Environmental Modelling and Software 152 (2022) 105399

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft

An integrated framework for improving sea level variation prediction based on the integration Wavelet-Artificial Intelligence approaches

Ahmed Alshouny [a,b,*], Mohamed T. Elnabwy [c], Mosbeh R. Kaloop [d], Ahmad Baik [a], Yehia Miky [a,e]

[a] Department of Geomatics, Faculty of Architecture and Planning, King Abdelaziz University, Jeddah, 21442, Saudi Arabia
[b] Survey Research Institute, National Water Research Center, Giza, 12511, Egypt
[c] Coastal Research Institute, National Water Research Center, Alexandria, 21514, Egypt
[d] Department of Public Works Engineering, Faculty of Engineering, Mansoura University, Mansoura, 35516, Egypt
[e] Civil Engineering Department, Faculty of Engineering, Aswan University, Aswan, Egypt

ARTICLE INFO

Keywords:
Sea level variation
Artificial intelligence
Neighborhood component analysis
Deep learning neural network

ABSTRACT

Modeling of Sea Level Variation (SLV) is a complicated phenomenon owing to multiple factors that happen at different spatial and temporal scales. Thus, this paper presents an innovative multistep interdependent framework based on Wavelet Transformation (WT) and Artificial Intelligence (AI) algorithms for SLV prediction. A wavelet time-frequency approach along with harmonic analysis is performed firstly to understand deeply SLV behavior. Then, Neighborhood Component Analysis (NCA) is applied for the Feature selection (FS) purposes. Finally, a Deep Learning Neural Network (DLNN) algorithm is utilized to predict precisely SLV based on a newly compacted dataset. The findings revealed the potential of the DLNN model over the Machine Learning models as it improves the SLV prediction accuracy by 23%. Additionally, the proposed DLNN model can predict SLV for a time horizon of three days with a correlation coefficient = 0.91 that can help in predicting SLV early for disaster management purposes.

1. Introduction

Climate change is one of the main contributors to Sea Level Rise (SLR). This rise is expected to continue in the coming decades at rates higher than those projected in previous decades (Church et al., 2013). Understanding and analyzing temporary trends in sea level is of great importance in constraining future sea-level projections and in preparing the adaptation of densely populated coastal areas of the world. Also, SLR is important for designing coastal water supply infrastructures such as water desalination plants (Panagopoulos, 2021; Panagopoulos and Haralambous, 2020). In Egypt, numerous studies carried out to assess SLR risks due to climate change, most of them concentrated on the Nile Delta only, as it is featured as a low-lying coastal plain which makes it exposed to high risks of shoreline recession and flooding Risks (Sharaan and Udo, 2020). However, a recent study investigated that urban growth over the past decades on the northwest of the Egyptian coast had a greater impact with climate change on SLR, which threatens these areas with great danger (Hzami et al., 2021). Due to the fast-growing infrastructure of coastal tourism resorts on the northwest coast of Egypt wherein numerous water desalination plants are constructed, precise information is needed about SLR to mitigate its associated coastal hazards such as flooding and erosion. Therefore, the conservation of coastal tourism areas is a significant problem that needs to be incorporated into the Egyptian Shoreline Protection Authority (SPA) plans in order to protect beaches and prevent cost-cutting economic losses.

In General, Sea level Variation (SLV) can be classified based on the processes influencing it into two categories (Zubier and Eyouni, 2020). The impact of isostatic (earth-related) and global ice formation/melting (climates), with both a temporary scale from decadal to geological age, is the source of long-term sea-level fluctuations (Lange et al., 2014). Relatively short-term variations in the sea level are primarily caused by short-term astronomical (tidal) and atmospheric/climatic (non-tidal) astronomical impacts, which vary from hourly up to decadal. Tidal Sea level Variation (TSLV) owing to the gravitational forces of the Moon and Sun, upon the seas are periodic that makes it possible to measure and forecast easily with harmonic analysis techniques. Till now, harmonic analysis is performed by the application of Fourier analysis to the tidal motions (Abubakar et al., 2019). As there are hundreds of periodic motions of the Earth, Sun, and Moon, each of these motions or "constituents" are harmonic constants represented by a mathematical value describing the effect of cyclical motion on the tides. However, recent

* Corresponding author. , Department of Geomatics, Faculty of Architecture and Planning, King Abdelaziz University, Jeddah, 21442, Saudi Arabia.
E-mail address: ametaowa@kau.edu.sa (A. Alshouny).

https://doi.org/10.1016/j.envsoft.2022.105399
Received 26 April 2021; Received in revised form 14 November 2021; Accepted 5 April 2022
Available online 13 April 2022
1364-8152/

Abbreviations

SLV	Sea Level Variation
WT	Wavelet Transformation
AI	Artificial Intelligence
NCA	Neighborhood Component Analysis
FS	Feature selection
DLLN	Deep Learning Neural Network
ML	Machine Learning
SLR	Sea Level Rise
TSLV	Tidal Sea Level Variation
NTSLV	Non-Tidal Sea Level Variation
CC	Correlation Coefficient
ANN	Artificial Neural Network
SVR	Support Vector Regression
RNN	Recurrent Neural Networks
ANFIS	Adaptive Neuro-Fuzzy Inference System
ARMA	Auto-Regressive Moving Average
WANFIS	Wavelet-ANFIS
WANN	Wavelet ANN
LSTM	Long Short-Term Memory
WD	wind direction
WS	Wind Speed
R	Total Rainfall
RH	Relative Humidity
T	Air Temperature
E	Evaporation Rate
C	Cloudiness
P	Atmospheric Pressure
SST	Sea Surface Temperature
MSL	Mean Sea Level
SR	Solar Radiation
WR	Wind Run
DP	Dew Point Temperature
WBT	Wet Bulb Temperature
DWT	Discrete WT
CWT	Continuous Wavelet Transform
GBTR	Gradient Boosted Trees Regression
DT	Decision Tree
GPR	Gaussian Processing Regression
RMSE	Root Mean Squared Error
K-NNR	K-Nearest Neighbor Regressor
NCA	Neighborhood Component Analysis
VIRF	Variable Importance of Random Forest
PSD	Power Spectrum Density
EDA	Exploratory Data Analysis

studies have shown that the principal harmonic components of tidal movement However, recent studies have investigated the principal harmonic components of tidal movement and summarized them as shown in Table 1 (Guo et al., 2018; Imani et al., 2018).

In contrast to TSLV, Non-TSLV (NTSLV) are not frequent and thus require sophisticated techniques to analyze its behavior. The impacts of tides dominate the variability of water levels in some coastal regions (Li and Han, 2015). But when the tidal effects are smaller such as those of the Mediterranean Sea (Enríquez et al., 2017; Sammari et al., 2006), other mechanisms including atmospheric variables play a major role in SLV.

