# MACHINE LEARNING METHODS FOR FORECASTING OCEAN DYNAMICS (SEA LEVEL)

**N. Delpeche-Ellmann**[1], A. Ellmann[2] , S. Rajabi-Kiasari[2], V. Jahanmard [2] ,S.Varbla[2] , A. Kupavõh[2]

[1]Department of Cybernetics, School of Science, Tallinn University of Technology, Estonia

[2]Department of Civil Engineering and Architecture, Tallinn University of Technology, Estonia

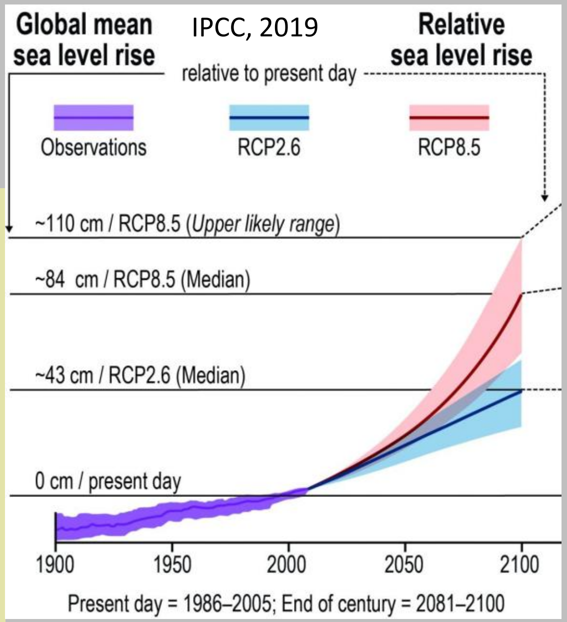
contact: nicole.delpeche@taltech.ee

NKG summer school "From Struve to the Space" Tartu, August 25-29, 2025

# MOTIVATION I

- ***<u>Consistent and accurate</u> sea level is a key component for:***
  - *Safe navigations, coastal protection, marine engineering, climate change, and coastal city flooding*
  - *Operational forecasting systems and planning decisions*
  - *Important for Digital Twin applications*
- ***Sea level maximas and extremes (SLM)*** are a major contributor of coastal flooding, erosion, infrastructure damage etc.
- Based on **climate projections** of IPCC **sea level is rising and extremes are expected to increase with magnitude, frequency and duration**
- ***This <u>signals</u> the need for <u>adaptation and mitigation solutions</u>***
- ***<u>Forecasting</u> of <u>sea level and their extemes are necessity</u> both on the <u>short-term</u> and the <u>long terms perspective</u>.***
- ***<u>Machine and deep learning (ML/DL)</u> approaches can be utilized for some of these <u>solutions</u>***

West gate on Sea, England (photo credit: Max Montagut, www.flickr.com)

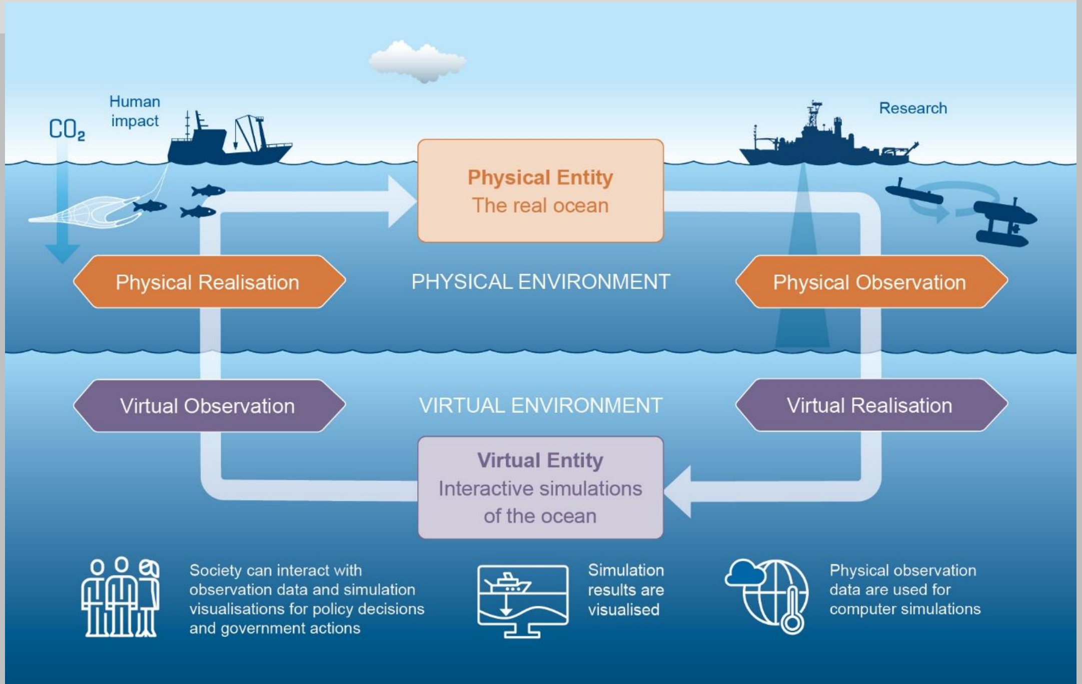


# MOTIVATION II: MACHINE LEARNING AND DEEP LEARNING

- Massive data sets (satellites, in-situ, models)

- Due to **advancements in computing technology** (language processing, computational power, etc.), machine learning (ML)/Deep learning (DL) algorithms have been widely **acknowledged** as robust tools in **finding patterns** and **forecasting in various fields** (Zhou et al., 2023).

- Technological change and **need of society** are increasing--→ digital transformation

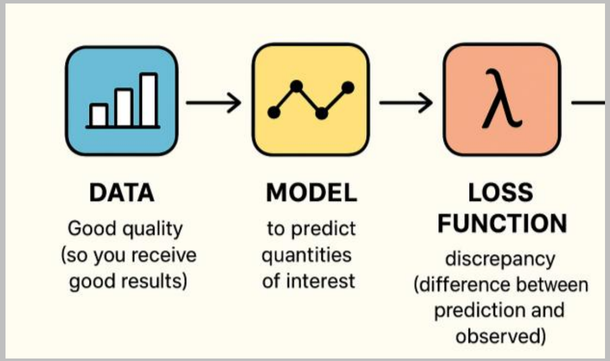



# OUTLINE

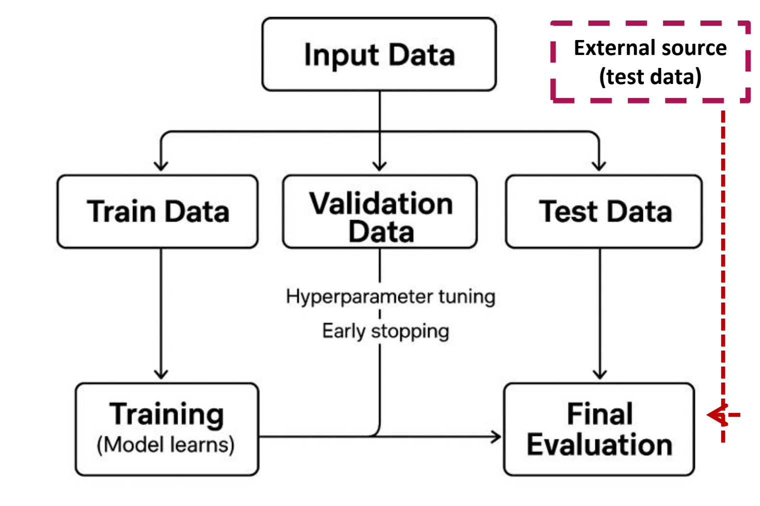
- Background of ML/DL
- Sealevel forecasting in the Baltic Sea
- Extreme sea level forecasting in Baltic Sea
- ML method to improve hydrodynamic model applied to Baltic and Barent Sea

# MACHINE LEARNING BACKGROUND

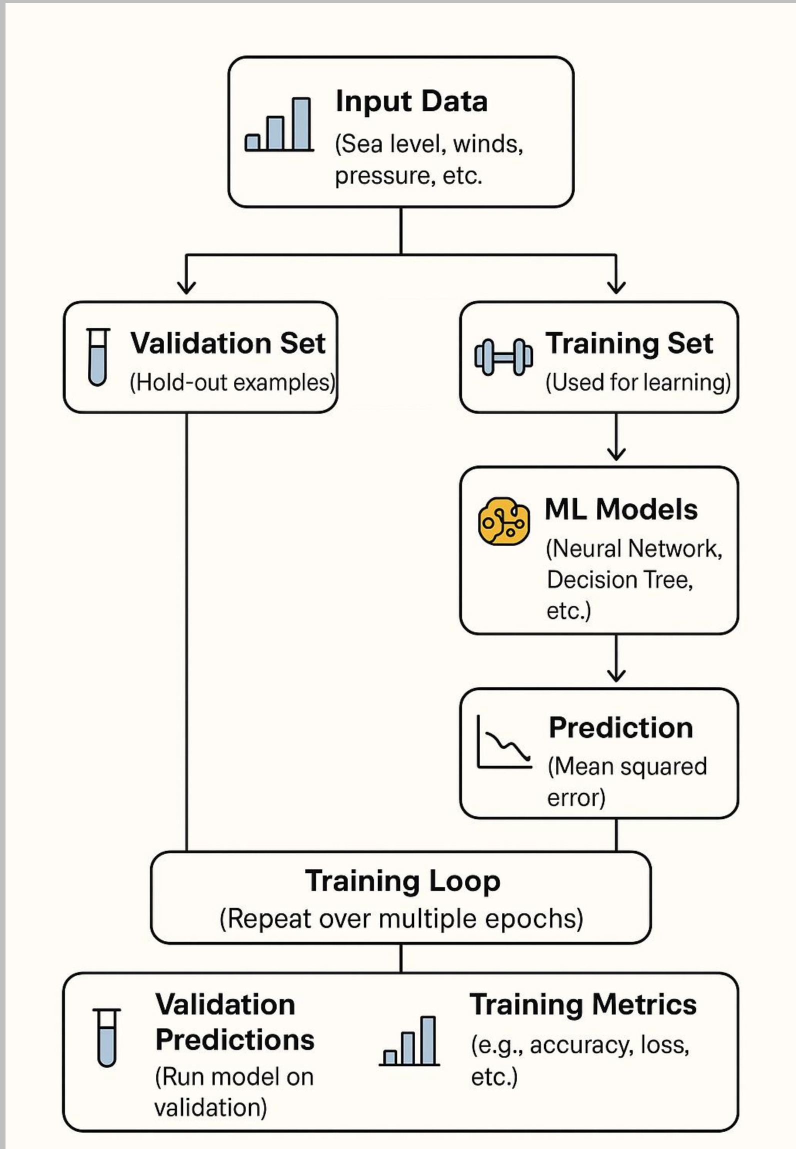
- ML/DL: computer learns to perform tasks based on experience it gains during training.
- Basic components:




- **Data** (as input)
- **A model (i.e a hypothesis):** to predict quantities of interest (model chosen by user)
- **Loss function**: the discrepancy (difference between prediction and observed)
- An iterative approach is used until the loss function is minimum
- Evaluation: test data




*Train*: fitting parameters of model
*Validation*: internal (tuning hyperparameters)
*Test*: final evaluation
External test set: independant source evaluation
**Ratio: 70% train, 30% test**




**Training Loop**

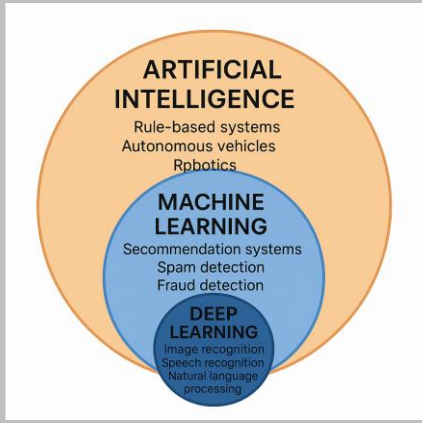
# ML COMPONENTS

| Component | Role |
|---|---|
| **Input Data** | Data source used to train the model; quality heavily impacts performance (e.g hourly sea level data) |
| **Model** | Mathematical structure that learns a mapping from input to output. |
| **Prediction** | Output of the model (e.g., sea level forecast for 24 hours). |
| **Loss Function** | Quantifies the error between predicted value and ground truth. |
| **Ground Truth** | The actual correct output (labels) used for comparison. |
| **Optimization** | Adjusts the model (weights/parameters) to reduce the loss |
| **Training Loop** | Repeated process of prediction → error → update, until model performance is satisfactory. |

TAL TECH

# ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, DEEP LEARNING AND NEURAL NETWORKS

- AI is the main system. Machine learning is a subset of AI. Deep learning is a subfield of machine learning, and neural networks make up the backbone of deep learning algorithms.




- DL is generated in almost the same way as ML, but it has many more levels, so that it attempts to function similar to brain, in that it can take an input, processes it and *then make its own intuitive decisions/predictions*. This, makes it *ideal for large and nonlinear data processing*.

- *Convolutional Neural Networks (CNNs) and Recurrent-based Neural Networks* such as Long Short-Term Memory Networks have been well-known DL methods.

# EXAMPLES OF STUDIES

| Category | Method / Architecture | Typical Applications |
|---|---|---|
| **AI (General)** | Rule-Based Expert Systems | Medical diagnosis support, navigation systems, fault detection |
| | Knowledge Graphs | Search engines, recommendation systems, biomedical discovery |
| **Machine Learning (ML)** | Linear / Logistic Regression | Economic forecasting, medical risk prediction |
| | **Decision Trees** | Fraud detection, medical decision support |
| | **Random Forest (RF**) | Remote sensing classification, finance risk assessment, anomaly detection |
| | **Gradient Boosting** (**XGBoost**, LightGBM, CatBoost) | Kaggle competitions, financial forecasting, ranking systems |
| | Gaussian Process Regression (GPR) | Time series prediction, robotics control, uncertainty quantification |
| **Deep Learning (DL)** | **Convolutional Neural Networks** (CNN) | Image recognition (e.g., ImageNet), facial recognition, medical imaging |
| | **Recurrent Neural Networks (RNN), LSTM, GRU** | Speech recognition, language modeling, stock prediction |
| | Transformers (BERT, GPT, ViT) | Natural language processing, translation, vision tasks |
| | Graph Neural Networks (GNNs) | Social network analysis, drug discovery, traffic prediction |
| | Deep Reinforcement Learning (DQN, PPO, A3C) | Robotics, AlphaGo (game playing), autonomous driving |

# ML/DL COMPONENTS: INPUTS/FEATURES

- DL models are renowned for their *ability* to automatically *extract influential features* and *patterns from raw data*, making them suitable for complex tasks such as time series analysis
- Thus it is *important* to *identify the most influential inputs* affecting the target variable:

**Methods to determine most influential inputs:**

- *Statistical boxplots*: mean, median, interquartile range, and extremes
- *Pearson correlation coefficient* (written as *r*):measures the linear relationship between two variables $X$ and *Y*. Can assist in prevent overfitting
- *Mutual Information (MI) index*: to examine relationships between variables. It measures how much knowing one variable reduces uncertainty about another variable. *Effective in detecting nonlinear relationships.*
- *Aprior knowledge* based on previous studies
- wrapper-type sequential feature elimination algorithm

**Pearson correlation**

$$r = \Sigma ( (x_i - \bar{x})(y_i - \bar{y}) ) / ( \mathrm{sqrt}( \Sigma (x_i - \bar{x})^2 ) * \mathrm{sqrt}( \Sigma (y_i - \bar{y})^2 ) )$$

$r=+1$ → Perfect positive linear relationship

$r=-1$ → Perfect negative linear relationship

$r=0$ → No linear relationship

**MI Index**

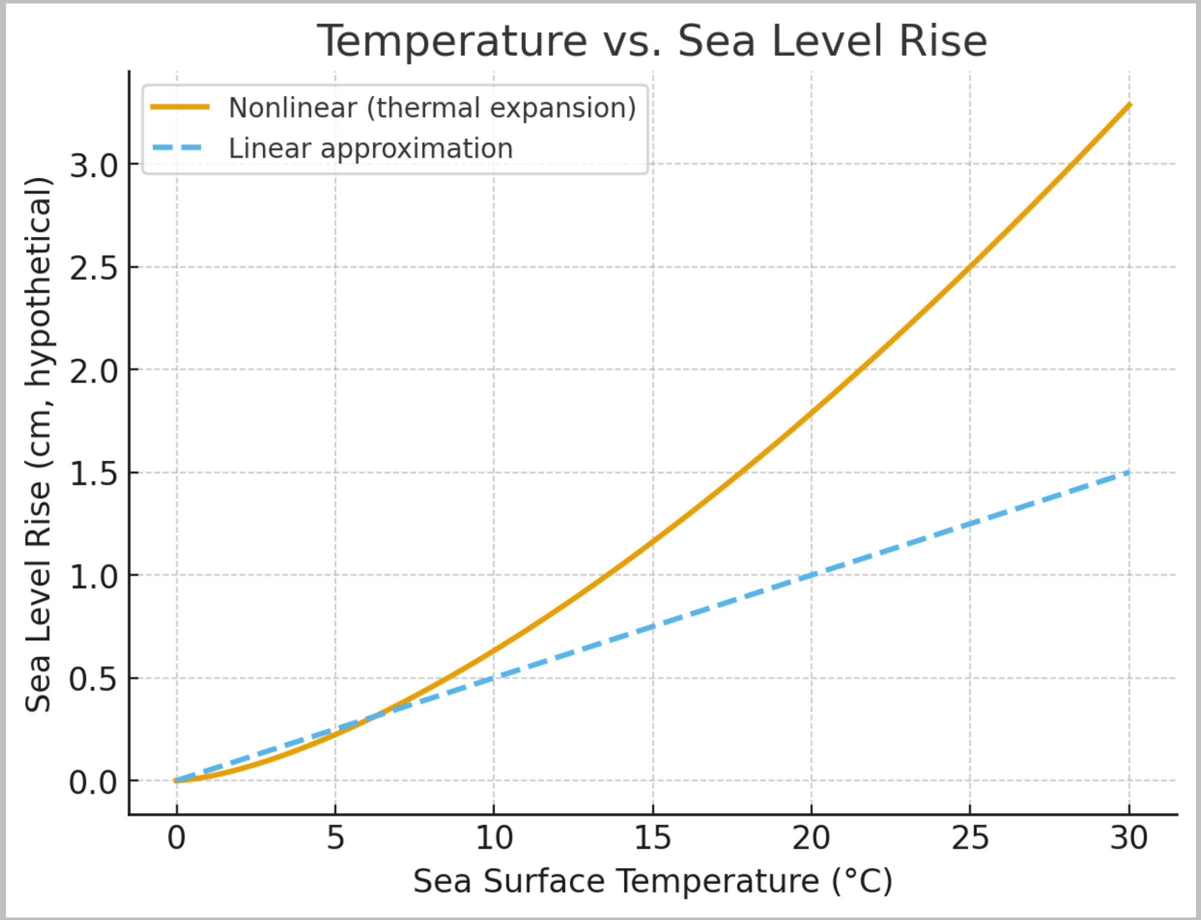
$$I(X;Y) = \Sigma \Sigma\, p(x,y) \cdot \log( p(x,y) / (p(x)p(y)) )$$

$p(x,y)$ = joint probability distribution of X and Y

$p(x),p(y)$ = marginal distributions

If X and Y are independent → $I(X;Y)=0$ (no shared information).

