Dense layer for each output time level.

- Investigate why the model seems to perform less for steep/sudden storm surge peaks.
    - Evaluate if the current input parameters (pressure, x-velocity and y-velocity) are sufficient or that other parameters such as significant wave height or wave direction should be included in the input.
    - The current configuration doesn't include the non-linear of tides and storm surge. It is recommended to check how non-linear tide and storm surge interaction should be included. Compare also the results with the time series method for forecasting tides, storm surge and their interactions, developed in an AI-impulse North Sea project (Verlaan at al, 2025)
    - Evaluate the pre-processing of the storm surge. Are all seasonal and tidal components sufficiently removed?
    - Evaluate DCSM model performance with measurements
- Optimising the duration of the historical period (input X1). Test longer & shorter periods.
- Scaling per feature (z0, z1 … z31 and s) should result in better LSTM model performance but that is not the case at the moment. This should be tested and optimised.

More nice-to-have recommendations are given below:

VAE:

- The meteo domain used in this study is actually larger than the domain where the storm surges along the Dutch coasts are sensitive to. It may be possible to reduce the meteo domain as input to the VAE. This may simplify the VAE model without losing much on accuracy.
- Check if it is possible to reduce the meteo data even further. The smaller the latent space, the easier it is to interpret and to relate the latent to the physical space.
- To get more insight about the working of VAE, compare the results with the ones obtained using a linear Auto Encoder based on Singular Value Decomposition (SVD).
- For a longer term, it is recommended to explore how to use VAE as a generative model for meteo data. This can be useful e.g. for generating ensemble forecasting of the target variables.

LSTM:

- Alternative configurations of the LSTM model can be tested (see for example Deng (2023) where different architectures are tested).
- Other AI architectures such as GRUs, TCNs etcetra should be evaluated and can be tested.

References

Accarino, G., Chiarelli, M., Fiore, S., Federico, I., Causio, S., Coppini, G., Aloisio, G. (2021), A multi-model architecture based on Long Short-Term Memory neural networks for multi-step sea level forecasting, Future Generation Computer Systems, 124, 1-9, ISSN 0167-739X, https://doi.org/10.1016/j.future.2021.05.008

Deng, J. (2023). Operational streamflow drought forecasting for the Rhine River at Lobith using the LSTM Deep Learning approach. https://repository.tudelft.nl/record/uuid:331a6ac2-0ba9-4726-8672-003cf762ad60

Behrens, G., Beucler, T., Gentine, P., Iglesias-Suarez, F., Pritchard, M., and Eyring, V., (2022), Non-linear dimensionality reduction with a variational encoder decoder to understand convective processes in climate models, Journal of Advances in Modelling Earth Systems, 14, e2022MS003130. https://doi.org/10.1029/2022MS003130

Gauch, M., Kratzert, F., Klotz, D., Nearing, G., Lin, J., and Hochreiter, S. (2021), Rainfall–runoff prediction at multiple timescales with a single Long Short-Term Memory network, Hydrol. Earth Syst. Sci., 25, 2045–2062, https://doi.org/10.5194/hess-25-2045-2021

Growth, A. and Chavez, E. (2024), Efficient inference and learning of a generative model for ENSO predictions from large multi-model datasets, Climate Dynamics, https://doi.org/10.1007/s00382-024-07162-w

Hermans, T. H. J., Ben Hammouda, C., Treu, S., Tiggeloven, T., Couasnon, A., Busecke, J. J. M., and van de Wal, R. S. W. (2025), Computing Extreme Storm Surges in Europe Using Neural Networks, EGUsphere [preprint], https://doi.org/10.5194/egusphere-2025-196

Hochreiter, S., Schmidhuber, J., (1997), Long Short-Term Memory. Neural Comput 1997; 9 (8): 1735–1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735

Hsieh,M.-K., Wu, C.-M., (2024), Developing an explainable variational autoencoder (VAE) framework for accurate representation of local circulation in Taiwan, JGR Atmospheres, https://doi.org/10.1029/2024JD041167

Kingma, D. P., and Welling, M. (2014). Auto-encoding variational bayes. In 2nd international conference on learning representations. ICLR 2014–Conference track proceedings (Ml)

Kingma, D. P., and Welling, M. (2019), "An Introduction to Variational Autoencoders", Foundations and Trends® in Machine Learning: Vol. 12: No. 4, pp 307-392. http://dx.doi.org/10.1561/2200000056

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M. (2018), Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrol. Earth Syst. Sci., 22, 6005–6022, https://doi.org/10.5194/hess-22-6005-2018

Mooers, G., Tuyls, J., Mandt, S, Pritchard, M, and Beucler, T, (2020), Generative Modelling for Atmospheric Convection, https://doi.org/10.48550/arXiv.2007.01444

Naeini, S.S., Snaiki, R. (2024), A novel hybrid machine learning model for rapid assessment of wave and storm surge responses over an extended coastal region, Coastal Engineering 190, https://doi.org/10.1016/j.coastaleng.2024.104503

Neal, R., Dankers, R., Saulter, A., Lane, A., Millard, J., Robbins, G., & Price, D. (2018), Use of probabilistic medium- to long-range weather-pattern forecasts for identifying periods with an increased likelihood of coastal flooding around the UK. Meteorological Applications, 25(4), 534–547. https://doi.org/10.1002/met.1719