Many researchers have investigated thoroughly the effect of atmospheric forces on SLV worldwide (Afshar-Kaveh et al., 2020; Jan et al., 2006; Meena and Agrawal, 2015; Pasarić et al., 2000; Raicich, 2010). Traditional approaches such as empirical formulations, frequency analysis, and numerical modeling, are used earlier for SLV predicting. Tools of the first approach seek empirical formulas to combine different effects into a single parameter (or more) for SLV forecasting. Applying statistical analysis such as linear regression for observed SLV data and associated atmospheric parameters gives satisfactory results (with Correlation Coefficient (CC) = 0.82) (Tilburg and Garvine, 2004). However, they are not detailed enough to be used in actual cases because of the simplifications produced in these formulas. In the second approach, researchers studied the relation between NTSLV and some meteorological forces such as wind and atmospheric pressure in both time and frequency domains (Afshar-Kaveh et al., 2020). They found that the effect of wind stress was higher than atmospheric pressure on NTSLV.

Table 1
Principal harmonic constituents.

Symbol	Constituent Description	Period (T)
M2	Principal semidiurnal lunar	12.42 Hours
S2	Principal semidiurnal solar	12.00 Hours
O1	Principal diurnal lunar	25.82 Hours
P1	Principal diurnal solar	24.06 Hours
N2	Elliptical of M2	12.66 Hours
K2	Declinational of M2	11.96 Hours
Q1	Elliptical of O1	26.86 Hours
K1	Declinational of P1	23.93 Hours

On the other hand, the third approach predicts numerically NTSLV based on atmospheric driving forces (Kurniawan et al., 2015; Shen et al., 2006). In these studies, the researchers employed numerous numerical models to investigate the effect of different atmospheric forces on NTSLV. They concluded that the modeled and measured NTSLV have some differences with relative errors (10%–50%). As these numerical models are based on water body motion equations, they may however be non-precise for several reasons such as spatial and temporal approximations, differential equations convergence approximations, and initial boundary uncertainties. Moreover, it's both time consuming and expensive at the same time. As a consequence, researchers have focused on developing new models to address the limitations of traditional approaches. An alternate way of modeling and forecasting SLV is Artificial Intelligence (AI) and Wavelet Transformation (WT) approaches. Most tidal researchers used the Artificial Neural Network (ANN) as an alternative to the traditional harmonic analysis technique for short-term SLV (Makarynska and Makarynskyy, 2008; Meena and Agrawal, 2015; Salim et al., 2015). Moreover, the performance of ANN is compared to other techniques such as Genetic Programming (GP) wherein both models perform well and could be regarded as alternatives to the harmonic analysis (Ali Ghorbani et al., 2010). Other researchers employed Support Vector Regression (SVR) tools for storm surges predicting (Rajasekaran et al., 2008). Also, few studies have been found in literature for SLV prediction with Deep Learning (DL) approaches. Some researchers used Recurrent Neural Networks (RNN) to integrate the effects of different SLV phenomena, such as sun and moon gravity attractions and climate change to study its effects on SLV (Ishida et al., 2020). Furthermore, many hybrid models are utilized for modelling SLV and for many other applications (El-Diasty et al., 2018; Huang and Wu, 2017; Ou et al., 2016, 2017; Ou and Hong, 2014; Turki et al., 2015). Globally, an extensive literature indicated that ANN are the most applied AI based model for SLV prediction. Table 2 summarizes the substantial improvement in NTSLV prediction and modelling via AI approaches.

As can be seen in Table 2 and considering the outcomes of AI approaches used in NTSLV forecasting from these preliminaries, the research gaps from this summary in the research area of SLR can be summarized as:

Table 2
Representative studies related to SLR Prediction.

Type of AI Approaches	Ref	Input Metrological Variables	Model Performance (CC)	Time Scale
ANN, ANFIS	Muslim et al. (2020b)	WD, WS, C, MSL	0.82 @ 1m	Monthly
ANN	Majdzadeh Moghadam (2017)	WS, R, RH, T, SST, P, MSL	0.56@ 1 m	Monthly
ANN	Zubier and Eyouni (2020)	WD, WS, P, T, E, NTSLV	0.95@1 d	Daily
ANN, ARMA, ANFIS	Karimi et al. (2013)	MSL(t)	0.8 @72 h.	Hourly
ANN	Tsai and You (2014)	WD, WS, P, MSL(t)	0.91@45 hr	Hourly
ANN	Chao et al. (2020)	WS, P, MSL(t)	0.93@12 hr	Hourly
ANN, ANFIS, WANN, WANFIS	Wang et al. (2020)	WS, P, MSL(t)	0.97@1 hr	Hourly
ANN	Filippo et al. (2012)	WD, WS, P	0.98@1 hr	Hourly

Where: ARMA: Auto-Regressive Moving Average; WANFIS: Wavelet-ANFIS; WANN: Wavelet ANN; LSTM: Long Short-Term Memory; DL: Deep Learning.
WD, WS are the Wind Direction, Speed; R is the Total Rainfall; RH is the Relative Humidity, T is the Air Temperature; E is the Evaporation Rate; C is the Cloudiness; P is the Atmospheric Pressure; SST is the Sea Surface Temperature; MSL is the Mean Sea Level.

1. Most of the researchers focused on investigating the effects of (WD, WS, R, RH, T, SST, C, P) as meteorological predictors for NTSLV forecasting without taking into consideration the effects of other important factors such as Solar Radiation (SR), Wind Run (WR), Dew Point Temperature (DP), and Wet Bulb Temperature (WBT).
2. In-depth analysis is required to assess the effect of metrological factors on NTSLV, most of the studies investigated the relation between metrological factors and absolute MSL as a target which may be influenced by both tidal and non-tidal effects without extracting NTSLV alone which seems to be non-precise analysis.
3. Additional Feature Selection (FS) analysis should be carried out, based on their significance. Alternative techniques, such as machine learning and metaheuristic algorithms for selecting the most essential input variables aren't considered.
4. DL algorithms have recently shown outstanding performance in classification tasks, especially for those with large data. However, few studies in the field of SLR have shown that these techniques can be used to predict precisely (Ishida et al., 2020). Further investigation in this field should be produced.
5. As can be seen clearly in Table 2, some earlier trials were made with the construction of a suitable AI model that captures the nonlinear relationship of NTSLV with meteorological conditions to create effective and accurate short term (1-h-head) prediction. There will also be little efficacy in managing coastal disaster (prevention/mitigation) by readiness systems and early warning systems. Nevertheless, the precise long-term NTSLV forecast remains challenging.