If knowing X perfectly predicts Y→ $I(X;Y)$ is **high**




As **sea surface temperature increases**, the contribution of **thermal expansion** to sea level rise accelerates

# ML/DL CONPONENTS: INPUTS AND METHOD

Various ML/DL approaches can be utilized:

- **Univariate** (i.e. it considers only the target variable) e.g:using traditional ML models such as *linear regression, regression tree, ensemble*

- **Multivariate** frameworks (where several va*deep learning, Convolution Neural Network, random forest, Recurrent Neural Networks (RNNs), and hybrid CNN-RNN models* with respect to the target parameter are considered) e.g

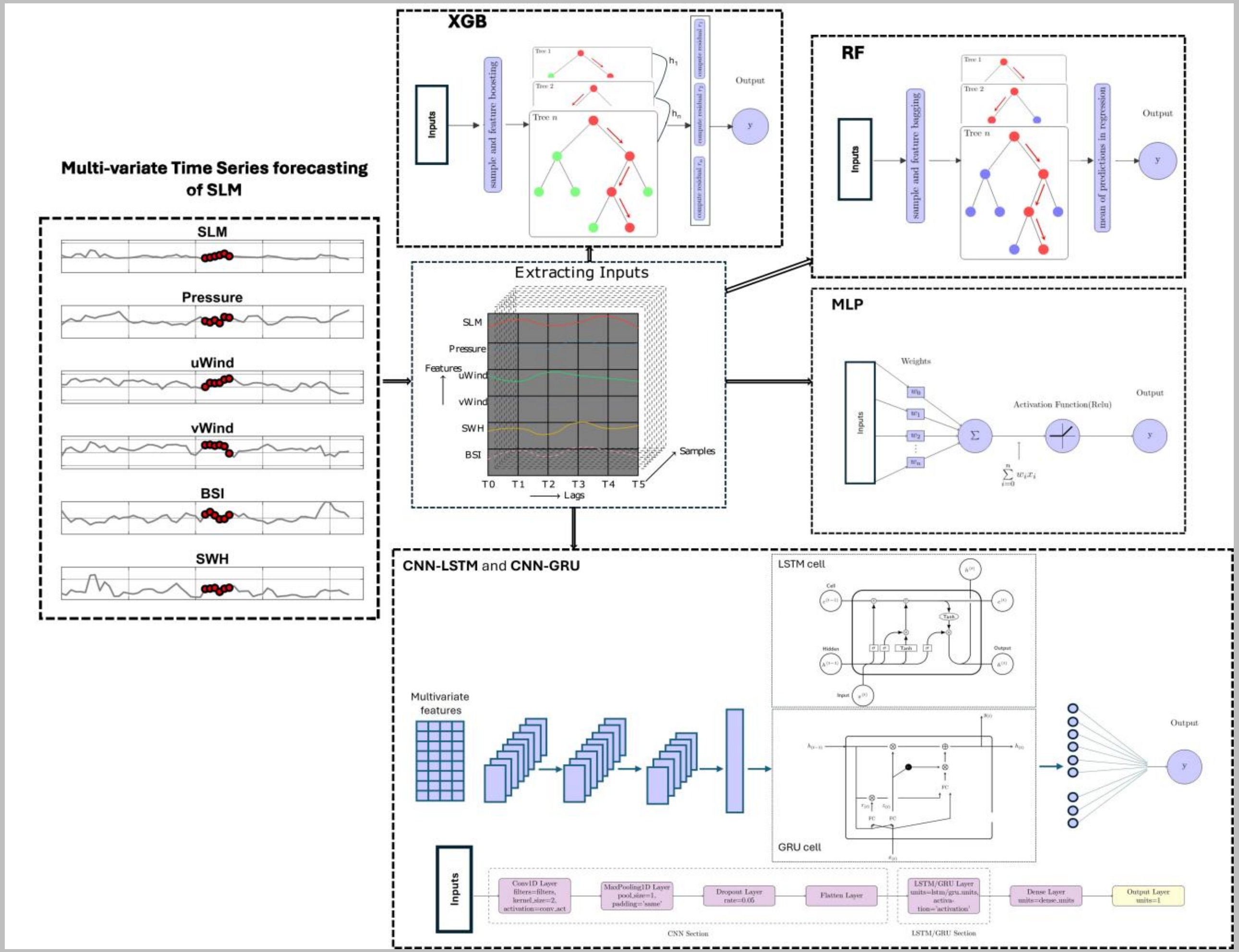
**Multivariate forecasting methods generally outperform univariate models**

**Supervised Learning**:The *model is trained on a labeled dataset*, meaning each input comes with the correct output. Learn a mapping function from inputs (X) to outputs (Y)
*Example: To forecast sealevel every 24 hours*

**Unsupervised learning**: *The model is trained on unlabeled data* — only the inputs (X) are given, without known outputs (Y). Discover hidden structures or patterns in data
*Example: using satellite SST to determine hotspot patterns of hot and cold regions*

# ML METHOD: RANDOM FOREST

- The RF algorithm is based on the bagging (Bootstrap Aggregating) technique.
- Generates multiple decision trees based on random subsets of the data.
- Each tree makes a prediction (like taking votes)
- The forest combines all votes (majority vote for classification, average for regression).



# DL METHOD: CONVOLUTION NEURAL NETWORK

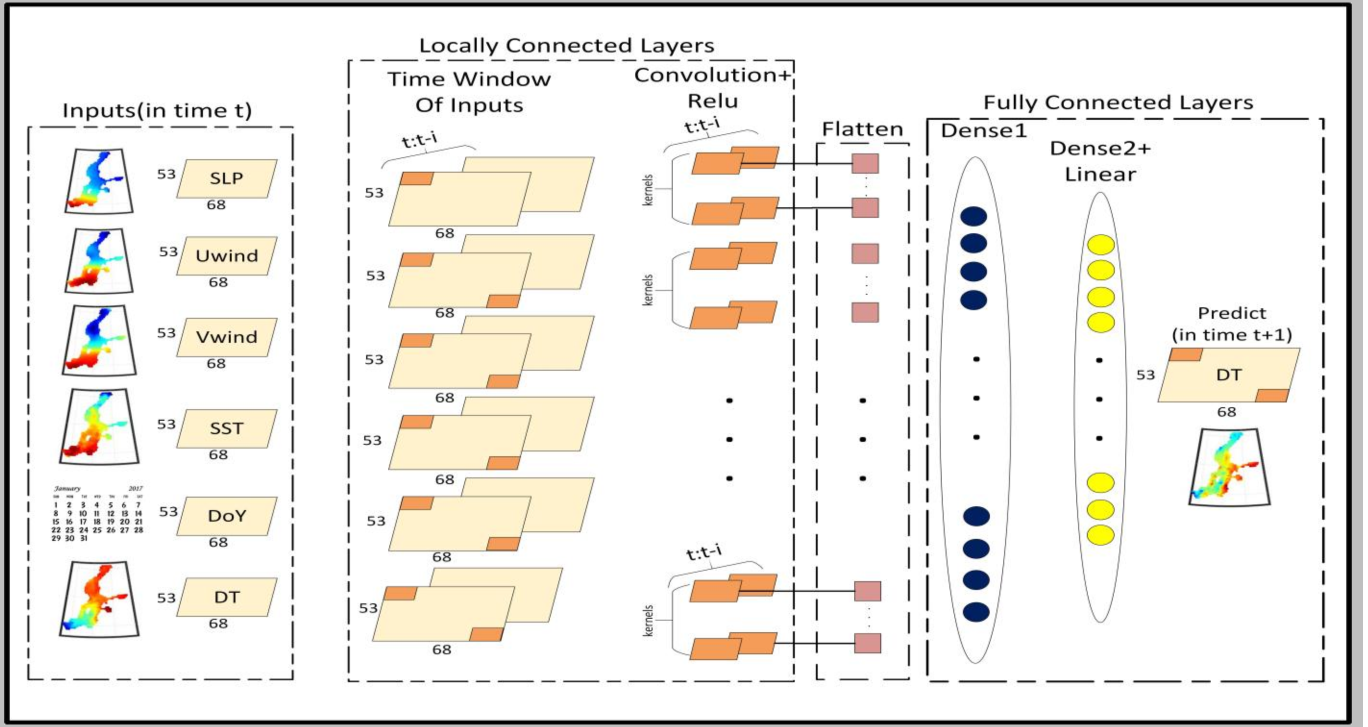
DL methods such as **Convolution Neural Networks (CNN’s) and Recurrent based Neural** Network applied *successfully* in prediction *sea level  tasks*

**Three  primary layers**:

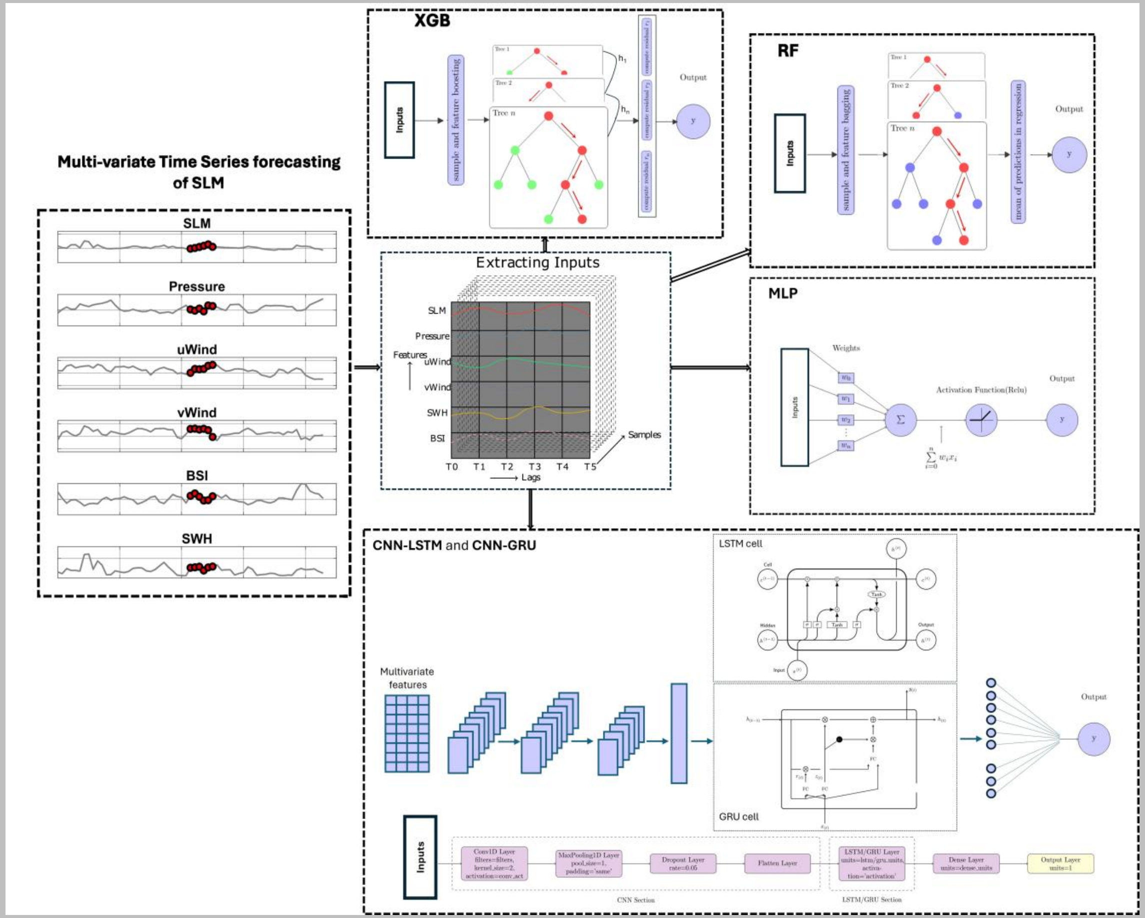
- **Convolution layer:** most critical step usually a linear process. Input data assigned weights and biases (filter or kernel). *ReLu function (for nonlinearity)*
- **Flattening layer:** produce a feature maps
- **Fully connected layer:** flattened data passed to CNN. Models are capable of spatio-temporal connections  and discern between dominating and low-level characteristics

TAL TECH

# METHODS



Locally Connected Layers
Inputs(in time t)
Time Window Of Inputs
Convolution+ Relu
Flatten
Fully Connected Layers
Dense1
Dense2+ Linear
SLP
Uwind
Vwind
SST
DoY
DT
53
68
t:t-i
kernels
Predict (in time t+1)

Multi-variate Time Series forecasting of SLM
SLM
Pressure
uWind
vWind
BSI
SWH
XGB
sample and feature boosting
Inputs
Output
RF
sample and feature bagging
mean of predictions in regression
Extracting Inputs
Features
Samples
Lags
T0 T1 T2 T3 T4 T5
MLP
Weights
Activation Function(Relu)
CNN-LSTM and CNN-GRU
LSTM cell
GRU cell
Multivariate features

# OVERFITTING

Overfitting in machine learning happens when a model learns the training data too well, including its noise, outliers, and random fluctuations, instead of just the underlying patterns.

- *Signs*
  - Performs very well on training data (low training error).
  - Performs poorly on new/unseen data (high test/validation error).

- *Causes of Overfitting*
  - Model is too complex (too many parameters compared to data size).
  - Not enough training data.
  - Training for too many epochs.
  - Including irrelevant features (noisy data).

Example: Studying for an exam if you memorize past questions word for word, you'll ace practice tests but fail on new questions

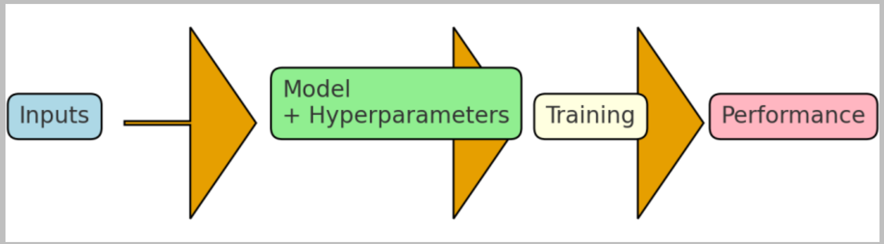
- *Ways to Prevent Overfitting*
  - Simplify the model (reduce parameters, prune trees, etc.).
  - Regularization (dropout, weight decay).
  - Early stopping during training.
  - More training data or data augmentation.
  - Cross-validation to tune hyperparameters.

# HYPERPARAMATERS

- The selection of the hyperparameters influences the model's architecture, training process, and overall performance. This is one of the most important steps to obtain better model accuracy, achieve best possible performance  and can prevent models from overfitting
- Hyperparameters, which are predetermined settings (e.g. learning rate, number of layers, batch size, etc.), must be set before the learning process.

Several optimization approaches for hyperparameter tuning

- trial-and-error
- random search
- grid search
- genetic algorithms (Holland, 1992),
- particle swarm optimization algorithms (Kennedy and Eberhart, 1995),
- Bayesian optimization (BO) (Snoek et al., 2012)



# HYPERPARAMATERS METHODS

| Method | How it Works | Advantages | Disadvantages |
| --- | --- | --- | --- |
| **Trial-and-Error** | Researcher manually tests different hyperparameter values and adjusts based on performance. | - Simple to implement- Uses domain intuition | - Very slow- No systematic exploration- Not scalable |
| **Grid Search** | Tests all combinations of hyperparameters on a predefined grid. | - Systematic- Easy to implement | - Computationally expensive- Inefficient in high dimensions- Wastes trials on unimportant parameters |
| **Random Search** | Randomly samples hyperparameter values from defined ranges. | - More efficient than grid search- Covers more diverse space- Simple automation | - Still requires many trials- No memory of past results |
| **Genetic Algorithms (GA)** *(Holland, 1992)* | Mimics evolution: populations of hyperparameters evolve through selection, crossover, mutation. | - Good at exploring large, complex spaces- Can escape local optima | - Computationally expensive- Many hyperparameters to tune in the algorithm itself |
| **Particle Swarm Optimization (PSO)** *(Kennedy & Eberhart, 1995)* | Models hyperparameters as particles moving through space, guided by best performers. | - Efficient for continuous spaces- Good balance of exploration/exploitation | - Can get stuck in local optima- Sensitive to parameter choices |
| **Bayesian Optimization (BO)** *(Snoek et al., 2012)* | Builds a probabilistic model of performance, chooses next hyperparameters based on expected improvement. | - Very sample-efficient- Finds good hyperparameters with fewer trials- Strong theoretical basis | - More complex to implement- Slower for very high-dimensional spaces |

# EVALUATION

***Root Mean Squared Error (RMSE):*** RMSE provides a measure of the typical prediction error, with higher weight given to larger errors. The formulation is defined as below

***Coefficient of Determination (R-squared):*** R-squared quantifies the proportion of variance in the sea level data that are captured by the model predictions. The formulation is defined as below:

$$RMSE_{(\phi_s,\lambda_s)} = \sqrt{\frac{\sum_{t=1}^{n}\left(\widehat{DT}_{(\phi_s,\lambda_s,t)} - DT_{(\phi_s,\lambda_s,t)}\right)^2}{n}}$$

$$R^2 = 1 - \frac{\sum_{s=1}^{m}\sum_{t=1}^{n}\left(\widehat{DT}_{(\phi_s,\lambda_s,t)} - DT_{(\phi_s,\lambda_s,t)}\right)^2}{\sum_{s=1}^{m}\sum_{t=1}^{n}\left(\widehat{DT}_{(\phi_s,\lambda_s,t)} - \overline{DT}_{(\phi_s,\lambda_s)}\right)^2},$$

$$where\ \overline{DT} = \frac{1}{n}\sum_{t=1}^{n} DT_{(\phi_s,\lambda_s,t)}$$

TAL TECH

# SEA LEVEL FORECASTING: SHORT TERM (HOURS, DAYS)

References

Rajabi-Kiasari, S.; Ellmann, A.; Delpeche-Ellmann, N. (2025). Sea level Forecasting using Deep Recurrent Neural Networks with High-Resolution Hydrodynamic Model. Applied Ocean Research, 157, #104496. DOI: 10.1016/j.apor.2025.104496.

Rajabi-Kiasari, Saeed; Delpeche-Ellmann, Nicole; Ellmann, Artu (2023). Forecasting of absolute dynamic topography using deep learning algorithm with application to the Baltic Sea. Computers & Geosciences, 178, #105406. DOI: 10.1016/j.cageo.2023.105406.

Jahanmard, Vahidreza; Hordoir, Robinson; Delpeche-Ellmann, Nicole; Ellmann, Artu (2023). Quantification of Hydrodynamic Model Sea Level Bias Utilizing Deep Learning and Synergistic Integration of Data Sources. Ocean Modelling, 186, #102286. DOI: 10.1016/j.ocemod.2023.102286.

# FACTORS THAT AFFECT SEA LEVEL

Several components based on different time frames affect the sea level dynamics in the Baltic Sea.