Olivera, D.A.B, Dias, J.G., Zadrozny, B., and Watson, C.D., (2021), Controlling weather field synthesis using variational autoencoders, https://arxiv.org/pdf/2108.00048

Verlaan, M., Zijl, F., Zhao, J., Tromp, W., Sumihar, J., Yan, K., Muis, S. (2025), Forecasting tides, storm surges, and their interaction around the North Sea using machine learning, 4th International Workshop on Waves, Storm Surges, and Coastal Hazards

Tiggeloven, T., Couasnon, A., van Straaten, C., Muis, S., Ward, P.J. (2021), Exploring deep learning capabilities for surge predictions in coastal areas. Sci Rep 11, 17224. https://doi.org/10.1038/s41598-021-96674-0

Yang, C.H., C.H. Wu, C.M. Hsieh (2020), Long short-term memory recurrent neural network for tidal level forecasting, IEEE Access, 8 (2020), pp. 159389-159401, 10.1109/ACCESS.2020.3017089

A Model set-up VAE

1 VAE2024:

Model: "encoder"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 103, 201, 3)]	0	[]
flatten (Flatten)	(None, 62109)	0	['input_1[0][0]']
dense (Dense)	(None, 256)	15900160	['flatten[0][0]']
dense_1 (Dense)	(None, 128)	32896	['dense[0][0]']
dense_2 (Dense)	(None, 64)	8256	['dense_1[0][0]']
z_mean (Dense)	(None, 32)	2080	['dense_2[0][0]']
z_log_var (Dense)	(None, 32)	2080	['dense_2[0][0]']
sampling (Lambda)	(None, 32)	0	['z_mean[0][0]', 'z_log_var[0][0]']

Model: "decoder"

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 32)]	0
dense_3 (Dense)	(None, 64)	2112
dense_4 (Dense)	(None, 128)	8320
dense_5 (Dense)	(None, 256)	33024
dense_6 (Dense)	(None, 62109)	15962013
reshape (Reshape)	(None, 103, 201, 3)	0

2 VAE2025

```
Model: "encoder"
__________________________________________________________________________________________________
 Layer (type)                Output Shape                 Param #   Connected to
==================================================================================================
 input_1 (InputLayer)        [(None, 103, 201, 3)]        0         []

 batch_normalization (Batch  (None, 103, 201, 3)          12        ['input_1[0][0]']
 Normalization)

 conv2d (Conv2D)             (None, 103, 201, 32)         896
['batch_normalization[0][0]']

 batch_normalization_1 (Bat  (None, 103, 201, 32)         128       ['conv2d[0][0]']
 chNormalization)

 conv2d_1 (Conv2D)           (None, 103, 201, 64)         18496
['batch_normalization_1[0][0]'
                                                                    ]

 batch_normalization_2 (Bat  (None, 103, 201, 64)         256       ['conv2d_1[0][0]']
 chNormalization)

 flatten (Flatten)           (None, 1324992)              0
['batch_normalization_2[0][0]'
                                                                    ]

 z_mean (Dense)              (None, 32)                   4239977   ['flatten[0][0]']
                                                          6

 z_log_var (Dense)           (None, 32)                   4239977   ['flatten[0][0]']
                                                          6

 sampling (Lambda)           (None, 32)                   0         ['z_mean[0][0]',

'z_log_var[0][0]']

==================================================================================================
__________________________________________________________________________________________________
Model: "decoder"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 32)]              0

 batch_normalization_3 (Bat  (None, 32)                128
 chNormalization)

 dense (Dense)               (None, 1324992)           43724736

 batch_normalization_4 (Bat  (None, 1324992)           5299968
 chNormalization)

 reshape (Reshape)           (None, 103, 201, 64)      0
```

```
conv2d_transpose (Conv2DTr   (None, 103, 201, 32)      18464
anspose)

batch_normalization_5 (Bat   (None, 103, 201, 32)      128
chNormalization)

conv2d_transpose_1 (Conv2D   (None, 103, 201, 3)       867
Transpose)
```

B Additional figures

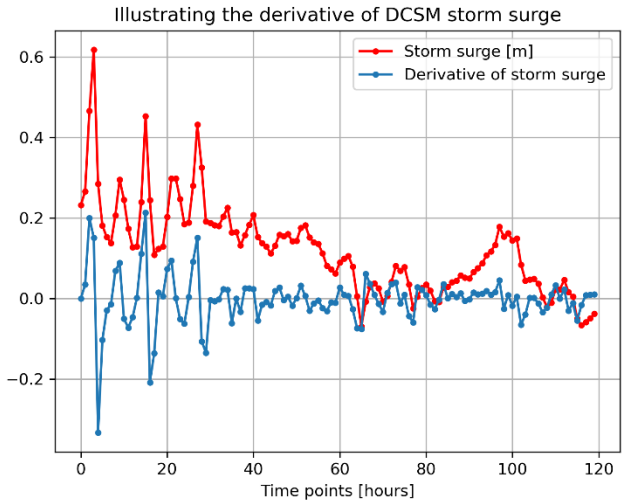

Figure appendix B.1 Example of the derivative of DCSM storm surge (blue line) and the actual DCSM storm surge (red line).