2. Research innovation and goals

A thorough comprehension of the processes driving its dynamic spatial-time evolution of sea behavior is the basis for reliable predictions of future SLR. In the aforementioned existing SLV prediction studies, several deficiencies are still essential to clarify. The current study, therefore, introduces an integrated approach focused on sophisticated methodologies for precisely forecasting SLV. The major contribution of the current research can be summarized as:

1. Investigating the SLR due to meteorological parameters through an extensive analysis by implementing frequency analysis and t-tide algorithms to extract NTSLV components only.
2. Due to existing Large datasets which consists of 8758 series in the current study, traditional ML algorithms can't obtain satisfactory results, thus a novel approach called Neighborhood Component Analysis (NCA) is utilized to reduce the dataset dimensionality and get the optimal input features to enhance the prediction process. Most of NCA algorithm applications are for classification tasks (Yang et al., 2012). To the best of the authors' knowledge, no previous study utilized NCA through applying the FS strategy for regression tasks in the research area of SLR.
3. DL algorithm is developed based on the selected features to develop a promising AI predictive model with higher performance.
4. To validate the proposed model outcomes, the obtained SLV values are compared with similar previous studies to measure the prediction capability of the proposed model.

3. Material and methods

3.1. Study area

In this research, the hourly recorded time series sea level data with associated meteorological variables are collected during the year 2019 by Campbell Scientific Weather Station deployed on the northwest coast of Egypt (31.22 E, 27.84 N), near Matruh City (Fig. 1). This study area has been developed and Egypt's government has built a number of new coastal towns in the past and current periods. Based on the Fifth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC) (Church et al., 2013), the Egyptian Mediterranean coastal area is one of the most vulnerable to SLR due to climate change impacts. Thus, this location is primary chosen to develop an efficient AI predictive model which may play an essential role in early warning and coastal protection planning for this vital region.

3.2. Wavelet analysis

Unlike the Fourier Transform technique which only revealed frequencies existing in the whole time series and information on individual occurrence may be lost, Wavelet Transform (WT) expands time series into time-frequency space and thus localized intermediate periodicities can be identified (Grinsted et al., 2004).

Generally, there are two categories of WT; the Discrete Counterpart (DWT) and its Continuous Wavelet Transform (CWT). The DWT is a compact display of the data, especially suitable for noise reduction and data compression, whilst the CWT is better for the extraction of features (Muslim et al., 2020b; Vu et al., 2010). The Morlet wavelet transform is used as a common CWT function in the current research to understand the spectral features and investigate the SLV cycle terms (tidal) for stating the most significant tidal harmonic constituents using Eq. (1) (Muslim et al., 2020a; Pancheva and Mukhtarov, 2000; Vu et al., 2010). This WT formula can be used to analyze time series containing non-stationary power at various frequencies in the time and frequency domains.

$$\psi_{a,b}(t) = \pi^{-1/4}\, e^{i\omega_0\eta} e^{-\gamma^2/2} \tag{1}$$

Where ψ_0 (t) presents the wavelet function, i is the imaginary unit number, ω_0 is the non-dimensional frequency, and γ is the non-dimensional time parameter. At higher frequencies, the width of the wavelet function is narrower and at lower frequencies wider (Grinsted et al., 2004). The WT expresses arbitrary functions by superimposing the

(a) (b) (c)

Fig. 1. (a) Location of Study Area Near Matruh in the Mediterranean Sea (Fig. 1 a source is Google Earth website) **(b)** Land Meteorologic Sensors **(c)** Sea Level Radar and SST Sensors.

wavelet mother function and each function has a different scale and a corresponding resolution.

3.3. Machine learning algorithms

In the current research, a first attempt to develop a promising predictive model using traditional supervised ML algorithms. It would be infeasible to evaluate all of the ML regression algorithms used for SLV modeling published in the literature. In reality, the best machine learning algorithms relies on the problem at hand and the nature of dataset (Almaliki, 2019). As a result, the authors focused on a study of commonly used algorithms that have proven to be efficient when compared to other ones especially for large datasets issues as in the current research (Cervantes et al., 2008; Das et al., 2018; Franco-Arcega et al., 2012; Zhang et al., 2018a). Thus, four selected algorithms namely Gradient Boosted Decision Trees Regression (GBTR), Supported Vector Regression (SVR), Gaussian Processing Regression (GPR), and K-Neighbor Regressor (K-NNR). A brief description of the mechanism of each model and its important parameters will be introduced in the following subsections.

3.3.1. Gradient Boosted Trees Regression (GBTR)

Gradient Boosted Trees Regression (GBTR) is a type of Decision Tree (DT) meta learner, as the name implies, is a tree structure-like flow that operates with the condition's theory. It has efficient algorithms that are used for prediction and classification. It mainly allocated internal nodes, branches, and terminal leaves (Pekel, 2020). A regression tree is rendered almost in the same way as a classification tree, with the exception of a regression measure replacing the impurity measures suitable for classification. Also, boosting methods are intended to enhance the performance by turning weak learners into strong ones.

Without a commonly re-weighing mechanism, GBTR uses a gradient descent algorithm for poor learners' shortcomings. GBTR has multiple parameters such as the loss function, split sample, leaf sample, and the learning rate that can affect the accuracy of the estimate. GBTR algorithm aims at reducing the loss function error. Root Mean Squared Error (RMSE) is one of the most commonly used loss functions in regression problems (Pekel, 2020). So, it will be used in the current research. Assume loss function (RMSE) = f (y_i, F(x_i)) where y m is actual output and F(x_i) is the model we want to fit in. Then, the aim is to minimize RMSE (j) function.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - F(x_i))^2} \quad 2$$

By applying gradient descent algorithm,

$$\mathrm{F}(x_i) = \mathrm{F}(x_i) - \alpha \frac{\delta J}{\delta J \mathrm{F}(x_i)} \quad 3$$

Where α is learning rate that accelerates or decelerates the learning process and δJ is the gradient of the loss function. In the current research, both the number of leaf samples and the learning rate will be fine-tuning until reaching the maximum performance of the GBTR model.

3.3.2. Support Vector Regression (SVR)

SVR is a statistical learning-based neural network that successfully uses for several engineering regression problems (Patil et al., 2012). The SVR is classified as a supervised Ml algorithm that uses a hyperplane for separating data from one dimension into a high dimensional space and then solves the regression problems using the following equation:

$$y = f(x) = \sum_{i=1}^{n} w \times \varnothing_i(\mathrm{X}) + b \quad 4$$

Where w is the weight vector, b is the bias factor, n is the size of the data set, and $\varnothing_i(\mathrm{X})$ is the Kernel function which may be Linear, polynomial, radial basis, or sigmoid that maps the input variables in a large space of features in order to model the non-linear regression using linear regression. The most commonly used cost function for SVR modeling is called ε-SVR which can be expressed as:

$$|\varepsilon| = |y - f(x)| = \begin{cases} \varepsilon & if \quad y - f(x) \leq \varepsilon \\ |y - f(x) - \varepsilon| & otherwise \end{cases} \quad (5)$$

The main idea of the ε-SVR function is to maximize the ε-derivation. The minimization function can be explained as:

$$Minimize \quad \frac{1}{2}w2 + C\sum_{i=1}^{n}(\xi_i + \xi_i{}^*) \quad 6$$

subjected to yi (w × xi + b) ≥ 1-ξ_i where. $\xi_i \geq 0 \& C > 0$

Where C is a cost factor that is assumed to determine a trade-off between empiric risk and model flatness, (ξ_i) and ($\xi_i{}^*$) are slack variables that represent the distance from the real values to the corresponding boundary values. In the present research, different Kernel functions and C values will be investigated via a trial-and-error technique to get the optimal performance of the SVR algorithm.