- **Long term (decadal, centuries):**
  - Global sea level change (due to thermal sea water expansion and the melting of glaciers) will influence the Baltic Sea's level
  - Variation in temperature, precipitation, and evaporation is expected to mostly exert influence on a decadal time scale
- **Short-term (yearly, seasonally, daily, etc.):**
  - Major Baltic inflow, meteorological factors such as wind speed, sea level pressure, tides
  - River runoff also affects the water balance, with the biggest freshwater contributor being the Neva River located on the eastern side of the Baltic
  - Sea ice
- **Much shorter time frames (e.g., weekly, daily, and hourly)**
  - Localized events also affect the sea level. Most of these events tend to be influenced by meteorological factors especially the winds
  - Surface waves
  - Storm surges

- **Including relevant components enhances the accuracy and performance of the models**

- **HOWEVER INCLUDING TOO MANY INPUTS CAN LEAD TO OVERFITTING AND REQUIRES INCREASED COMPUTER PROCESSING RESOURCES**

# CASE STUDY: GULF OF FINLAND, BALTIC SEA

- To forecast dynamic topography multi-step time ahead (3h, 6h, 9h, 12h, 24h)

- Several inputs were examined: winds, temperature, salinity, pressure, dynamic topography

$$\widehat{DT}_{(\phi_s,\lambda_s,t+(1:\Delta))} = f\left(Pressure_{(\phi_s,\lambda_s)}, uwind_{(\phi_s,\lambda_s)}, vwind_{(\phi_s,\lambda_s)}, SST_{(\phi_s,\lambda_s)}, SSS_{(\phi_s,\lambda_s)}, DT_{(\phi_s,\lambda_s)}\right)_{(t-w:t)}$$

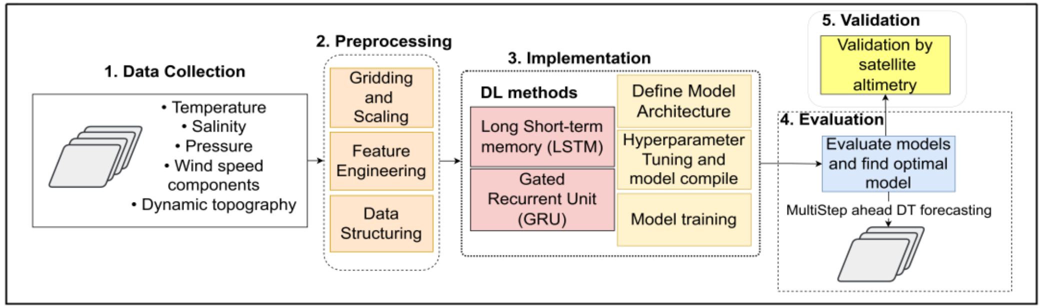
$$, where \;\; s = 1{:}\, number\ of\ grid\ points$$

w, Δ, and $f$ define the temporal lag, the lead time (also called forecast horizon) and the mapping function, respectively

| Variable | Spatial resolution | Temporal resolution | Source |
|---|---|---|---|
| Wind speed (u and v) | 1nm | Hourly | Nemo Nordic |
| Surface Pressure | 0.25°×0.25° | Hourly | Era5 |
| Sea Surface temperature | 1nm | Hourly | Nemo Nordic |
| Sea Surface Salinity | 1nm | Hourly | Nemo Nordic |
| Dynamic Topography | 1nm | Hourly | Corrected Nemo Nordic |
| Sea Surface Height | 300m | 27 days revisiting time, 20Hz data at each pass | Along-track Sentinel 3A and 3B (EUMETSAT) |

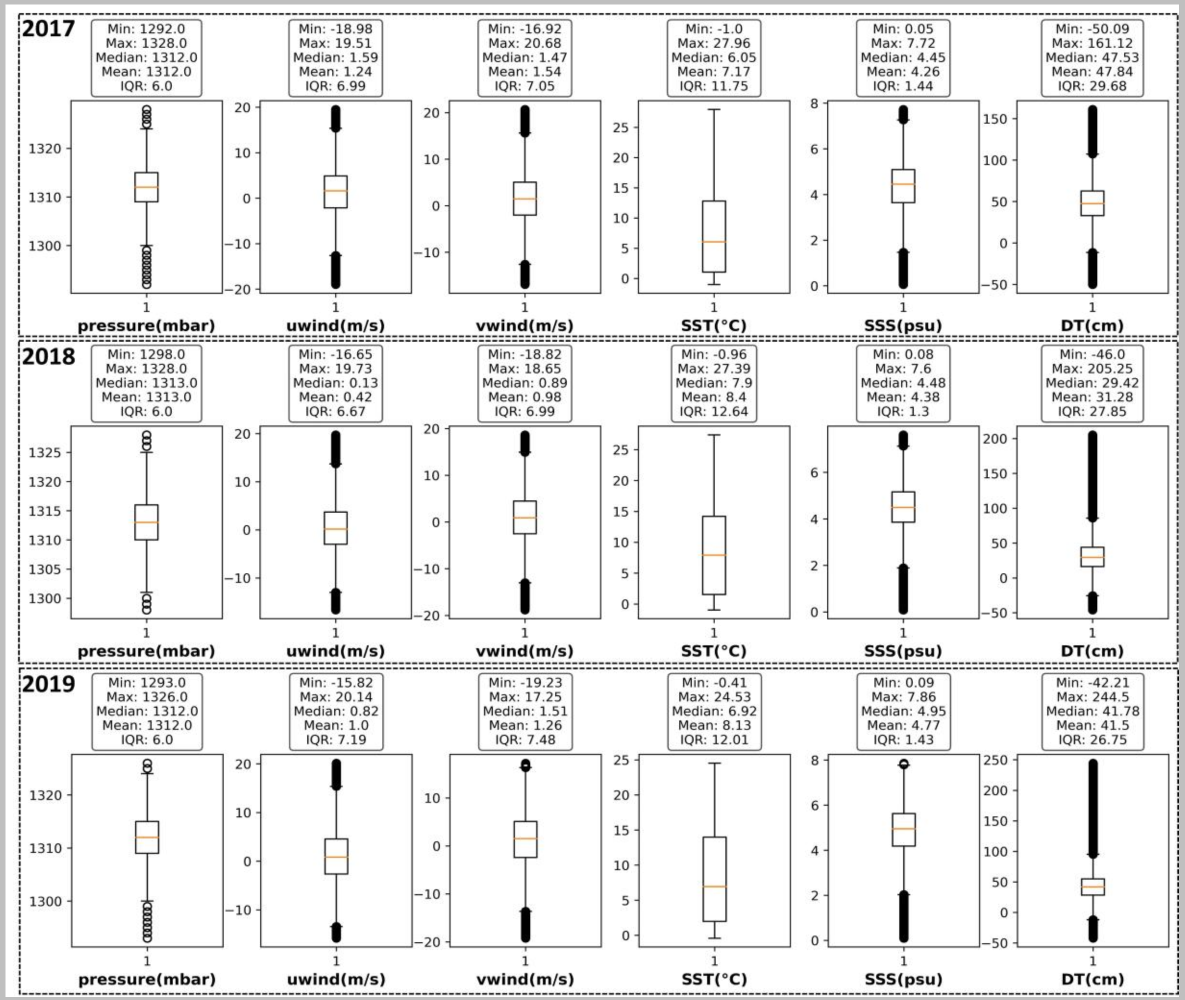
DT has been referred to the European Vertical Reference System (EVRS) which the tide gauges and geoid model referred to

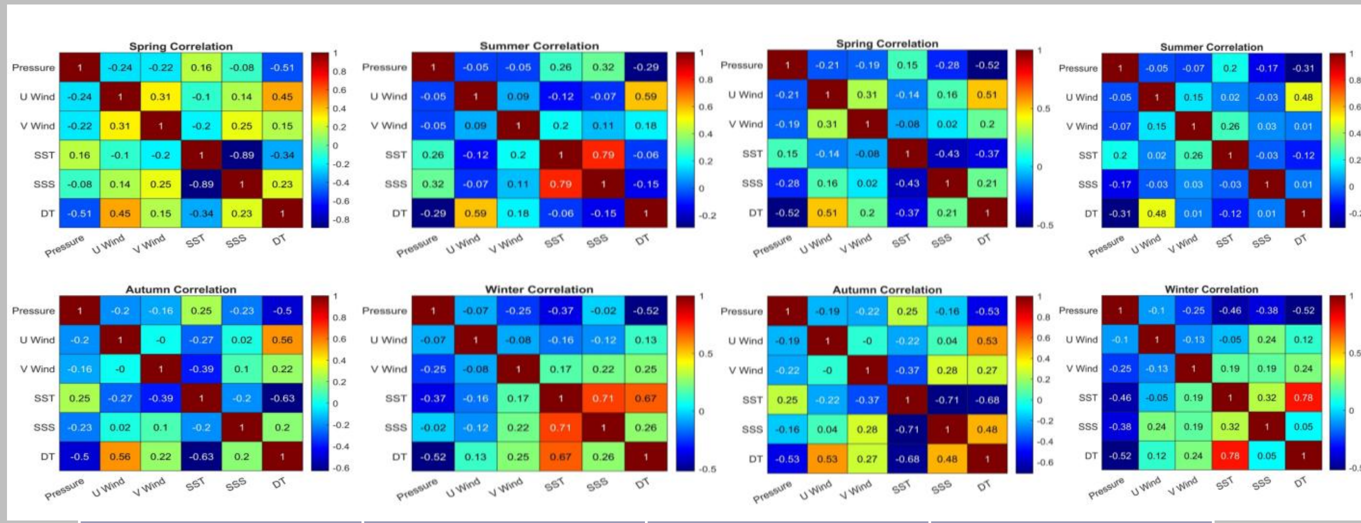
Data 2017 to 2019 (85% for train and 15% for test)
Train data : 2017-01-01 to 2019-07-20
Test data: 2019-07-21 to 2019-12-30
External validation SA: 2019-07-21 to 2019-12-30



# CASE STUDY: CHOOSING INPUTS

- Box plots of mean, median, extremes, IQR
- Pearson correlation coefficients were calculated for all input components across distinct seasons (spring, summer, winter, autumn)

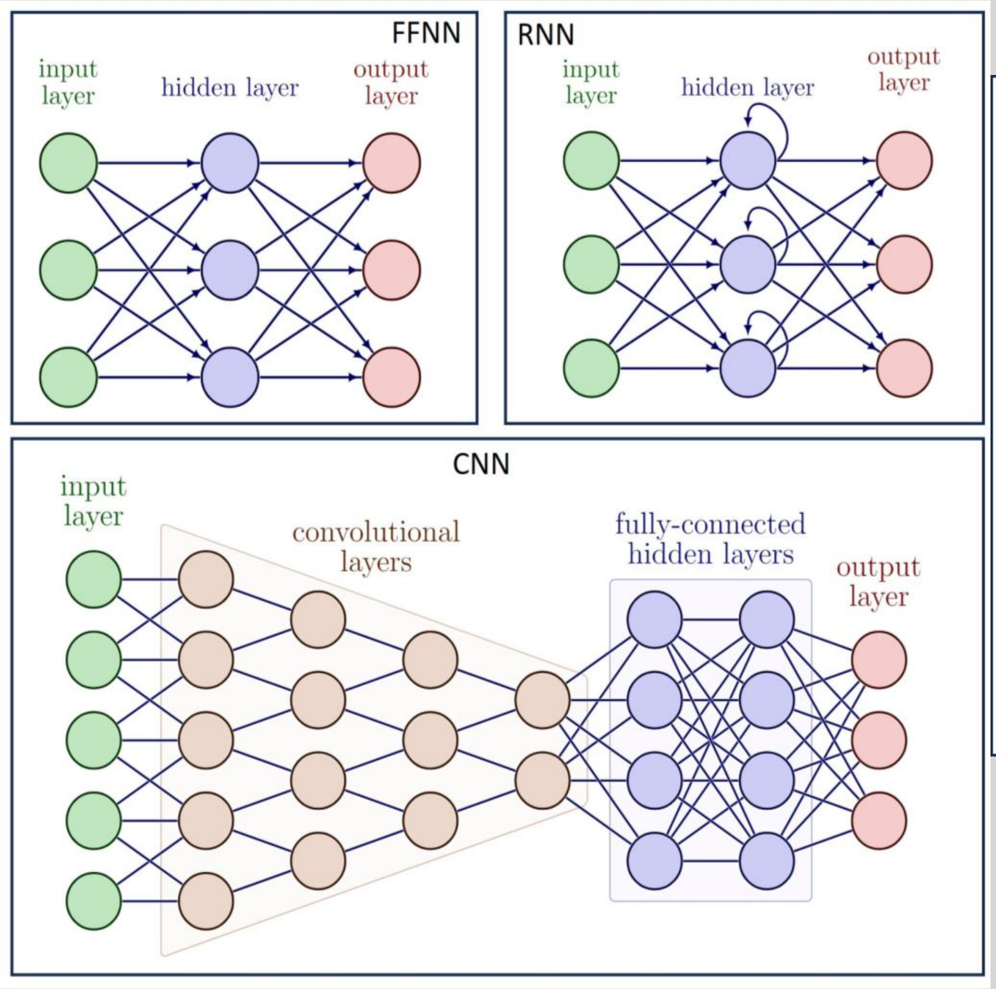


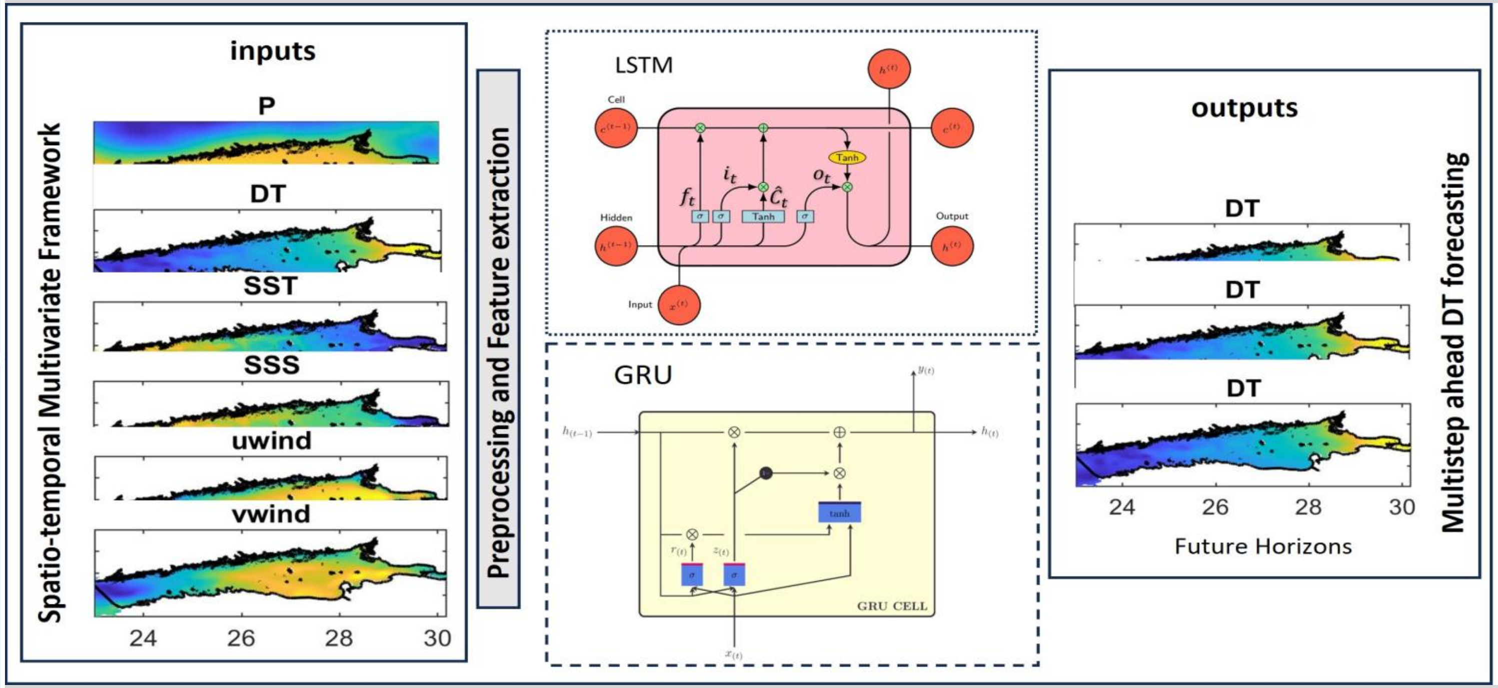




| Variable | Spatial resolution | Temporal resolution | Source |
|---|---|---|---|
| Wind speed (u and v) | 1nm | Hourly | Nemo Nordic |
| Surface Pressure | 0.25°×0.25° | Hourly | Era5 |
| Sea Surface temperature | 1nm | Hourly | Nemo Nordic |
| Sea Surface Salinity | 1nm | Hourly | Nemo Nordic |
| Dynamic Topography | 1nm | Hourly | Corrected Nemo Nordic |
| Sea Surface Height | 300m | 27 days revisiting time, 20Hz data at each pass | Along-track Sentinel 3A and 3B (EUMETSAT) |

# CASE STUDY: DL MODELS

Two recurrent neural network-based models such as the Long Short-Term Memory Networks (LSTMs), and the Gated Recurrent Unit (GRU)



# CASE STUDY: HYPERPARAMATER OPTIMIZATION

- Trial-and-error method

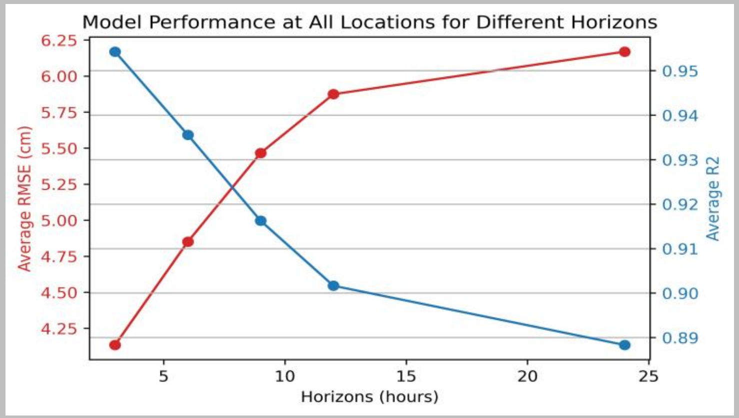
| Model parameters | Description | Chosen hyperparameter |
|---|---|---|
| **LSTM/GRU Units** | Specifies the dimensionality of the model's internal state. | 512 |
| **Activation Functions** | The activation function is applied after each layer in the model to add nonlinearity. The common choice for RNNs is 'Tanh' and Sigmoid. | 'default' |
| **Batch Size** | Determines the number of samples used in each forward and backward pass during training. | 128 |
| **Number of Training Epochs** | Specifies how many times the model will be exposed to the entire training dataset during training. | 50 |
| **Loss Function** | Determines the objective function that the model is trying to minimize during training. | 'MSE' |
| **Optimizer** | The optimizer determines the specific algorithm used to update the model's weights during training. Common optimizers include Adam, RMSprop, and Stochastic Gradient Descent (SGD). | 'Adam' |
| **Dropout Rate Regularization** | A regularization technique that helps prevent overfitting. It specifies the proportion of neurons or units that are randomly dropped out during training, forcing the model to be more robust. | 0.1 |
| **Kernel Regularization** | Technique used to limit the model's weights with certain values. It adds a penalty term to the loss function based on the magnitude of the weights. Common regularization techniques include L1 and L2 regularization. The regularization Strength hyperparameter controls the strength of the kernel regularization | L2, 0.01 |