3.3.3. Gaussian Processing Regression (GPR)

GPR is categorized as a versatile, non-parametric, and supervised ML algorithm. GPR's primary advantages are the interpretability and probabilities of prediction and outcomes when embedding previous models. In recent decades, theoretical work and implementation have shown that GPR is an efficient tool for developing ML applications(Jiang et al., 2019). Considering a training dataset $\{(x_i,y_i);\ \ i\ =1,2,\ldots.n\}$, the input data $X \in RD^{*}n$ is called the dataset matrix and $y \in Rn$ is the desired output vector. In light of the new input vector x_{new} and based on the training results, the GPR model addresses the predicted the new response variable y_{new} can be estimated as:

$$y_{new} = x_{new}^{T}\beta + f(x_i) + \varepsilon_i \tag{7}$$

Where, β is the bias a vector of basis function coefficients, $f(x_i)$ is a GP with zero mean and covariance function $K(x.x')$, and ε_i is the Gaussian noise with variance (σ_n^2). In the Gaussian process, the signal term $f(x_i)$ is often considered a random variable.

$$f(x_i) = \mathscr{GP}[m(x).:K(x.x')] \tag{8}$$

Where $m(x)$ is a mean function, and $K(x.x')$ is a covariance function that explains previous assumptions, such as probable smoothness of data and trends. $\mathscr{GP}$ term creates a non-linear relationship between the input, response variable including high data uncertainties. Additionally, for more effective estimates, numerous covariance functions with desired signal variances values will be utilized in the current study.

3.3.4. K-Nearest Neighbor Regressor (K-NNR)

The k- Nearest neighbor Regressor (K-NNR) is one of the simplest nonparametric ML algorithms used for regression and classification tasks (Zhang et al., 2018b). It's having a higher predictive capability especially for large datasets that have little to no prior knowledge. Therefore, it should be one of the first choices as it doesn't have any assumptions on underlying data and doesn't affect the outliers. The KNNR algorithm is based primarily on the high probability of the most related samples of the same class. KNN stores the data set during the training process and when new data is received, it can easily be labeled into a well-suite category that is very similar to new ones (Zhang et al., 2018b). Considering $S = \{(x_i,y_i);\ \ i\ =1,2,\ldots.n\}$ to be a training set containing n observations for a regression problem. Where $x_i\ =\ (x_{i1},x_{i2},\ldots\ldots,,x_{im})$ is the i-th instance denoted by m features with its response y_i. When new tested data (x_t,y_t), is acquired, it's required to know how close each testing point to each training point in S by calculating the distance metrics (d) which can be estimated. Then, the distance d is sorted by its value to the closest i-th instance which called the k-nearest neighbor with output y_i (x). Lastly, the final predicted output y_{av} is the average of the outcomes of its k nearest neighbors which can be estimated as follow:

$$y_{av} = \frac{1}{\mathrm{k}}\sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{y_i}(\mathrm{x}) \tag{9}$$

In KNR algorithm, two issues are still challengeable for researchers, the choice of the K-value and the suitable distance metrics parameters (Kim et al., 2016). These two parameters must therefore be investigated iteratively in order to get the best KNR's performance.

3.3.5. Feature selection Neighborhood Component Analysis (NCA)

Feature Selection: Recently, Feature selection became a key step for high dimensionality ML issues. The main objective of the feature selection is to pick the best possible combination of features that will improve the regression model performance. Feature selection increases the prediction precision by reducing the data dimension (Zhang et al., 2018a). Feature selection algorithms can be classified into three main categories filters, wrappers, and embedded techniques. This classification is based on the usage of a learning algorithm in the approach (Gregorutti et al., 2017). In the current research, the filter type Neighborhood Component Analysis (NCA) and the embedded type Variable Importance of Random Forest (VIRF) will be investigated for these purposes.

NCA is a supervised ML algorithm which functionally similar to a K-NNR algorithm that explicitly uses a related term called the closest neighboring stochastic. It selects significant features in order to optimize the predictive precision of regression models. Considering the same training set of K-NNR, a randomized regression model picks a point from the input variables in S , say a point x_j with its corresponding output value y_j. The probability P $(x_j|S)$ that x_j is selected from S as the reference point for x is higher if x_j is closer to x as measured by the distance function (d_w) which can be estimated by:

$$d_w(x_i,x_j) = \sum_{r=1}^{p} w_r^2 |x_{ir} - x_{ij}| \tag{10}$$

Where, w_r are the feature weights assuming that P $(x_j|S)$ α K $(d_w(x_i,x_j)$, K is a kernel function that assumes large values when $d_w(x_i,x_j)$ is small. P $(x_j|S)$ for all j must be equal to 1 and therefore the probability to select a feature x_j from S can be formulated as:

$$P(x_j|S) = \frac{kd_w(x_i,x_j)}{\sum_{j=1}^{n} kd_w(x_i,x_j)} \tag{11}$$

Herein, let y_b is the predicted output value by the regression model and y_a is the true output of x_i and l be the loss function that measures the difference between y_a and y_b, the average value of $l\ (y_a,\ y_b)$ becomes $l_i = (l\ (y_a,\ y_b))\ |S^{-i}) = \sum_{j=1,j\neq i}^{p} p_{ij}(y_i,y_j)$. In the current study, the mean squared error is utilized as the loss (cost) function (MSE) $= (y_i - y_j)^2$. Also, for preventing overfitting of the regression model, A regularization term λ is added to the final objective function(w) which can be finally formulated as:

$$(\mathrm{w}) \ = \ \frac{1}{\mathrm{n}}\sum_{j=1}^{n} l_i + \lambda\sum_{j=1}^{p} w_r^2 \tag{12}$$

The criteria for selecting significant features based on the definition of a relative (T) threshold can be formulated as

$$T = \tau * \max(w) \tag{13}$$

Where, τ is tolerance fixed to 0.02 as recommended by and w is updated weight of features (Yang et al., 2012).

On the other hand, Variable Importance of Random Forest (VIRF) is one of the most important FS techniques used for quantifying the variable importance based on the permutation techniques (Breiman, 2001). The principle behind measuring the importance of variable x_j is to permute all of its values, and the variable importance measure is defined as the difference in prediction accuracy generated by the permutation. Breiman (2001) proposes that the good performance of random forests is due to the high quality of each tree (at least in terms of bias) combined with the forest's small correlation, where the correlation between trees is defined as the ordinary correlation of predictions on so-called out-of-bag (OOB) samples. The difference in prediction accuracy before and after permuting x_j is assumed to be a measure of variable relevance, averaged over all trees.

Let B^t is the OOB samples for a tree (t) $\varepsilon t = \{1,2,\ldots n_{tree}\}$ and $(L(T_t(x_i),y_i)$ is the prediction accuracy at i th training example, then the VIRF for a variable x_j in a tree (t) can be calculated as:

$$\mathrm{VIRF}^t\left(x_j\right) = \sum_{i\varepsilon B^t}\left((L(T_t(x_i),y_i) - \left(L\left(T_t\left(x_{i,\pi_j}\right),y_i\right)\right)\right) \tag{14}$$

Where $x_{i,\pi_j} = = (x_{i,1},\ldots,x_{\pi_{j(i),j}},x_{i,j+1},\ldots,x_{i,p})$ and π_j is a random permutation of n integers. In regression tasks, the prediction accuracy is defined as the in the mean squared error (MSE). Finally, the VIRF for each variable is computed as the average importance over all trees as follow:

$$VIRF(x_j) = \frac{\sum_{i \epsilon B'} \mathrm{VIRF}^t(x_j)}{n_{tree}} \quad 15$$

3.4. Deep learning neural network (DLNN)

One of the most popular models of the Artificial Neural Networks ANNs in the field of deep learning is the multi-layered perceptron (MLP). MLP consists of three main layers: an input layer and an output layer with one or more hidden layers. The MLP can be called a DLNN if the hidden layer consists of more than two layers (Zemouri et al., 2020).

MLPs are fully connected ANNs, each node in one layer connects to every node in the following layer with a certain weight. Every node is a neuron with a non-linear activation function, except for the inputs' nodes. For training, MLP utilizes a supervised learning algorithm called backpropagation (Zemouri et al., 2020). DLNN can differentiate non-linear separable data thanks to its multi-layer, non-linear activation functions.

MLP has a linear neuron activation function that represents a linear network of functions connecting weighted inputs to the output. Also, Linear algebra demonstrates that any number of layers can be reduced to a two-input-output layer. The creation of the DLNN network via non-linear activation functions is therefore important in order to enhance the precise model of biological neurons and better imitate the working mechanism. In DLNN, the use of sigmoid activation functions with the following formulas was widely adopted:

$$y(v_i) = \tanh(v_i) \& y(v_i) = \frac{1}{1+e^{-v}} \quad 16$$

The first formula includes a hyperbolic tangent ranging from −1 to 1, while the second formula is a similarly-shaped logistic function with ranges from 0 to 1. Herein, y is the output of the i-th node (neuron) and v_i is the weighted sum of the input connections. Recently, numerous activation functions such as rectified linear unit (ReLU), Identity, Logistic, and Tanh are also proposed (Pham et al., 2020).

In MLP, the initial connection weights are chosen randomly and then fixed by the results of the training process. Then, backpropagation (BP) training process based on the least-squares average algorithm is used to minimize the error (e) between predicted outputs and observed outputs (in the training dataset) and maintaining good generality of the networks

$$e_i = d_i(\mathrm{n}) - y_i(n) \quad 17$$

Where, d_i (n) is the target (observations) at any n output node. Next, node weights can be adjusted based on corrections that minimize the error in the entire output, given by

$$\varepsilon(n) = \frac{1}{2}\sum_j e_j^2(n) \quad 18$$

Next, using gradient descent, the change in each weight will be

$$\Delta w_{ij}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial v_i(n)} y_i(n) \quad 19$$

Where $y_i(n)$ is the output of the previous neuron and η is the learning rate chosen to make sure weights converge rapidly without oscillations to the response. Next, the calculated derivative depends on the induced local field that differs. This derivative can be easily proven for an output node as

$$-\frac{\partial \varepsilon(n)}{\partial v_i(n)} = e_j(n)\varphi'(v_i(n)) \quad 20$$

Where, φ' is the derivative of the activation function. Finally, the corresponding derivative can be formulated with the changes in weight associated with a hidden node as:

$$-\frac{\partial \varepsilon(n)}{\partial v_i(n)} = \varphi' v_i(n) \sum -\frac{\partial \varepsilon(n)}{\partial v_k(n)} w_{kj}(n) \quad 21$$

This relies on the change in weights of the K-th nodes. This algorithm represents the reverse phase of back-propagation since the resulting weights vary according to the activation function derivative, and the hidden layer weights therefore changed.

3.5. Models' accuracy evaluation

Recently, different evaluation measures have been widely considered to evaluate the AI model's prediction capability. Among those, the RMSE (root mean square error), the CC (Pearson's correlation coefficient), these indices can be expressed as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - M_i)^2} \quad 22$$

$$CC = \frac{\sum_{i=1}^{n}(P_i - \overline{P})(M_i - \overline{M})}{\sqrt{\sum_{i=1}^{n}(P_i - \overline{M})^2 \sum_{i=1}^{n}(M_i - \overline{M})^2}} \quad 23$$

where (M_i), (P_i) are the measured and predicted values, n is the number of datasets, ($\overline{P}$) , ($\overline{M}$) are the average values of measured and predictions, respectively.

3.6. The methodology framework of proposed model

Fig. 2 presents the proposed framework flowchart for predicting the SLV, and the following are the steps of the proposed technique.

- **Decomposition**: a preprocessing harmonic analysis via t-tide algorithm is adopted to decompose effectively the measured sea level time series dataset to TSLV and NTSLV based on wavelet time-frequency analysis.
- **Dataset Exploration and preprocessing Models Evaluation:** an exploratory data analysis is performed for searching extensively on variable correlations and system complexity. Then, a preprocessing technique is performed via the standardization of datasets.
- **Building preliminary ML model:** an initial attempt using traditional ML algorithms is performed for developing a predictive model using classical ML algorithms.
- **FS process:** an efficient NCA algorithm is applied for FS purposes to reduce the data dimensionality and enhance the model's prediction capability.
- **DL predictive model:** a DLNN predictive model is developed based on the new datasets after the FS process for predicating NTSLV.
- **Proposed model evaluation:** numerous statistical and robust evaluation indicators are provided for deeply evaluating the prediction ability of the developed model. Furthermore, the proposed model results are compared with previous studies to confirm its superiority.

4. Results and discussion

Aiming at validating of the effectiveness of the developed model in the current study, several stages are established and presented in the following subsections.