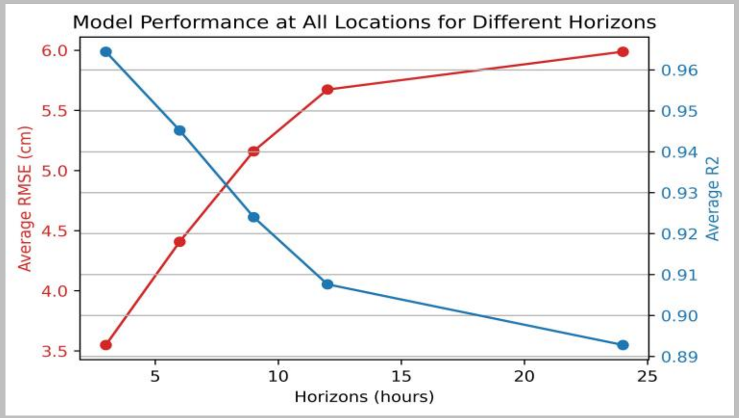
TAL TECH

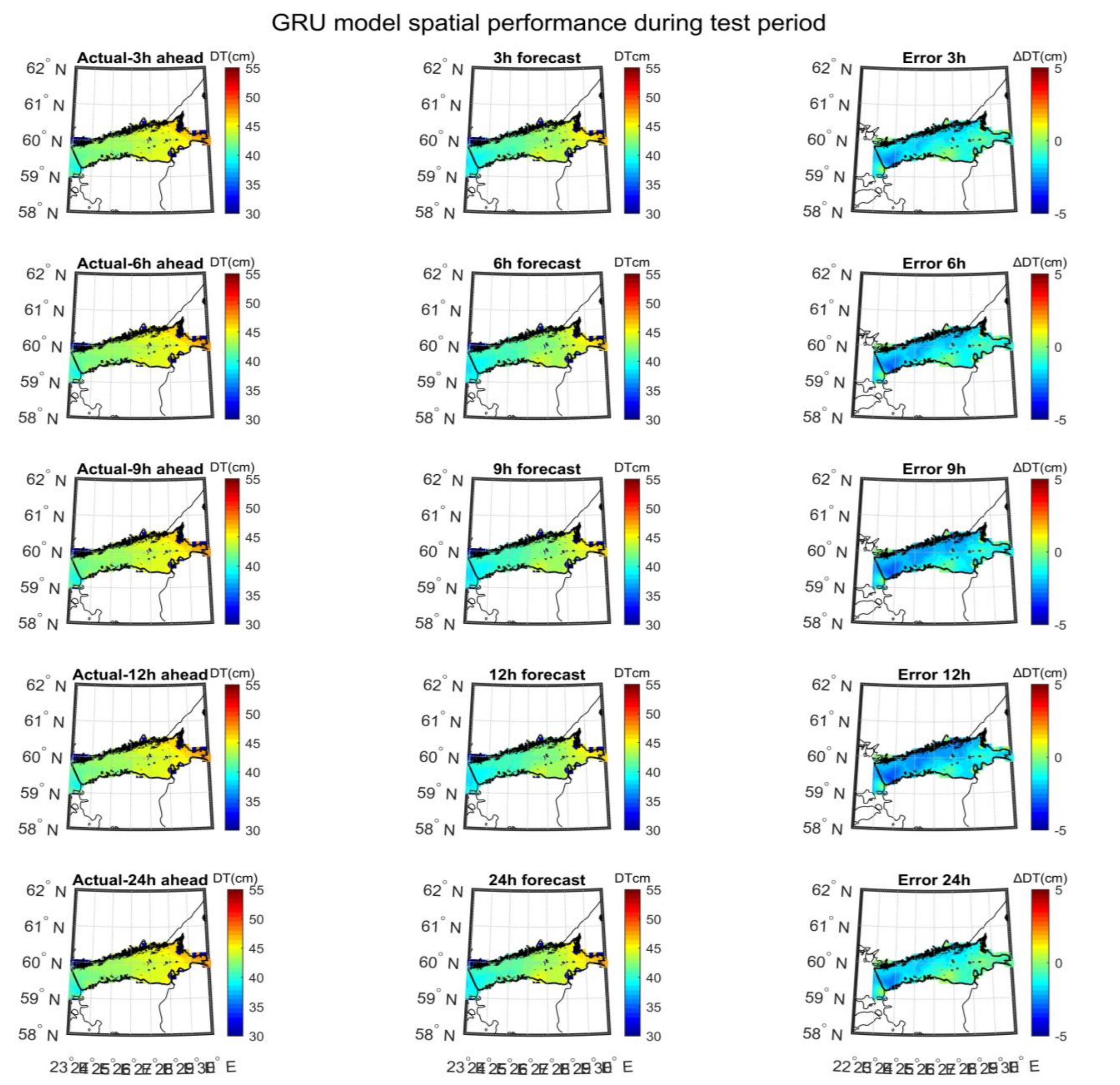
# CASE STUDY: RESULTS (OVERVIEW)

- Both LSTM and GRU methods are strong choices for sea level forecasting with RMSE <6 CM. GRU performed slightly better with R2 and RMSE of 0.93, 4.96 cm
- Main difference between the LSTM and GRU model was that the GRU model has a simpler method in storing and updating the connections between the different variables resulting in fewer complexities and less computing time.

| Horizons (hours) | Models | | | |
|---|---|---|---|---|
| | GRU | | LSTM | |
| | $R^2$ | RMSE (cm) | $R^2$ | RMSE (cm) |
| **3** | 0.96 | 3.55 | 0.95 | 4.13 |
| **6** | 0.95 | 4.41 | 0.94 | 4.85 |
| **9** | 0.92 | 5.16 | 0.92 | 5.47 |
| **12** | 0.91 | 5.67 | 0.90 | 5.87 |
| **24** | 0.89 | 5.99 | 0.89 | 6.17 |
| **average** | 0.93 | 4.96 | 0.92 | 5.3 |



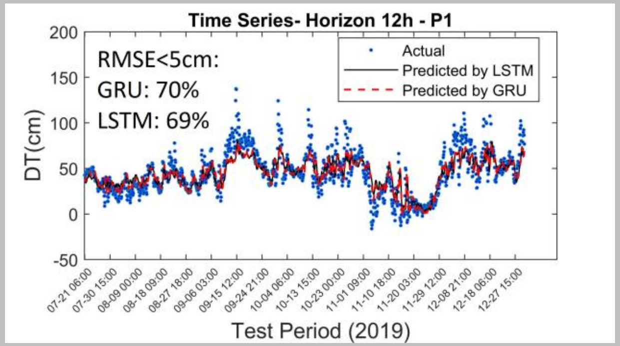



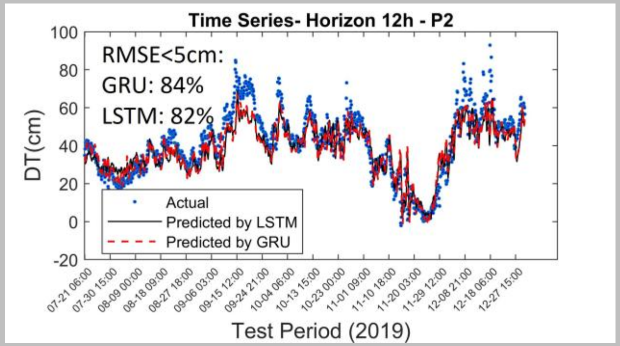


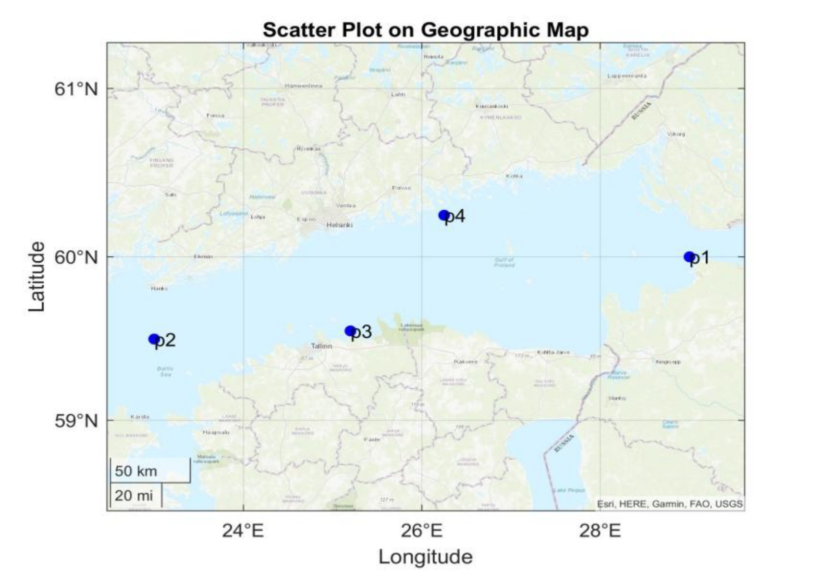

- The input component (v winds, sea surface salinity, river discharge) were not included in the final variable selection

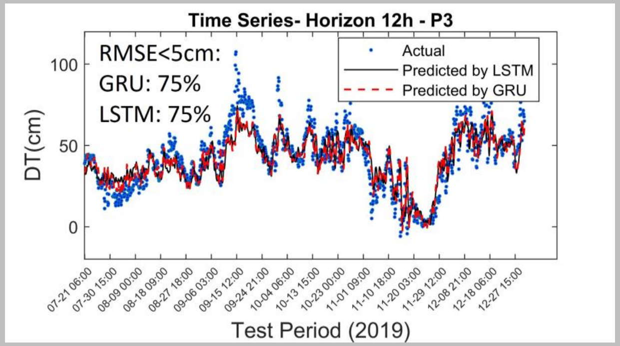
- This exclusion may have contributed to the poorer performance experienced at eastern and other sections

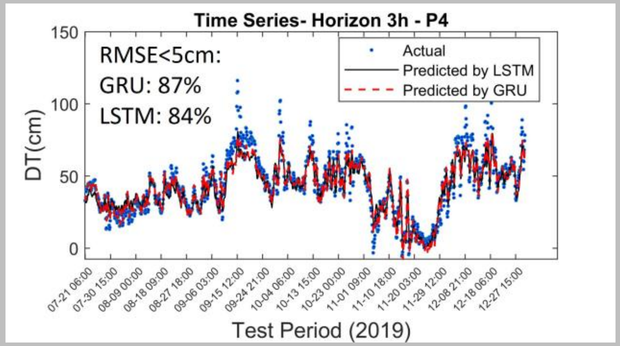
# CASE STUDY: SPECIFIC SITE RESULTS



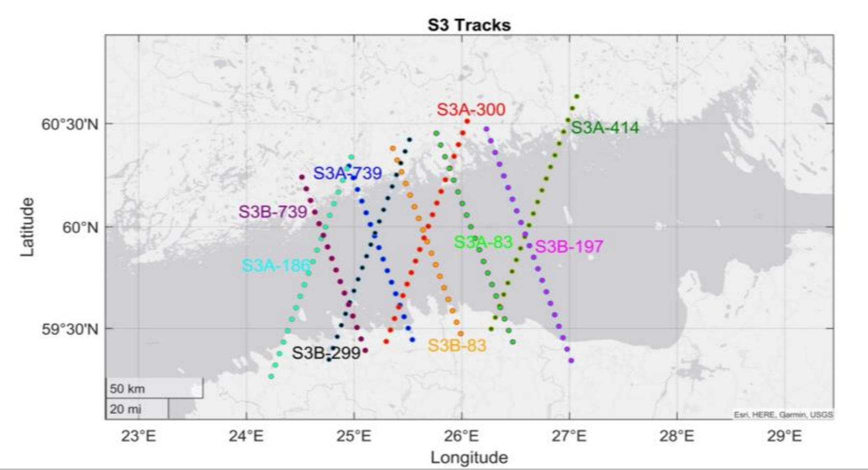



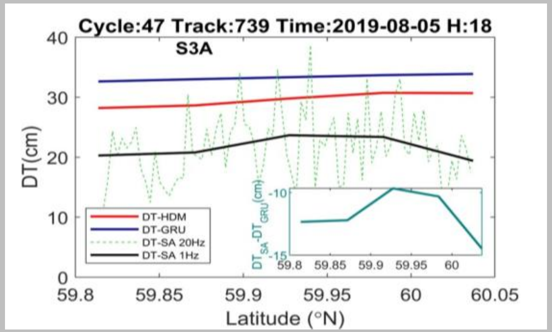



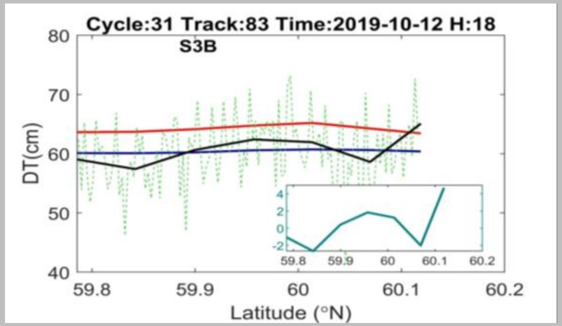




- Both methods forecasted the normal sea level very good
- Both experience difficulties with the sea level maxima/extremes
- Insufficient representation of maximum/extreme events in the training dataset (skewness towards normal sea levels than extremes)

# CASE STUDY: EXTERNAL TEST DATA WITH SATELLITE ALTIMETRY




- External test data was performed with S3A and S3B
- GRU-forecasted DTs and the HDM DT are for most occasions in good agreement with SA DT values, with the discrepancy of lower than 5 cm for tracks S3A-83, S3A-300, S3A-414, S3B-83 and S3B-197.
- However, the GRU model had poorer validation results for tracks S3A-739, S3A-186, S3B-739, and S3B-299 (10-15 cm).
- The reason for these larger discrepancies may be due to:
  - HDM model not accurately modelling the observed ocean dynamics.
  - HDM corrected DTs had better consistency with Sentinel 3A tracks compared to the Sentinel 3B, which is also in agreement with previous results (Mostafavi et al., 2023).

TAL TECH

# FORECASTING OF SEA LEVEL EXTREMES (SHORT TERM & LONG TERM)

Reference
Rajabi-Kiasari, S.; Ellmann, A.; Delpeche-Ellmann, N. Soomere, T. (submitted, Under review ). Forecasting Sea Level Maxima using Machine Learning with Explainability and Extreme Value AnalysisSea level Forecasting using Deep Recurrent Neural Networks with High-Resolution Hydrodynamic Model, International Journal of Applied Earth Observation and Geoinformation

TAL TECH

# MACHINE LEARNING AND DEEP LEARNING

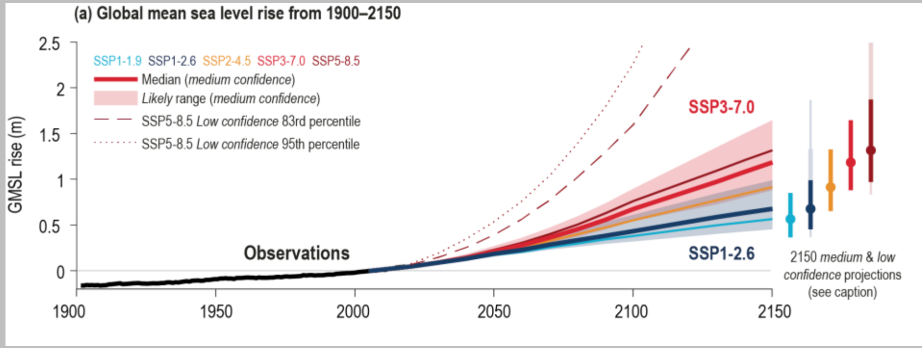
- ***Sea level maximas and extremes (SLM)*** are a major contributor of coastal flooding, erosion, infrastructure damage etc.

- The ***SLM are often characterized*** as (i) ***occurring suddenly*** and usually ***having a time scale from minutes*** (rogue waves, edge waves) to ***a few days*** (storm surges); (ii) ***being site-specific rather than basin-wide*** (Pindsoo et al., 2020); (iii) primarily ***driven*** by very strong ***storms***

- ***Semi-enclosed sea areas*** such as (Baltic Sea, Meditteranean, Caspian Sea) ***most at risk for SLM***, most impactful on coastal areas that affects several countries.

- **Influenced by compound events** such as waves, tides storms that influence each other

West gate on Sea, England (photo credit: Max Montagut, www.flickr.com)

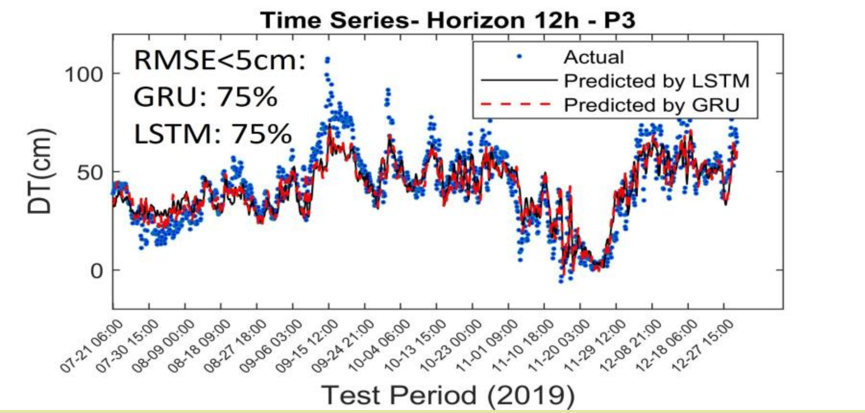


# CHALLENGES

**Challenges:**

- ***Machine/Deep Learning (ML/DL) models*** have been shown to be efficient in forecasting mean sea level,
- ***ML/DL models often under-estimate sea level maxima/extremes and there exist uncertainty on the influence of the drivers.*** Possible reasons:

  - ***Lack of adequate representation*** of extreme events in training data
  - Selecting of best **hyperparameters** and **optimizing models** are crucial steps in developing ML/DL models for capturing complex peak patterns *(Li et al., 2024).*
  - ***Compound events***, whereby some inputs not considered in model
  - ***Some extreme conditions such as storm surges, seiches*** due to their frequency and complexity are ***challenging to model***



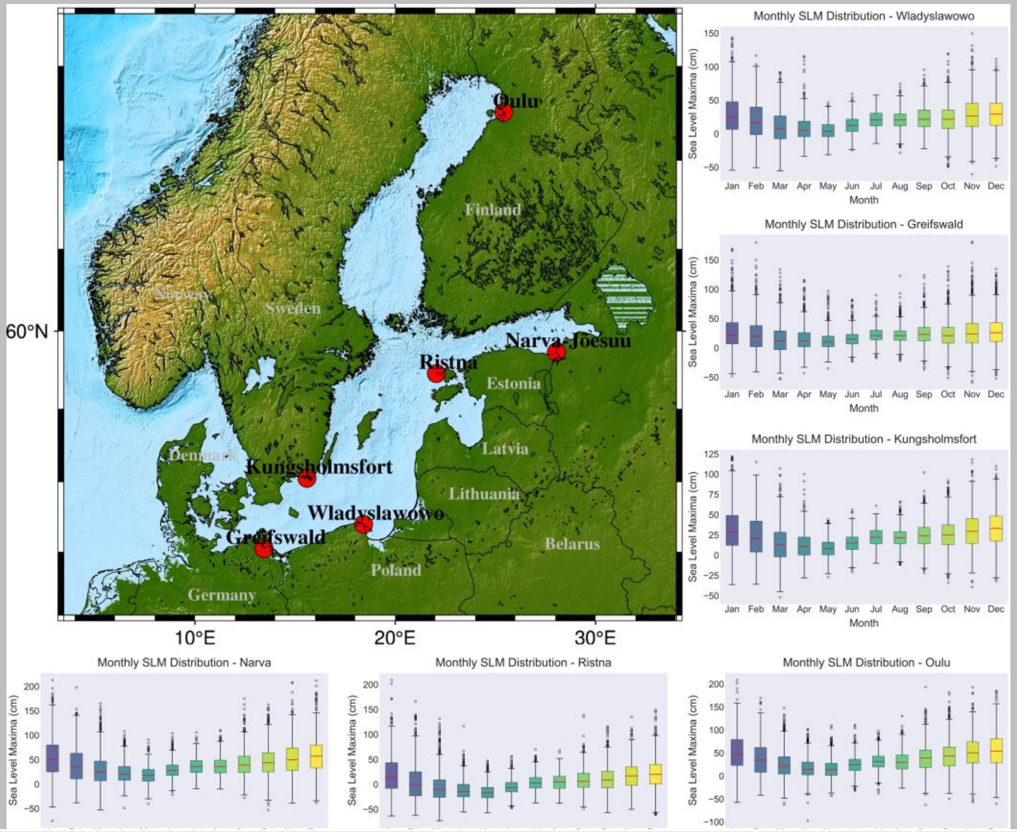

**Opportunities/Objectives:**

- ***Machine/Deep Learning approaches*** that can specifically ***examine SLM***
- ***Deeper insight into the role of the drivers*** influencing these extremes by using ***explainability analyis***
- ***Linking ML/DL results with traditional methods*** such as extreme value analysis gives deeper insight into the long-term forecasting