4.1. Decomposition of sea level into tidal and non-tidal components

To understand the sea level trends and produce more precise SLR prediction, time-frequency analysis is initially implemented using Morlet CWT function in MATLAB software for stating the most significant tidal harmonic constituents; and then, a harmonic analysis is implemented using T-Tide algorithm for decomposing of tidal and non-

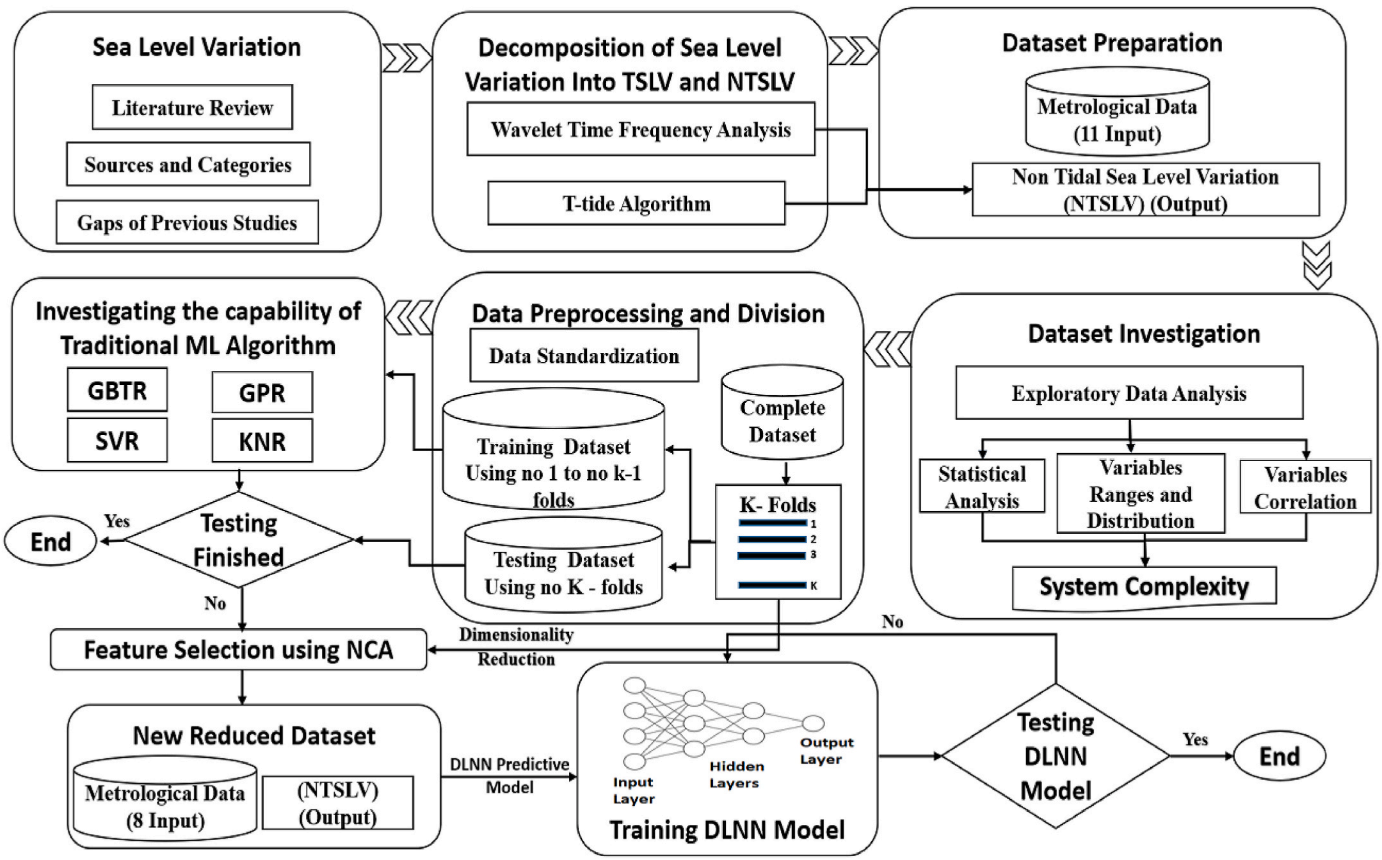

Fig. 2. Structured methodology and objectives of the current study.

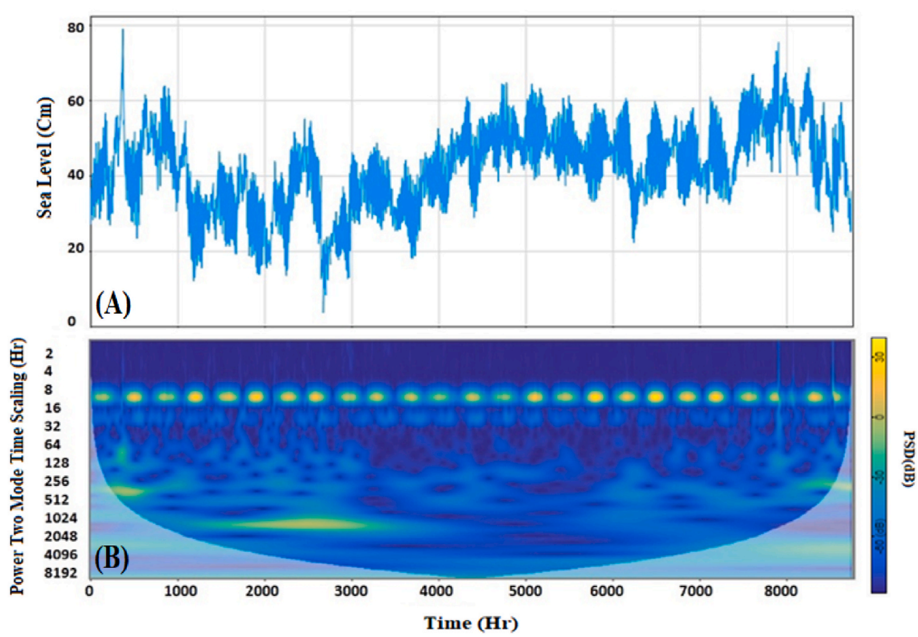

Fig. 3. Results of wavelet analysis on hourly sea-level records: (a) Time series of sea level and (b) Power spectrum WT analysis of sea level.

tidal sea level motions. Here, the power spectrum of sea level data is extracted based on time series measurements of sea level. From the time-frequency spectrum resulted from wavelet analysis, the magnitude and periods of the different frequency sea-level signals are presented in Fig. 3. In Fig. 3, the three parameters, measured time, signal period and the Power Spectrum Density (PSD), are presented to estimate the powerful signal frequency. The findings show that a significant energy content (high PSD) is concentrated on tidal frequencies which can be clearly observed at 12 h period, which demonstrates that the principal semi-diurnal tidal constituents (M_2 and S_2 (as presented in Table 1)) are the most significant tidal components, and this is consistent with the results of (Vu et al., 2010). Thus, these constituents will be considered during harmonic analysis.

On the other hand, a summary of the sea level variability can be computed by Eq. (24) as follow

$$X(t) = Z_0(t) + TSLV\ (t) + NTSLV(t) \quad (24)$$

where X(t) is the observed sea level, $Z_0(t)$ is the mean sea level, TSLV (t) is the tidal part of the variation, NTSLV (t) is the non-tidal (residual) part of the variation, and (t) is time. To extract the NTSLV from the above equation, harmonic analysis is performed using MATLAB® based T-Tide for the recorded data (Pawlowicz et al., 2002). The decomposition results of observed sea level are presented in Fig. 4, it can be seen that the annual NTSLV ranged from +13.5 cm to −16 cm.

4.2. Data exploration and preprocessing

After extracting NTSLV from the measured sea-level and based on the previous review wherein no previous study considered all possible factors which may affect NTSLV. The final input predictors for the current study as presented in Table 3 are (1) WS; (2) WD; (3) WR; (4) SST; (5) T; (6) RH; (7) P; (8)R; (9) SR; (10) DP and (11) WP. Also, NTSLV is the

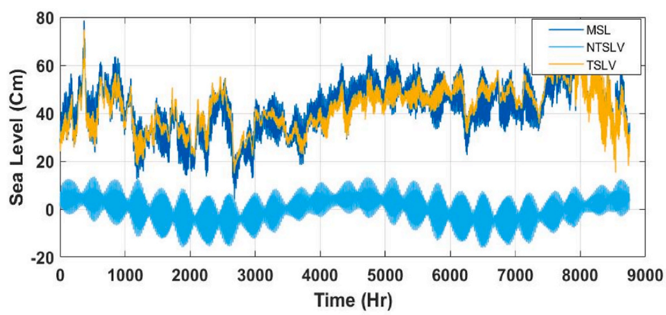

Fig. 4. Decomposition of measured sea-level into TSLV and NTSLV.

unique output as presented in Table 3. To gain maximum insight into the data set and its underlying structure, an Exploratory Data Analysis (EDA) is performed to take a bird's eye view of the data and tries to make some sense of it. For this purpose, a correlation heatmap matrix is executed and automatized by using Python scripting including Seaborn and Matplotlib powerful visualization libraries to describe visually the relation between the different variables as outlined in Fig. 5.

Additionally, dataset is investigated by means of statistical metrics, such as max, min, standard deviation and variables skewness as presented in the lower part of Table 3. The key insights from the conducted EDA in this section can be concluded as: (1) There is no strong correlation between any climatic variables and NTSLV as shown in Fig. 6 which indicate that NTSLV modeling is a highly complicated process, and (2) climatic variables are also interrelated in a complex way as shown by the weak interrelationships. These conclusions demonstrate the complexity of the studied system (i.e., the manner in which the predictors and response variables are altogether related), thereby asserting the complex nature of SLV and the associated climatic variables. This sophistication has led to the selection of a predictive data analysis method to create an efficient NTSLV predictive model, as will be investigated in the next sections.

In this research, ML algorithms are chosen from other methods for predictive data analysis such as statistical learning (SL), because of its ability to avoid the limitations of any strictly programmed model structure instructions and all hypothesis concerning distributions of data frequencies as exists in SL technique.

However, preprocessing procedures are required for data set to ensure its quality for processing with ML algorithms. First, according to the statistical skewness indicator in Table 3, most of the inputs are well distributed and suitable for training with ML algorithms. Second, it's well recognized that ML algorithms forecast recklessly for unscaled variables. Since the parameters presented in Table 3 have different magnitudes, units, and ranges (e.g., wind velocity (WS) is meters per second, while direction (WD) is in degree), this leads to magnitude conflicts, so that all variables must be added at the same scale. Data

Table 3
Hourly recorded metrological parameters and NTSLV with associated statistical indicators.

Inputs												Output
Datasets No.	WS	WD	WR	SST	T	RH	P	R	SR	DP	WP	NTSLV
Unit	m/s	Degree	Km	C^0	C^0	M	mbar	mm	W/m^2	C^0	C^0	Cm
1	3.596	209.8	0.036	17.75	13.55	63.6	1016	0	0	6.797	10	2.23
2	3.754	212.4	0.037	17.72	13.13	64.09	1016	0	0	6.503	9.68	2.06
3	3.083	217.2	0.031	17.65	12.7	66.02	1015	0	0	6.53	9.49	2.56
4	3.55	219.3	0.035	17.56	12.78	63.88	1015	0	0	6.128	9.35	3.61
5	4.033	206.2	0.041	17.48	11.96	65.73	1015	0	0	5.764	8.81	4.95
6	4.002	200.2	0.04	17.43	11.2	68.21	1014	0	0	5.573	8.36	6.12
7	4.952	204.7	0.05	17.38	10.81	68.57	1014	0	0	5.266	8.04	6.72
8	4.743	210.6	0.047	17.33	11	66.58	1015	0	0	5.03	8.02	6.55
9	4.413	211.4	0.044	17.25	11.27	64.72	1015	0	27	4.873	8.08	5.69
10	4.564	208.2	0.046	17.24	11.74	63	1016	0	182	4.933	8.34	4.29
11	4.054	216.6	0.04	17.23	12.65	61.49	1016	0	350	5.45	8.99	2.66
12	4.345	218	0.044	17.23	14.19	56.55	1015	0	490	5.683	9.79	1.26
13	4.168	222.6	0.042	17.24	15.82	51.16	1014	0	568	5.755	10.56	0.55
14	4.556	231.6	0.045	17.25	16.41	51.77	1014	0	587	6.476	11.12	0.77
. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....
. ....	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .
.... .	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....
.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .
23	4.227	229.2	0.042	17.21	13.36	68.59	1014	1.42	0	7.701	10.33	5.93
24	3.754	212.9	0.037	17.22	12.77	73.93	1013	0.02	0	8.25	10.33	4.01
.... .	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....	. ....
.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .	.... .
8757	3.035	205.3	0.031	17.81	14.71	62.12	1016	0	0	7.549	10.86	3.76
8758	3.03	201.9	0.03	17.78	14.14	63.36	1016	0	0	7.295	10.49	4.29
Min	0.338	0.015	0.003	14.25	7.263	9.9	1001	0	0	7.473	3.58	−16.03
Max	15.34	360	0.153	31.54	42.71	92	1026	14.24	976	25.29	26.21	13.53
Std	1.83	104.49	0.018	5.44	5.3	13.21	15.93	0.27	285.5	6.7	5.31	5.63
Skewness	0.74	−0.7	0.74	0.03	−0.22	−1.14	0.35	3.2	1.06	−0.47	−0.2	−0.32