# CASE STUDY: EXTREMES BALTIC SEA

***SLM on Baltic coasts*** occur at ***different locations*** with ***different influential forces*** which can be due to:

- *Initial sea Level* (filling-up or prefilling) of the Baltic Sea
- *Wind Stress*: magnitude, direction and duration
- *Low-Pressure Systems*: Storms
- *Other factors*: surface waves, water exchange between the Baltic and the North Sea, precipitation, seasonal changes in water density, and the occurrence of seiches (*Weisse and Weidemann, 2017*)




**Characteristics:**

- Typical ***SLM*** in the Baltic Sea is ***0.8 m***
- ***SLM*** in the Baltic Sea are more pronounced in the ***winter season*** due to the seasonal cycle of wind
- **Wave set up** may influence the ***SLM***
- ***Maximum SWH of 8.2 m*** was recorded in December 2004 in the northern Baltic Proper

- ***Six tide gauges*** stations selected: ***Narva, Ristna, Oulu, Kungsholmsfort, Greifswald, Wladyslawowo***
- Data between ***1971 to 2022***. All data are referred to BSCD 2000 indicating vertical reference compatability
- Relative Sea level utilized (Land uplift correction not applied)
- ***Gaps in TG data*** filled by using ***bilinear interpolation***

| tion | Latitude (°N) | Longitude (°E) | Country | Datum | Missing data rate | Conversion to BSCD2000 (cm) | Source |
|---|---|---|---|---|---|---|---|
| arva | 59.4691 | 28.0421 | Estonia | EH2000 | 0.1% | -500 | EEA (2024) |
| stna | 58.9212 | 22.0552 | Estonia | EH2000 | 0.3% | -500 | EEA (2024) |
| ulu | 65.0403 | 25.4182 | Finland | N2000 | 0 | - | FMI (202) |
| fswald | 54.0928 | 13.446 | Germany | DHHN92 | 0 | -496.9 | BSH (2024) |
| slawowo | 54.7968 | 18.4187 | Poland | PL-EVRF2007-NH | 0.5% | -494.4 | BOOS (2024) |
| Kungsholmsfort | 56.1053 | 15.5894 | Sweden | RH2000 | 0 | - | SMHI (2024) |

# FEATURE SELECTION RESULTS

- ***Initial feature selection:*** *wind speed (zonal, meridional and gust), surface atmospheric pressure, evaporation, precipitation, river runoff, Baltic Sea Index, significant wave height*

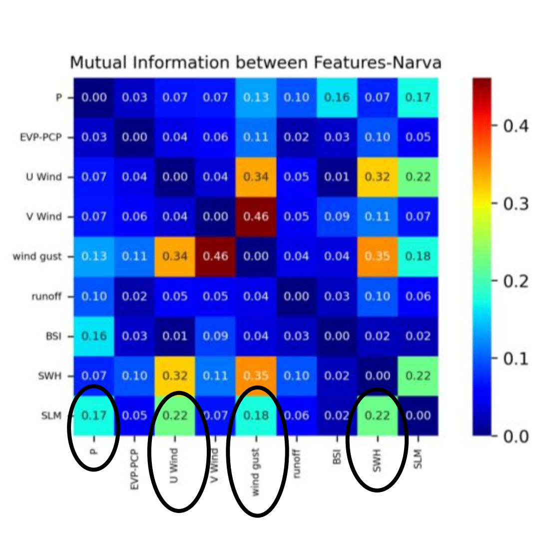
- **Mutual information (MI) index** to discover the influential parameters

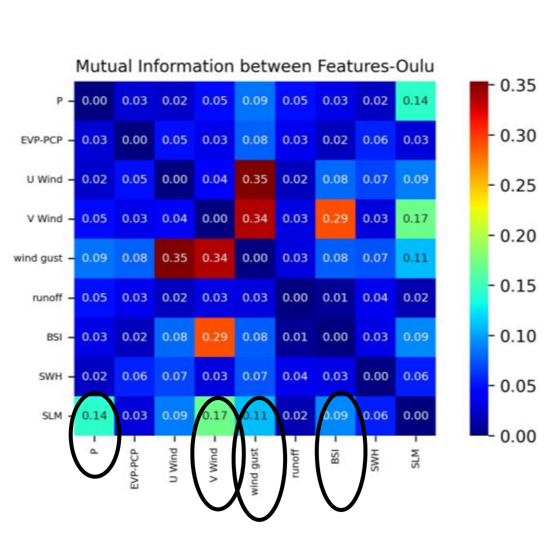
- ***Uwind, Vwind, SWH, BSI and P*** were selected as the basic features for all stations.

- ***Bayesian Information Criterion (BIC)*** index for each station separately identified: the previous timesteps to consider

| Variable | Units | Source | Statistics | | |
|---|---|---|---|---|---|
| | | | Min | Mean | Max |
| **Zonal Wind speed** | m/s | Era5 | -16.33 | 3.53 | 24 |
| **Meridional Wind speed** | m/s | Era5 | -14.52 | 3.21 | 20.41 |
| **Wind gust** | m/s | Era5 | 1.96 | 11.11 | 35.82 |
| **Surface atmospheric pressure** | Mbar | Era5 | 944.98 | 1008.2 | 1052.2 |
| **Significant wave height** | m | SWAN and WAM | 0 | 0.89 | 7.31 |
| **Evaporation minus Precipitation** | m | Era5 | -0.00090 | -0.000027 | 0.00019 |
| **Surface runoff** | m | Era5 | -4.34e-19 | 1.0196e-06 | 0.00082 |
| **Baltic Sea Index** | - | Era5 | -1.6192 | 0.2302 | 2.6213 |
| **Sea level** | cm | TGs | -97.5 | 23.2914 | 213 |

**Impacts of different features on Sea level maxima using mutual index**



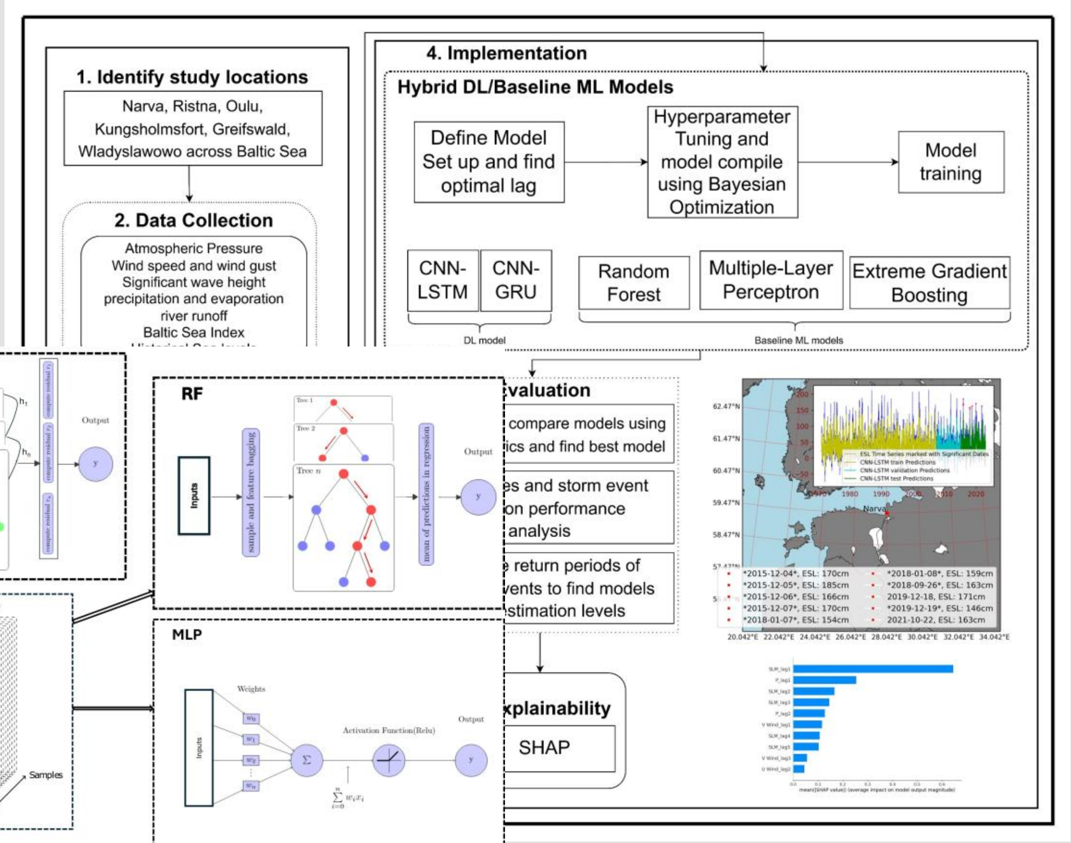




| stations | Narva | Ristna | Oulu | Kungsholmsfort | Wladyslawowo | Greifswald |
|---|---|---|---|---|---|---|
| **Selected features (MI)** | u,v,p,swh, wind gust | u,v,p,swh | u,v,p,swh, wind gust, BSI | u,v,p,swh, wind gust, BSI | u,v,p,swh, BSI | u,v,p,swh, wind gust, BSI |
| **Optimal lag (BIC)** | 5 | 3 | 5 | 5 | 5 | 5 |

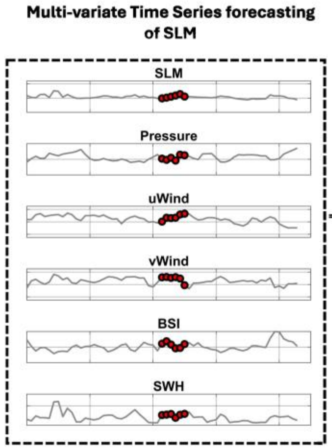
# PROPOSED STRATEGY FOR SEA LEVEL MAXIMA FORECASTING

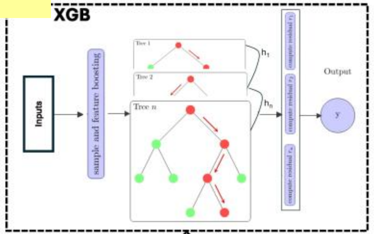
- Five ML/DL methods: ***Random Forest (RF), Extreme gradient boosting (XGB), Multi-layer perceptron (MLP) neural network, CNN-LSTM***
- ***Hyperpramater tuning*** (learning rate, number of layers, batch size, etc.): ***Bayesian Optimization*** configured with 50 iterations
- Other: ***'Adam' as the optimize***; ***loss function as Mean Squared Error (MSE)***

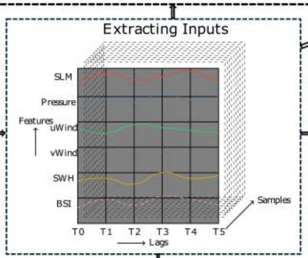
***training period:*** *1971-01-01 to 2007-05-27*
***validation period:*** *2007-05-28 to 2015-03-14*
***test period:*** *2015-03-15 to 2022-12-31*

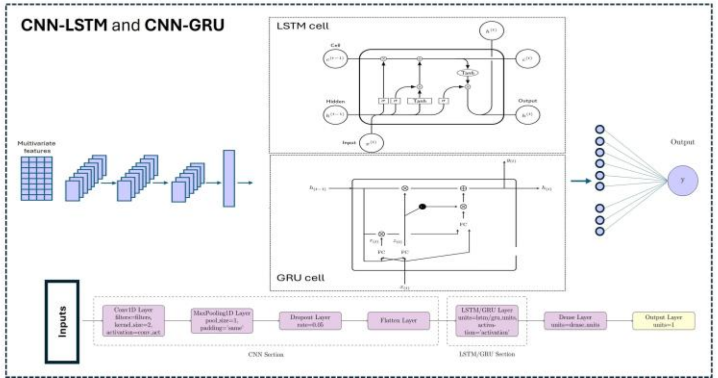



| Models | Hyper-parameters | Definitions | Ranges | Optimized value | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Narva-Jõesuu | Ristna | Oulu | Kungsholmsfort | Wladyslawowo | Greifswald |
| MLP | Number of hidden neurons | Number of neurons in hidden layers, controls model complexity | (10, 100) | 24 | 100 | 10 | 16 | 19 | 10 |
| | alpha | Regularization term to prevent overfitting | (0.01, 0.05) | 0.0221 | 0.0458 | 0.016 | 0.01 | 0.0437 | 0.037 |
| | learning_rate_init | Starting learning rate, controls how fast the model learns | (0.01, 1) | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| RF | Number of Trees (n_estimators) | Number of trees in the forest showing model complexity | (50, 100, 150, 200, 300) | 300 | 150 | 300 | 100 | 300 | 300 |
| | Tree Depth (max_depth) | Maximum depth of each tree | (3, 5, 7, 10) | 10 | 7 | 10 | 10 | 10 | 10 |
| | min_samples_split | Minimum samples required to split a node | (2, 5, 10, 20) | 20 | 2 | 2 | 2 | 5 | 2 |
| XGB | Number of Trees (n_estimators) | Number of boosting rounds (trees) | (50, 100, 150, 200, 300) | 150 | 100 | 150 | 200 | 200 | 200 |
| | Tree Depth (max_depth) | Maximum depth of trees | (3, 5, 7, 10) | 7 | 3 | 5 | 5 | 3 | 3 |
| | Learning rate | Controls the size of each step during training | (0.01, 1) | 0.1485 | 0.179 | 0.06 | 0.129 | 0.0176 | 0.124 |
| CNN-LSTM | Number of filters | Number of convolution filters, determines feature extraction | (8, 128) | 10 | 17 | 8 | 8 | 8 | 8 |
| | activation function | Function used to activate neurons (e.g., ReLU) | ['ReLU', 'tanh', 'Leaky ReLU] | 'tanh' | 'Leaky ReLU' | 'Leaky ReLU' | ' tanh' | 'tanh' | 'tanh' |
| | Dense units | Number of neurons in the fully connected layer | (16, 128) | 124 | 125 | 16 | 58 | 107 | 16 |
| | LSTM units | Number of units in LSTM layer, controls memory capacity | (16, 128) | 16 | 89 | 16 | 16 | 16 | 16 |
| CNN-GRU | Number of filters | Number of convolution filters | (8, 128) | 8 | 126 | 8 | 8 | 53 | 8 |
| | activation function | Activation function for neurons | ['ReLU', 'tanh', 'Leaky ReLU] | 'Leaky ReLU' | 'Leaky ReLU' | 'tanh' | 'tanh' | 'Leaky ReLU' | 'Leaky ReLU' |
| | Dense units | Neurons in the fully connected layer | (16, 128) | 16 | 35 | 16 | 16 | 16 | 128 |
| | GRU units | Number of units in the GRU layer, controls memory | (16, 128) | 16 | 16 | 26 | 16 | 16 | 16 |

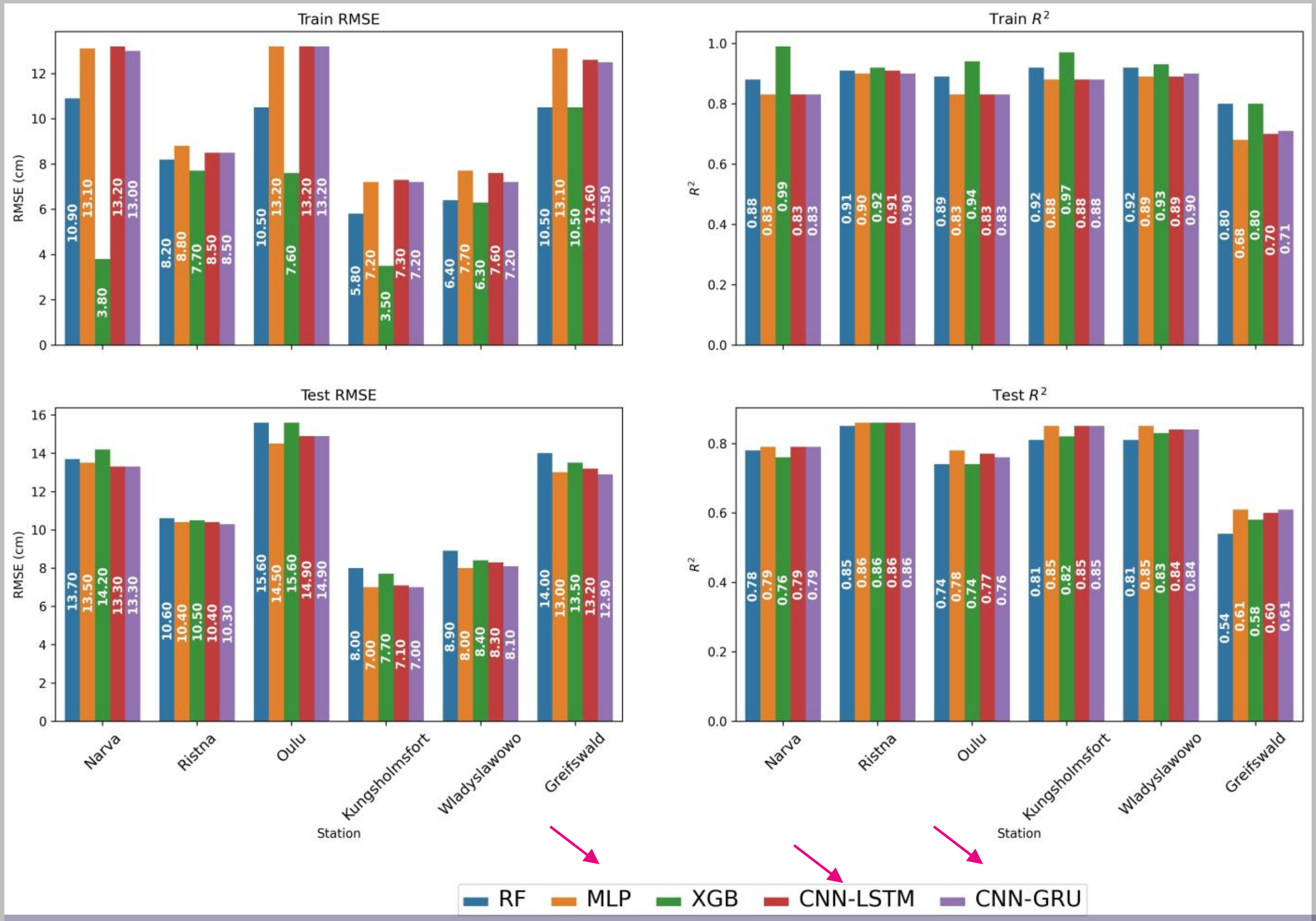


# RESULTS: MODEL PERFORMANCE

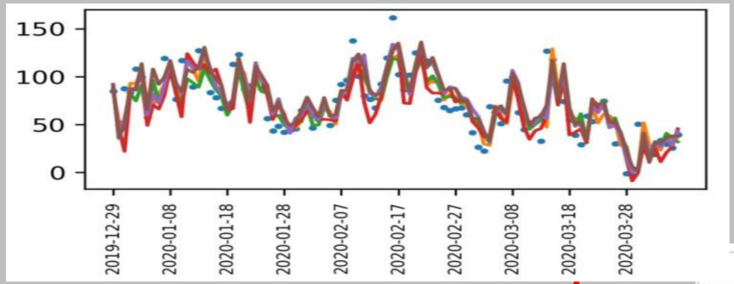


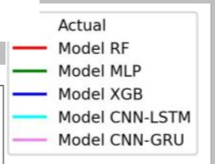


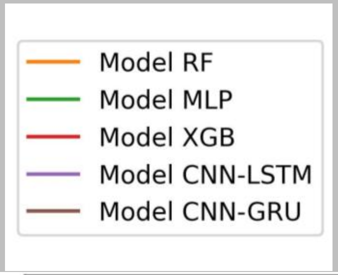
- Neural-network-based models ***MLP, CNN-GRU, and CNN-LSTM*** demonstrated ***better generalization capabilities***

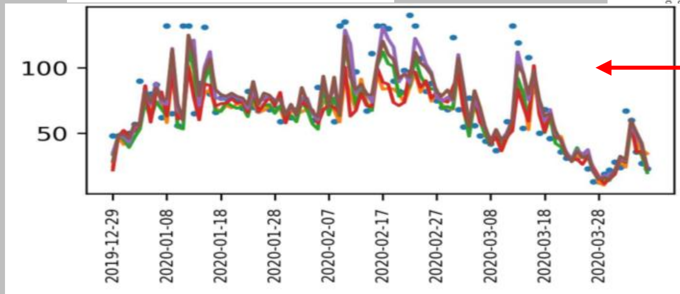
- ***RF and XGB models*** exhibited ***signs of overfitting***: ***drop in R² scores; increase in RMSE from training to test*** for XGB at Narva-Jõesuu, Oulu, and Kungsholmsfort, as well as for RF at Oulu and Wladyslawowo

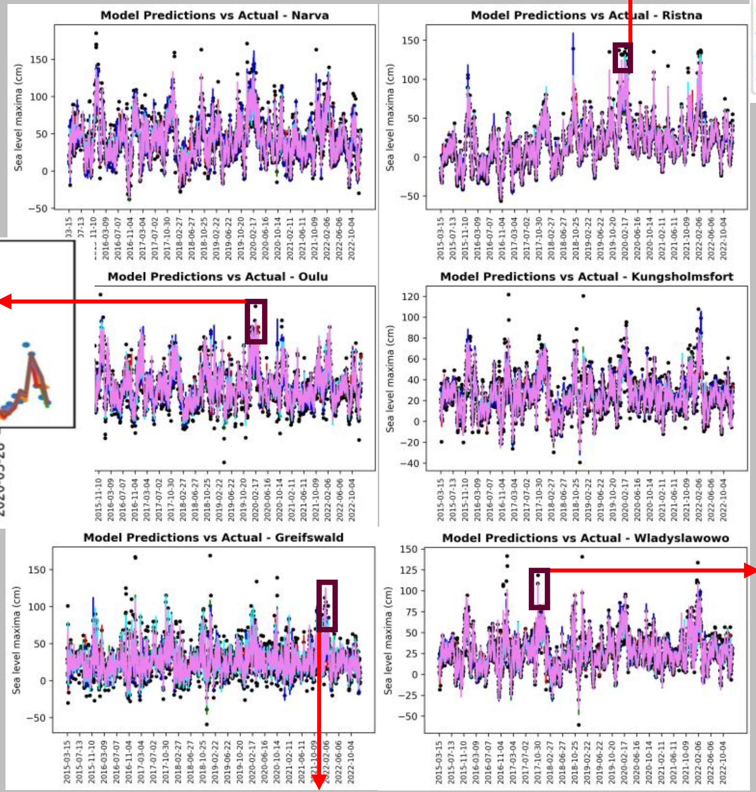
# RESULTS: INSTANTANEOUS

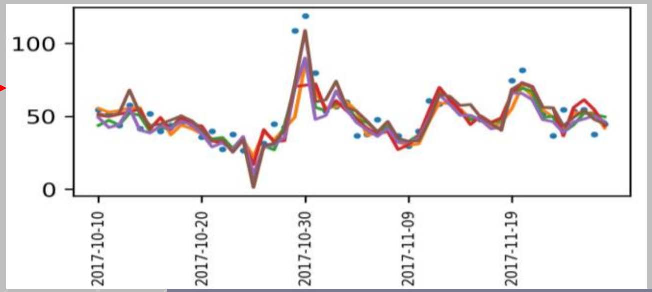


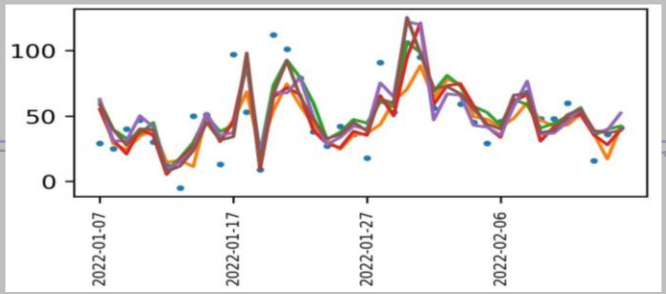


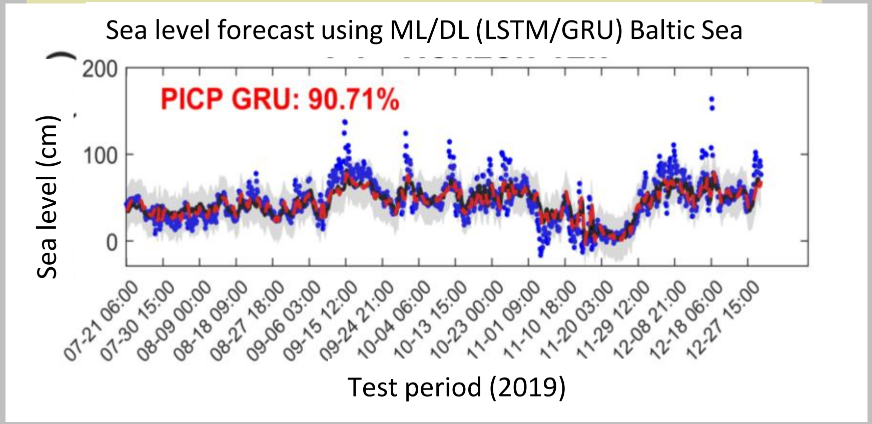







**_Previous studies: normal sea level_**

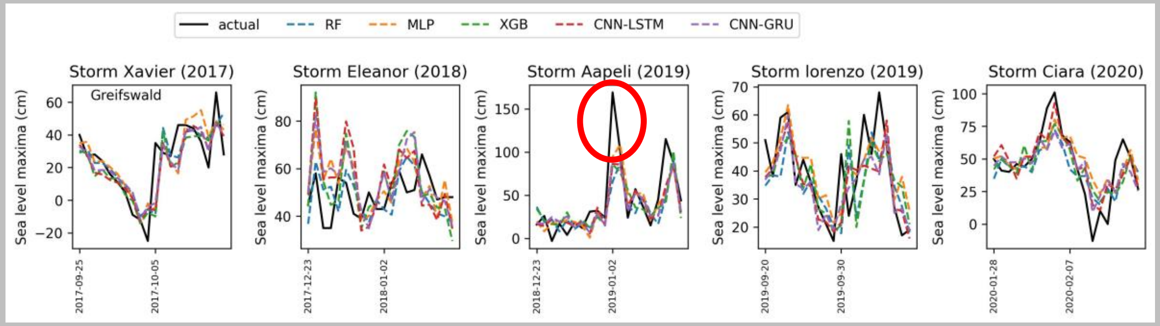


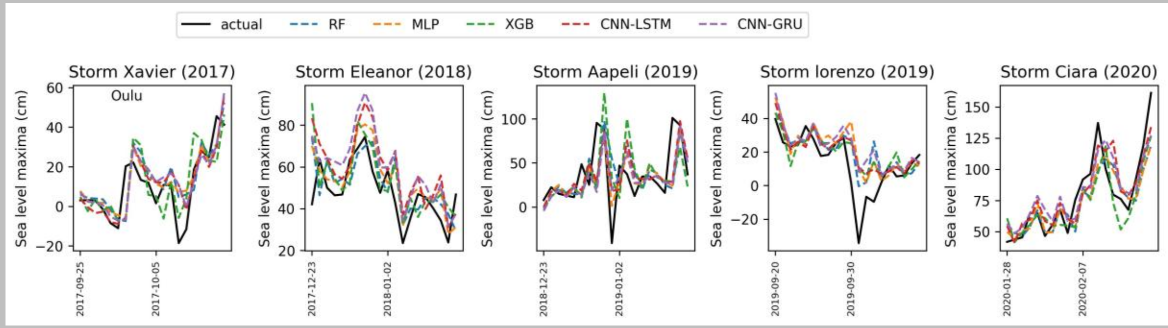

- ***MLP, CNN-GRU, and CNN-LSTM*** models performed reasonably well
- However, models ***underestimated the SLM peak values, specifically those exceeding 160 cm***

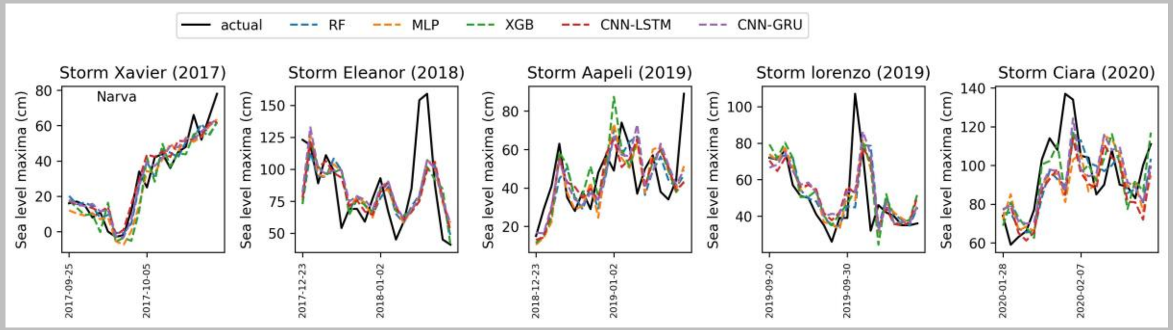
# RESULTS: STORM EVENT DETECTION

- ***Storm events*** are recognized as a ***major contributor*** to SLM
- ***Five major recent storm surge events*** in the Baltic Sea 2017–2020 are examined for their forecasting performance using ML/DL
- ***Different storms*** peaked at different stations

- ➢ Xavier (October 4–6, 2017, 118.6 cm at Wladyslawowo)
- ➢ Eleanor (January 2–4, 2018, 159 cm at Narva)
- ➢ Aapeli (January 1–2, 2019, 169 cm at Greifswald)
- ➢ Lorenzo (October 2–7, 2019, 107 cm at Narva)
- ➢ and Ciara (February 3–16, 2020, 161.30 cm at Oulu)



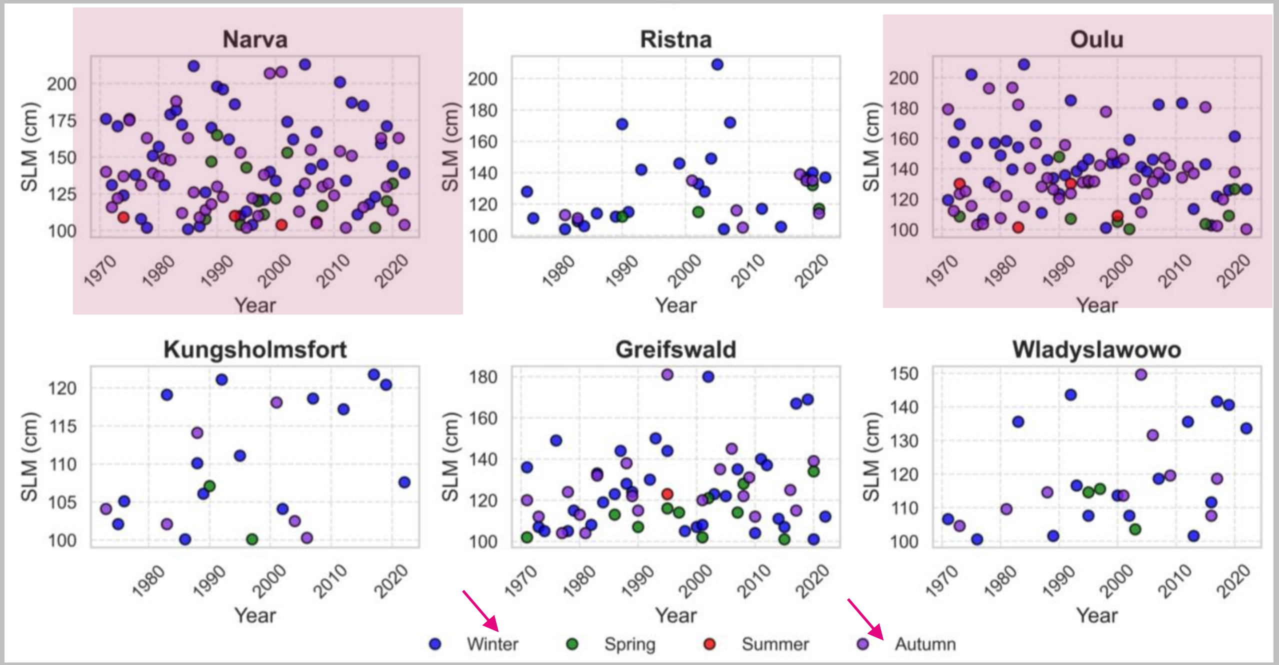



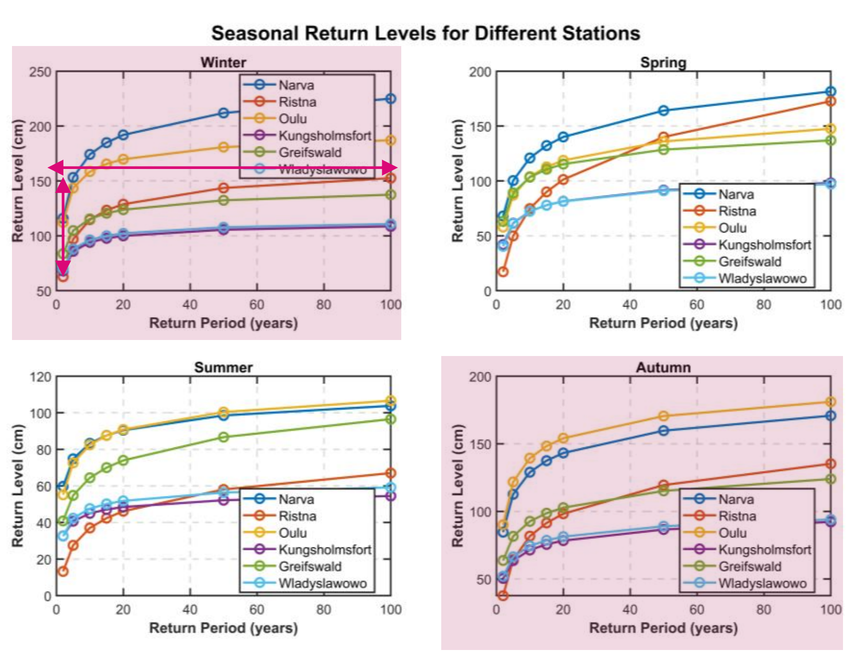

- During the analyzed storms, ***CNN-GRU model often showed better performance***, especially when dealing with sharp changes.
- Models mainly ***underestimated the peaks in storm Aapeli, especially for Greifswald with peaks at 155 cm***

# LONG TERM FORECAST: RETURN PERIODS OF EXTREMES-GEV FIT

- To ***understand intensity and frequency of SLM*** for long term forecast
- ***Deeper insight into the SLM*** not adequately represented by ML/DL

***GEV distribution with block maxima*** *(Arns et al., 2013)*

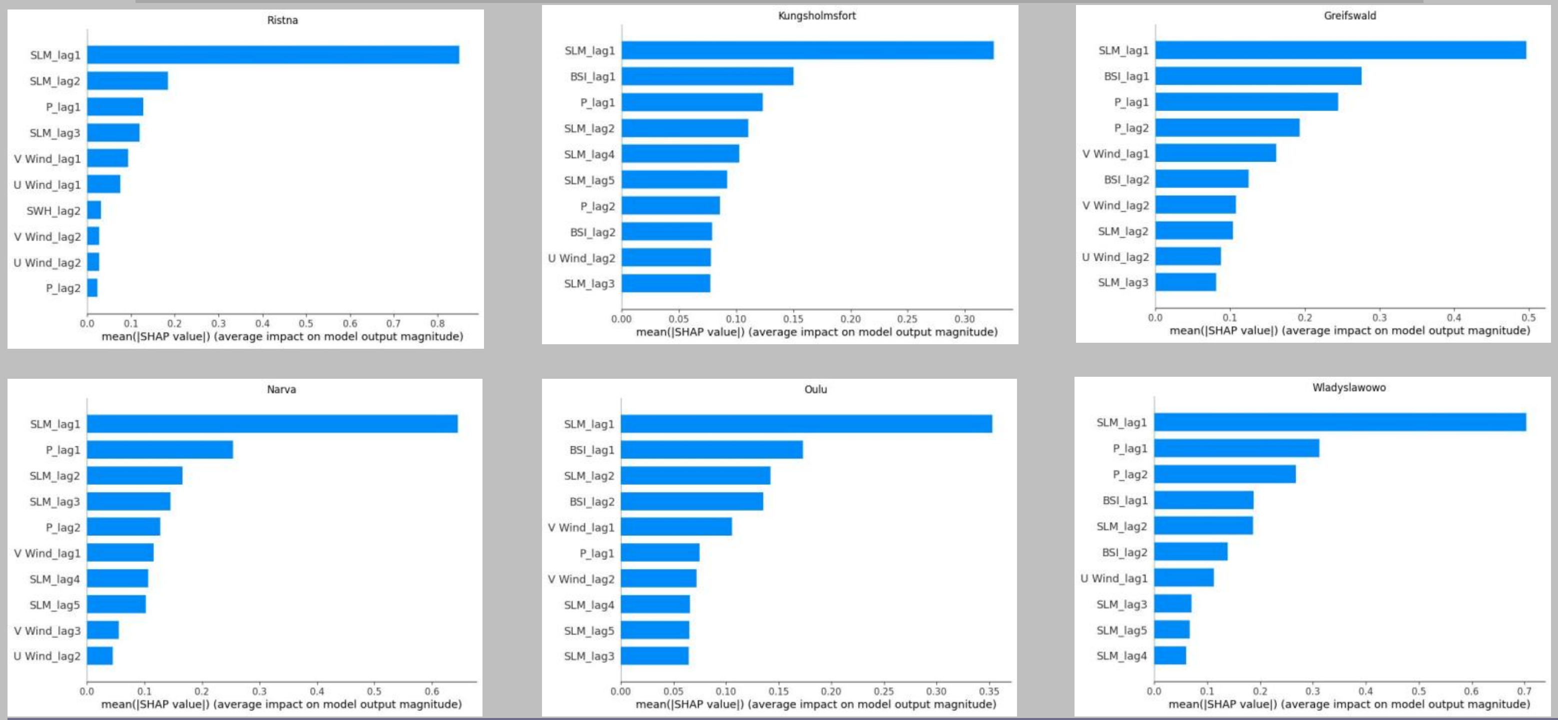



- ***Winter season*** tends to experience ***greater SLM***
- Seasonal return periods shows that ***sea level maxima of 150 cm (underestimated by models) has a 5-year winter return period in Narva and 7-year return period in Oulu stations*** (consistent with recent studies in the Baltic Sea, *(Wolski et al., 2025)*
- This study's ***37 yr trained DL/ML*** models were ***not sufficient*** in capturing these ***extremes***

# EXPLAINABILITY RESULTS: CNN-GRU MODEL

## *SHAP feature importance bar plot (Lundberg and Lee, 2017)*





- The ***SLM the day before had the greatest influence*** on models predictions
- Other features like „***pressure***", "***BSI***", *and* ***wind components*** also appear frequently in the rankings, although their significance varies depending on the location
- The ***Baltic Sea Index (BSI)*** lags are significant in mostly western locations, such as ***Kungsholmsfort***, ***Greifswald*** and ***Wladyslawowo***
- Wind-related variables, such as "***U Wind***" and "***V Wind***", had greater impacts for stations like Oulu in Finland
- ***SWH*** was most influential at ***Ristna station***

***Western locations more affected by atmospheric forcing from the North Atlantic than the eastern stations, usually more localized effects are frequent. Highest SLM found on the eastern section***

# SUMMARY

- Overall for forecasting SLM:
  - *Deep Learning method **CNN-GRU** model demonstrated **superior performance** (accuracy of 7–14.9 cm)*
  - *Other ML models like XGB and RF exhibited overfitting, (high training accuracy but lower test accuracy)*

- Models ***capturing*** most of the peaks around ***100–130 cm***, although ***missing*** some exceptional peaks ***e.g. 150 cm***

- *Key differences* in our approach that have led to this improvements in forecasting SLM include:
  - ***using daily maximum values** from a long historical dataset instead of hourly data,*
  - *a more extensive feature set using a **nonlinear mutual information (MI) method**,*
  - ***utilization of Bayesian optimization** that allows fine-tuning of hyperparameters for each station*

- Models still ***unpredicted results during some storms (SLM > (130 to 150 cm).*** This could be due to trainind data set too short or non-stationarity dynamics not catpured by models

- ***Winter season*** tends to experience ***frequent SLM***. ***SLM > 150 cm tends to occur every 5 to 7 year at Narva and Oulu stations***

- ***Western locations*** more affected by conditions of ***North Atlantic***, whilst ***eastern locations*** affected by ***localized atmospheric conditions. Eastern locations experienced greater SLM***

- A ***combination of methods*** allows a deeper understanding of SLM

# IMPROVING ON HYDRODYNAMIC MODELS

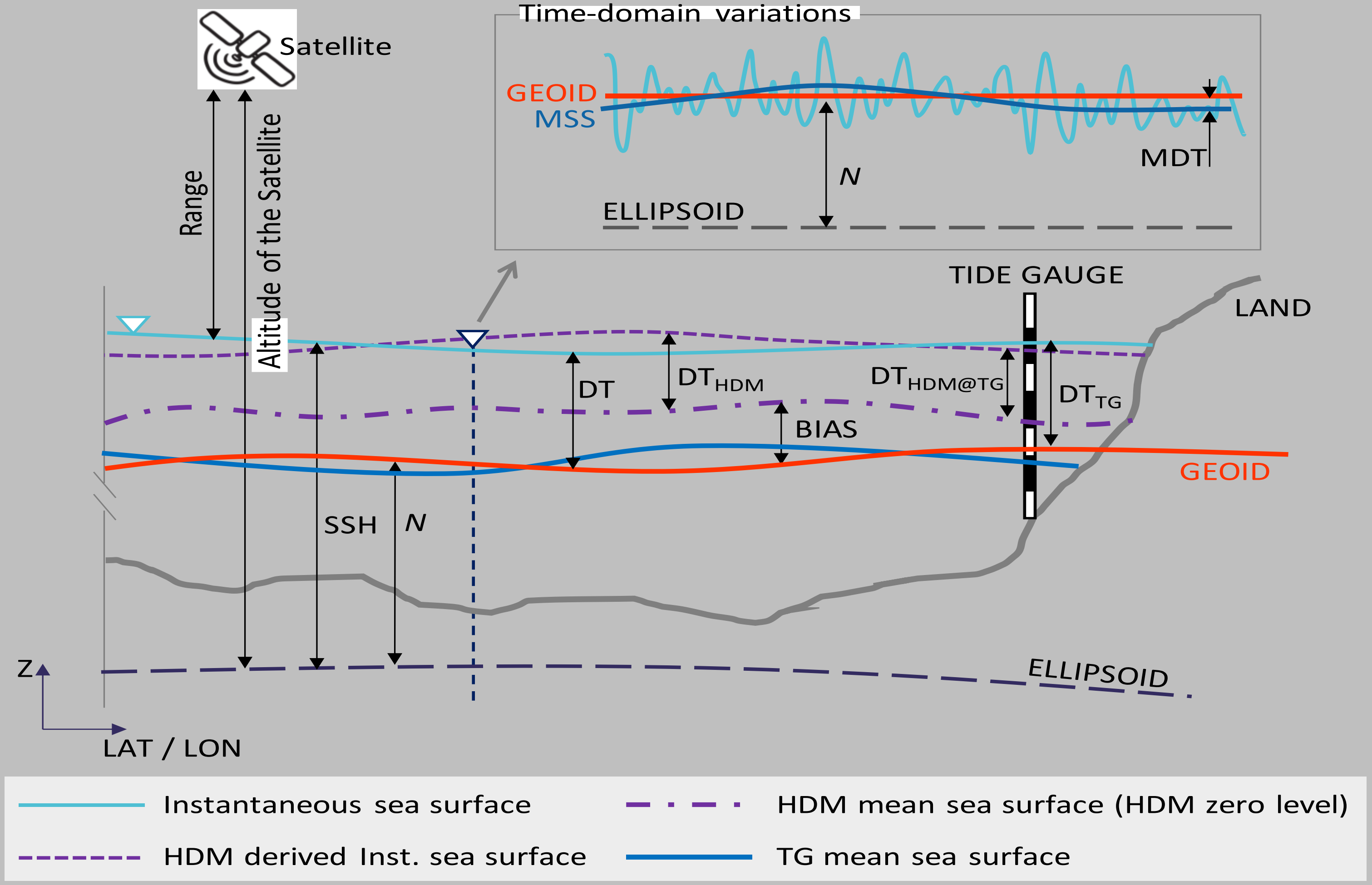
**References**


Jahanmard, Vahidreza; Löptien, Ulrike; Sandø, Anne Britt; Gierisch, Andrea M. U.; Dietze, Heiner; Lien, Vidar; Delpeche-Ellmann, Nicole; Hordoir, Robinson (2025). Barotropic Trends Through the Barents Sea Opening for the Period 1975–2021. Journal of Geophysical Research Oceans, 130, 1, 1–20. DOI: 10.1029/2024JC021663.

Jahanmard, Vahidreza; Hordoir, Robinson; Delpeche-Ellmann, Nicole; Ellmann, Artu (2023). Quantification of Hydrodynamic Model Sea Level Bias Utilizing Deep Learning and Synergistic Integration of Data Sources. Ocean Modelling, 186, #102286. DOI: 10.1016/j.ocemod.2023.102286.

Jahanmard, V.; Delpeche-Ellmann, N.; Ellmann, A. (2022). Towards realistic dynamic topography from coast to offshore by incorporating hydrodynamic and geoid models. Ocean Modelling, #102124. DOI: 10.1016/j.ocemod.2022.102124.

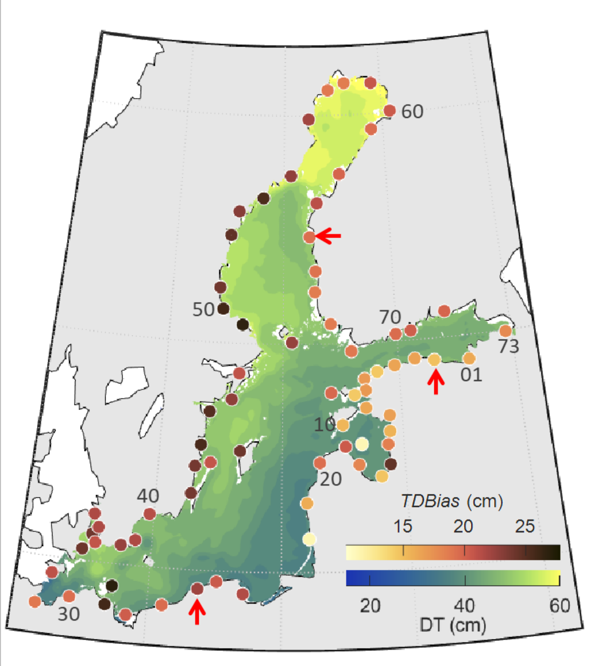
# SOURCES OF SEA LEVEL DATA: VERTICAL REFERENCE

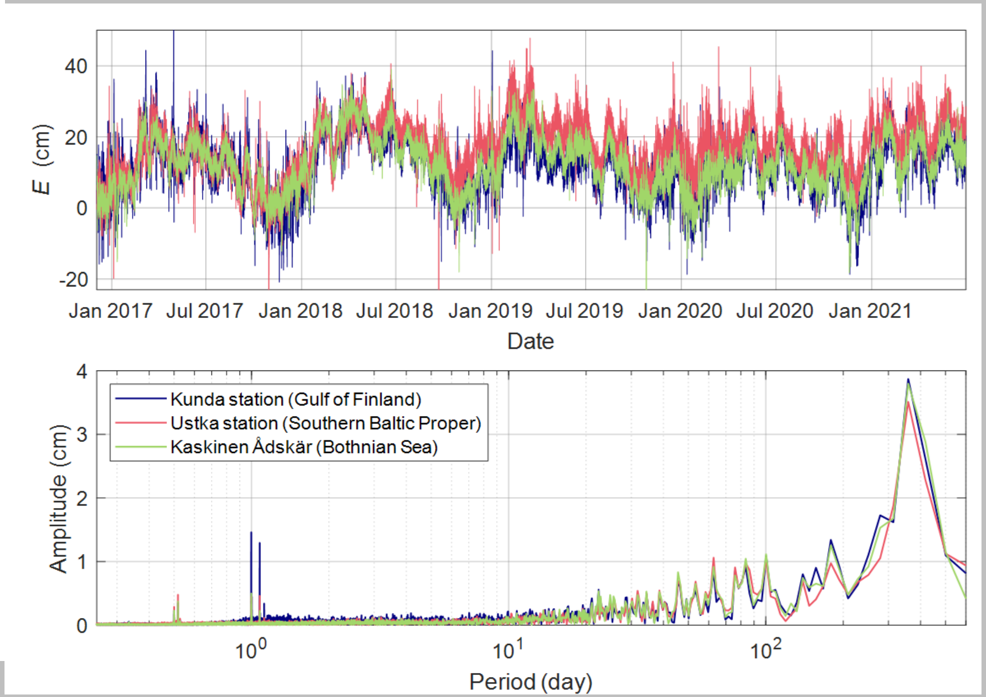



TAL TECH

# VERTICAL REFERENCE DIFFERENCES: HDM VS TG

- $E(\varphi\_TG,\lambda\_TG,t)=⟦DT⟧\_HDM\ (\varphi\_TG,\lambda\_TG,t)-⟦DT⟧\_TG\ (\varphi\_TG,\lambda\_TG,t)$




## Question/Challenge:

- Coastal areas can be corrected by TG but what is the procedure in the offshore areas?

## Observations:

- Difference can be as much as -20 to 40cm
- Stations follows similar pattern and frequency of error

## Differences:

- Spatial and temporal resolution differs
- Vertical datum differs
- Different mode of measurement

# METHOD FOR CORRECTING HDM BIAS (COASTAL TO OFFSHORE)

**Method I**: use of geoid-referenced TG network

- Use a dense close-loop network of TGs with a common geoid-based reference datum (i.e., BSCD2000).
- Propagate HDM discrepancies from stations to offshore using a bilinear interpolation at each time instant.

**TG treatment (73 TG)** by adopting a common geoid-based ref. datum → **Determine HDM time-domain bias** at TG stations → **Determine HDM spatial-domain bias** using bilinear interpolations → **Corrected HDM** by removing spatial bias → **Evaluation** Comparing corrected model with SA data, and comparing various interpolation techniques

**Method II**: use of deep learning (DL) model in a way that: $E(\varphi,\lambda,t) = \varepsilon(\varphi,\lambda,t) + RefBias$

where:

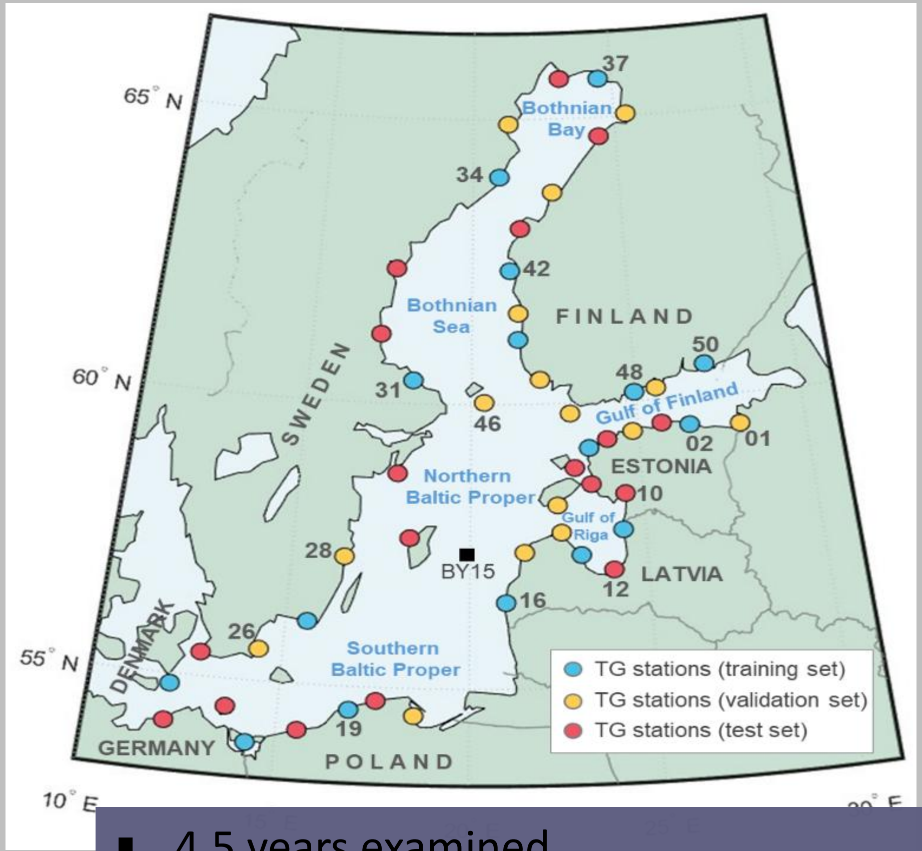
$\boldsymbol{\varepsilon}$ is HDM modelling errors (can be predicted by a DL model)

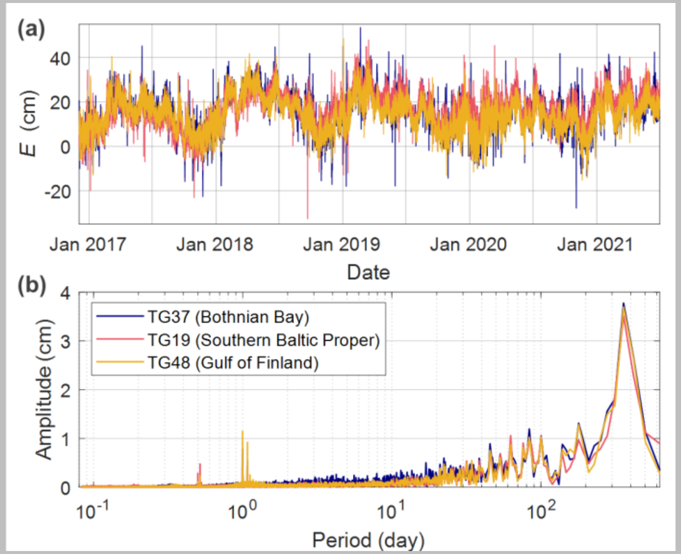
$\boldsymbol{RefBias}$ is the differences between HDM's reference surface and a particular geoid model.

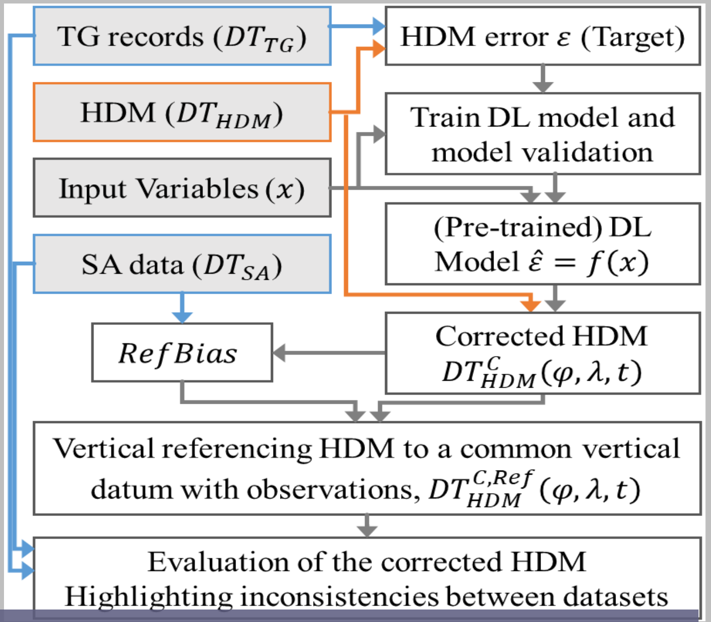
**Train a DL model on modelling errors ($\varepsilon$)** using TG records and spatiotemporal variables → **Remove HDM errors** by predicted modelling error ($\hat{\varepsilon}$) → **Vertical Ref. Bias** by comparing corrected HDM and SA data $Refbias$ = 18.3±1.9 cm → **Corrected HDM** based on new realizations of the EVRS → **Evaluation** Comparative passement between data sources

# RESULTS: METHOD 2, DEEP LEARNING (WAVENET APPROACH)

$$\mathbf{E}(\boldsymbol{\varphi}, \boldsymbol{\lambda}, \boldsymbol{t}) = \boldsymbol{\varepsilon}(\boldsymbol{\varphi}, \boldsymbol{\lambda}, \boldsymbol{t}) + \mathbf{RefBias}$$



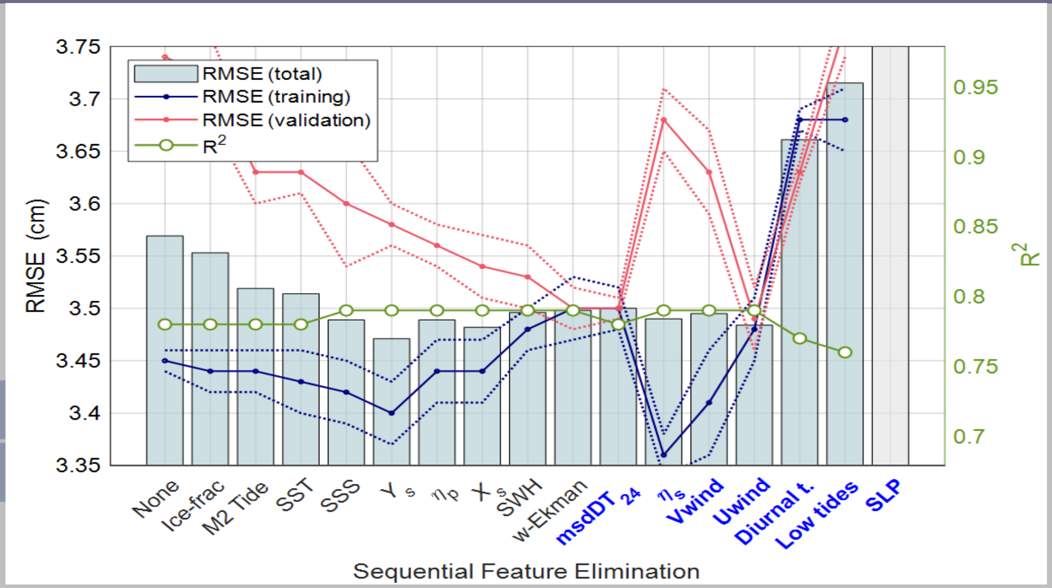

- 4.5 years examined
- Train: 16 TG stations (blue)
- Test: 18 TG stations (red)
- Validation : 16 stations (yellow)
- Evaluated: 52 stations

- The HDM error $\varepsilon$ expected to consist of different components that are most likely to be predictable both in time and space.
- RefBias is expected to be constant both in space and time
- DL model with temporal dilated causal convolution layers inspired by **WaveNet (Oord et al., 2016**)…(spectrum analysis)
- Causal convolution is unidirectional (1D), and the learnable parameters (i.e., weights and biases) are trained to predict

# METHOD II: DETERMINE RELEVANT INPUTS/VARIABLES

$$E(\varphi,\lambda,t) = \varepsilon(\varphi,\lambda,t) + RefBias$$

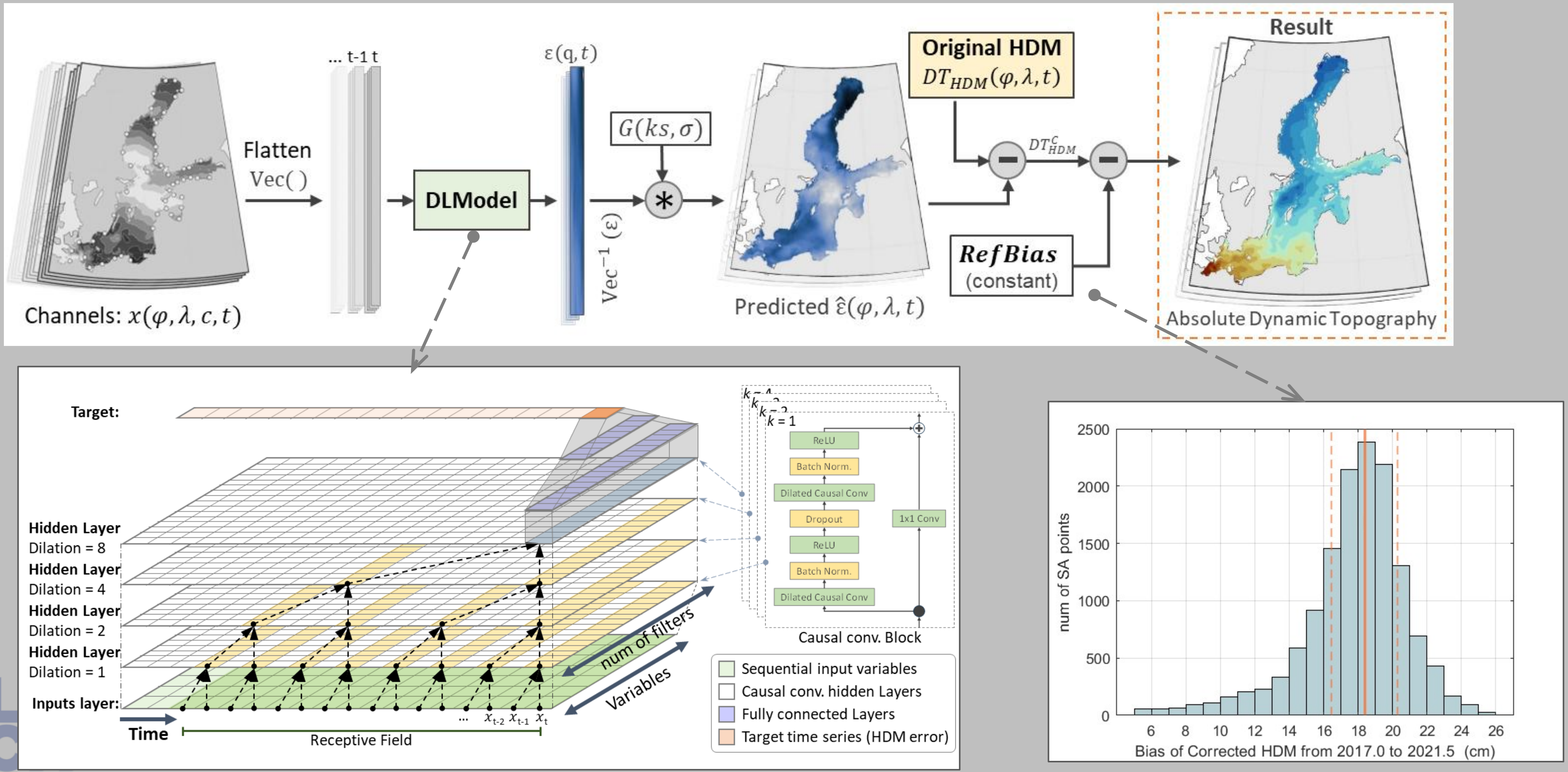
- A wrapper-type sequential feature elimination algorithm was utilized
- The algorithm states training with a subset of variables and then removes a variable based on an elimination criterion. This criterion is a combination of the RMSEs from both the training and validation sets,
- **DL model was generalized** over the spatial dimension using input variables: **'$msdDT_{24}$', '$\eta_s$', 'Uwind', 'Vwind', 'Diurnal tides', 'Low tides', and 'SLP'.**

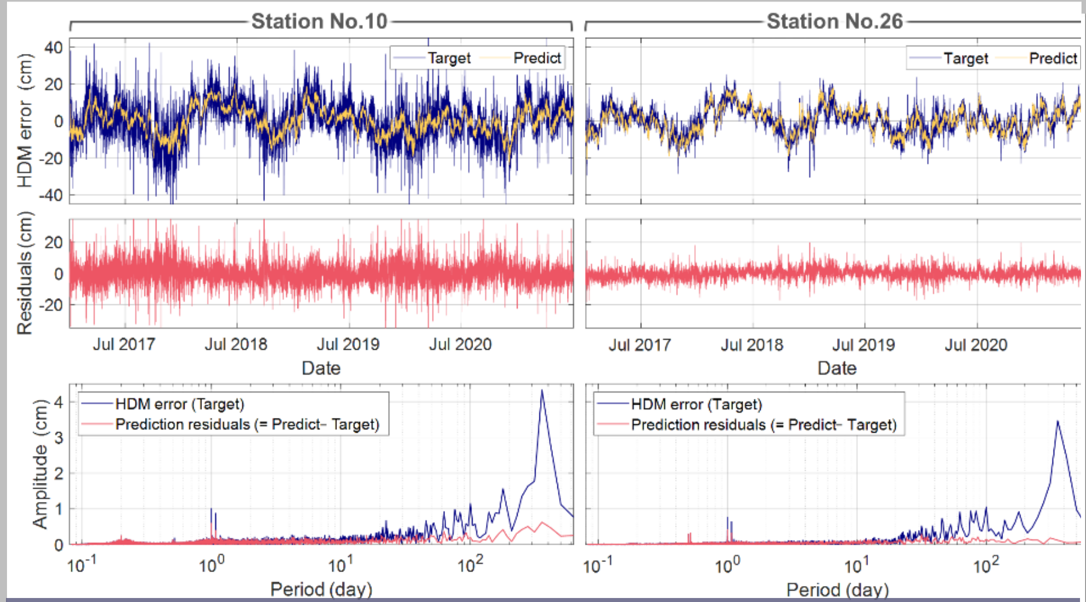


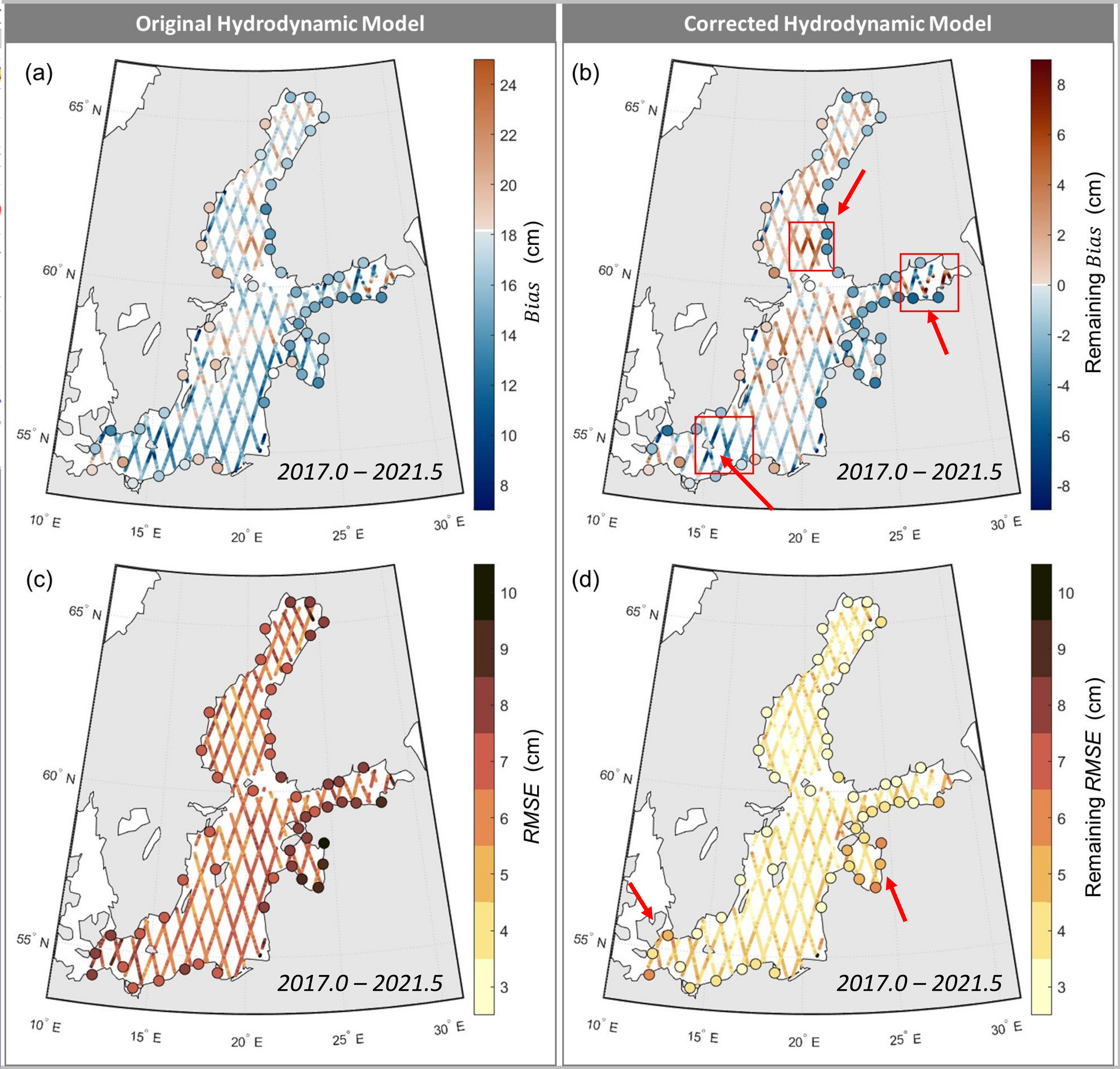

| | Variable | units | Sourced resolution | | Data source |
|---|---|---|---|---|---|
| | | | Temporal | Spatial | |
| 1 | Zonal wind (Uwind) | m/s | Hourly | 1 NM | Sourced from Nemo-Nordic dataset |
| 2 | Meridional wind (Vwind) | m/s | Hourly | 1 NM | |
| 3 | Sea surface temperature (SST) | °C | Hourly | 1 NM | |
| 4 | Sea surface salinity (SSS) | psu | Hourly | 1 NM | |
| 5 | Ice fraction (Ice-frac) | % | Hourly | 1 NM | |
| 6 | Zonal wind stress ($X_s$) | Pa | Computed at the HDM grid points with an hourly temporal resolution using U and Vwind | | |
| 7 | Meridional wind stress ($Y_s$) | Pa | | | |
| 8 | Ekman pumping (w-Ekman) | m/s | | | |
| 9 | Sea surface pressure (SLP) | Pa | 3-hourly | 5.5 km | Copernicus: https://doi.org/10.24381/cds.622a565a |
| 10 | Precipitation water col. ($\eta_p$) | cm | Hourly | 0.25° | MTPR was sourced from Copernicus: https://doi.org/10.24381/cds.adbb2d47 |
| 11 | Significant wave height (SWH) | m | Hourly | 2 km | Copernicus; https://doi.org/10.48670/moi-00014 |
| 12 | Semi-diurnal tide (M2) | cm | Computed at the HDM grid points with an hourly temporal resolution | | Aviso: https://www.aviso.altimetry.fr/ |
| 13 | Diurnal tides | cm | | | |
| 14 | Low tides | cm | | | |
| 15 | Steric height changes ($\eta_s$) | cm | | | Monthly profiles of S and T were sourced from SHARKweb: https://sharkweb.smhi.se/ |
| 16 | Sea level variability ($msdDT_{24}$) | cm | | | Computed |

# METHOD

$$E(\varphi,\lambda,t) = \varepsilon(\varphi,\lambda,t) + RefBias$$

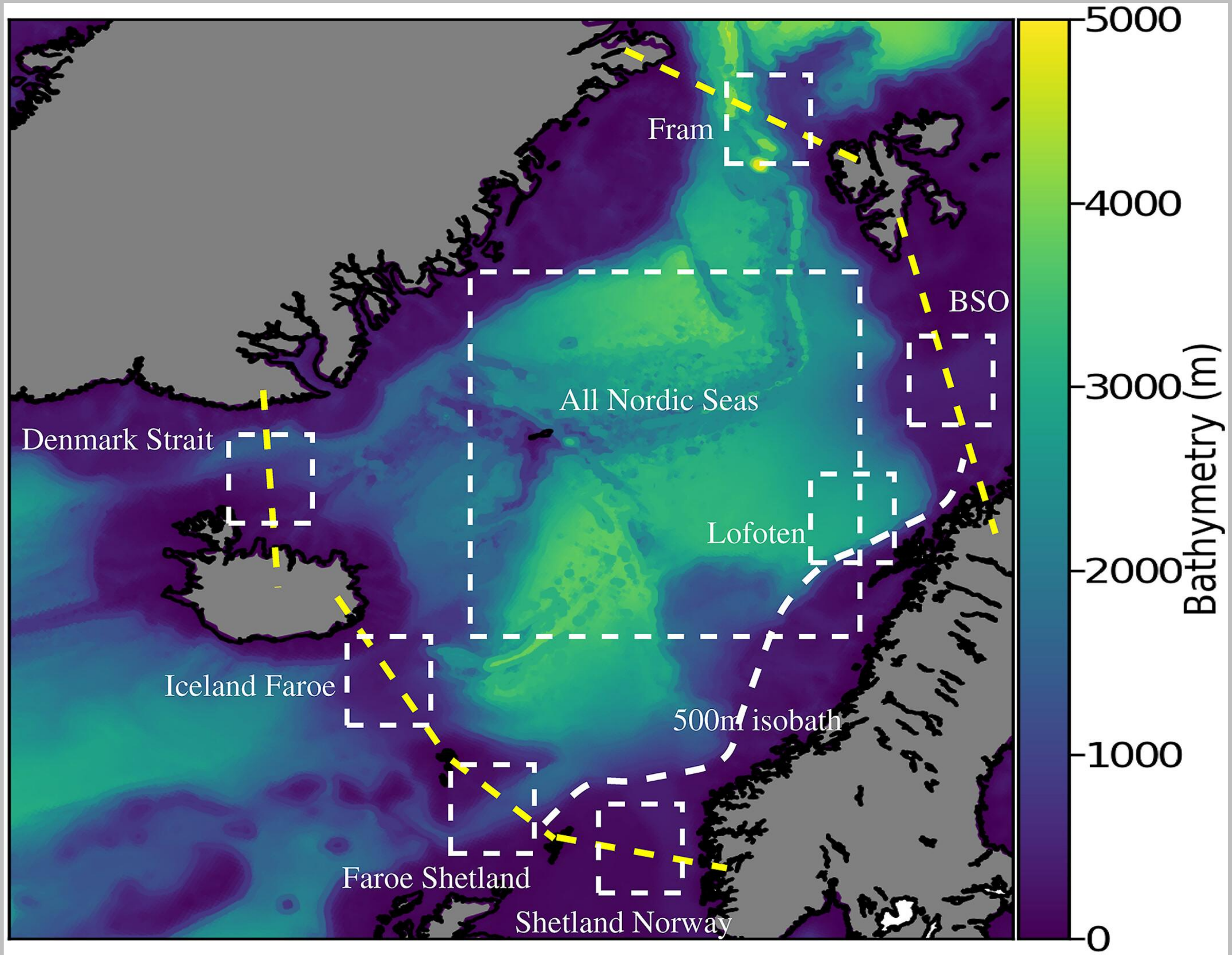





- RMSE of the Nemo-Nordic model relative to TGs improved from **7.6 cm** to **3.4 cm**.
- RMSE relative to satellite altimetry decreased from **6.5 cm** to **4.1 cm**.
- Some **problematic areas after correction** a (remaining bias exceeding ±7 cm): **eastern Gulf of Finland, Bothnian Sea,** and **the Southwest of the Baltic Sea (Bornholm)**
- Eastern Gulf of Finland, Bothnian Sea (geoid problem); Bornholm Is (uncertain)
- High RMSE areas Gulf of Riga and the entrance of the Baltic Sea where seiches may be present and that the DL model was not able to replicate

# APPLICATION TO BARENTS SEA




- Ocean model shows that the simulated volume transport at the BSO increases for the period 1975–2021. Thus bringing warmer waters into the Atlantic
- We attempted to reconstruct the temporal evolution of the BSO flow based on local time series of surface winds using a multivariate deep neural network.
- By combining expert knowledge with trial and error, we find that in order to reconstruct the flow (a) all wind data backlogged as far as 21 days and, occasionally, even as far back as 30 days is required and (b) daily resolution is insufficient, as it fails to capture the full amplitude of the trend in BSO flow

# SUMMARIZING

- **Hydrodynamic Improvements**
  - A temporal-spatial bias exist in HDM that consists of a reference bias and modelling errors
  - Machine learning  using WaveNet approach can: (i) increase accuracy of Nemo Nordic; (ii)identify and quantify errors (reference bias and modelling errors)
  - DL model identified seven main input variables: sea level pressure, diurnal and low tides, zonal and meridional wind, steric height, and sea level variability for predicting the modelling errors
  - Machine Learning depends on input variables considered so often generalized approach utilized. So location dependent variables should also be considered e.g. ice conditions
  - DL model is successful in estimating the low-frequency HDM errors, including annual and seasonal cycles. Further efforts are required for high-frequency errors.
  - Corrected HDM improved by a factor of 2, RMSE of the Nemo-Nordic model relative to TGs improved from 7.6 cm to 3.4 cm and
  - Satellite altimetry crucial for validation especially in offshore areas
  - **We applied similar approach to the Barent Sea Opening**

# THANK YOU FOR YOUR ATTENTION!

QUESTIONS