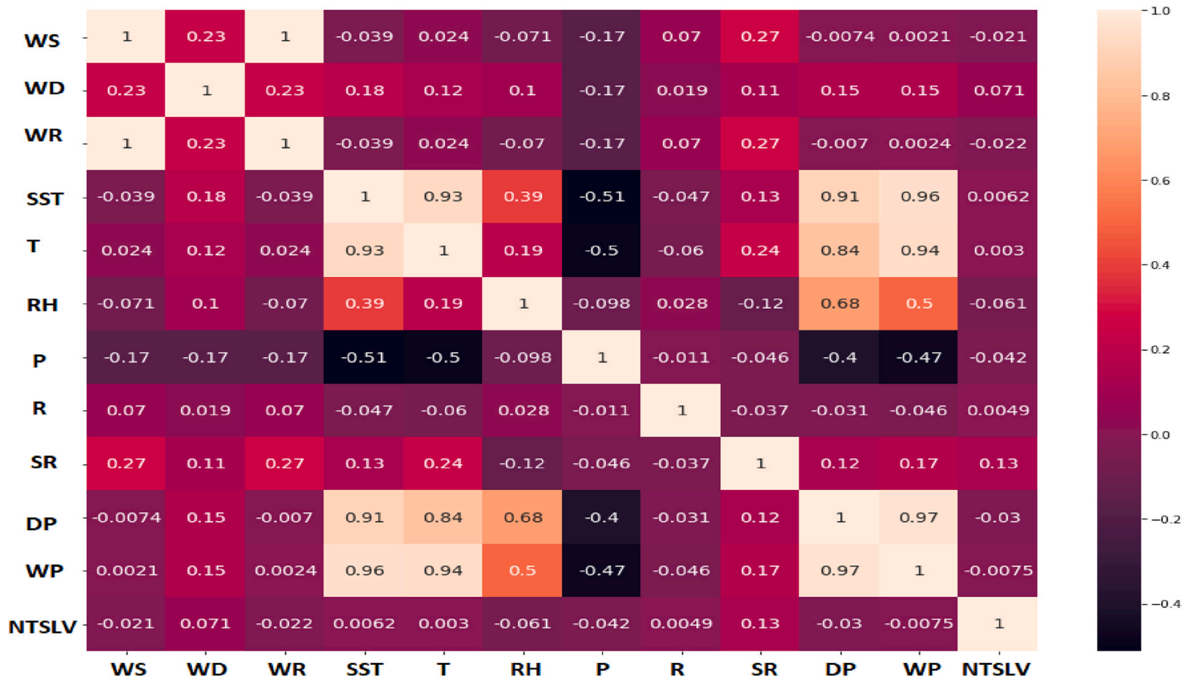

Fig. 5. Heatmap matrix of climatic variables with NTSLV.

standardization has therefore been used to solve this issue. Eq. (23). In addition, data standardization guarantees that all variables have a zero-based distribution to ensure efficient data processing.

$$\mathscr{Z} = \left(\frac{x-\mu}{\sigma}\right) \tag{23}$$

Where, $\mathscr{Z}$ is the scaled value of the unscaled variable x with a mean (μ) and a standard deviation (σ).

4.3. *Prediction capability of the traditional ML algorithms*

In the current stage, the ML approaches were designed and evaluated using MATLAB software. For evaluating the performance of ML algorithms, k-fold cross-validation has been applied as it is considered to be a reliable approach, mitigating the bias and variance in other validation methods such as a holdout.

In the k-fold validation technique, the complete dataset is split into k-separate subsets or folds that are approximately identical, where k is a positive integer (Seong et al., 2018). Then the holdout technique is repeated k times: one K folds is rotated every time during the test stage and the other K-1 folds are set up for training. This mechanism guarantees that the entire dataset is used in both the training and testing stages (Wei et al., 2013). Thus, the best number of K- folds will be investigated for each ML algorithm in the current study.

As an initial attempt for developing NTSLV predictive model, this section deals with the issues of selecting the right parameters and functions for the traditional ML Algorithms. Each ML algorithm has been run 50 times with different hyperparameters discussed previously in ML

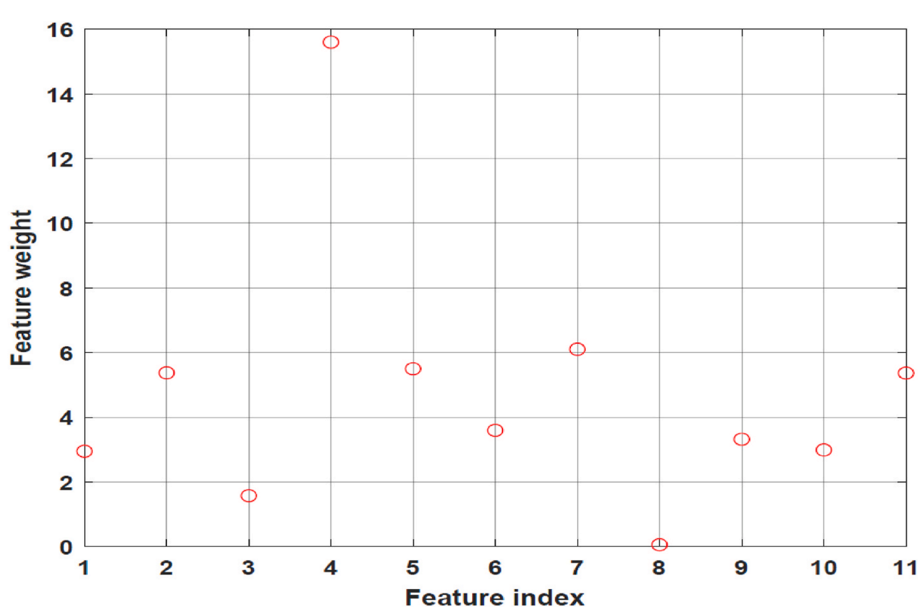

Fig. 6. Input features weights with default lambda values.

Table 4
Statistical measures for the proposed ML algorithm.

Model	Hyperparameters	Hyperparameters		Metrics			
		Investigated	Optimum	Training		Testing	
				RMSE	CC	RMSE	CC
GBTR	No of Leaf samples	(50–200)	50	2.82	0.71	3.84	0.51
	Learning rate	(0.1–0.8)	0.2				
	Number of K-fold	(4–10)	6				
SVR	Kernel Function	Linear, Quadratic, Cubic, Gaussian	Gaussian	2.71	0.75	3.63	0.54
	Cost Factor(C)	(0.1–0.8)	0.55				
	Number of K-fold	(4–10)	5				
GBR	Covariance Function	Rational Quadratic, Squared Exponential, Matern	Squared Exponential	2.31	0.81	2.61	0.76
	Desired Variance (σ)	(0.1–0.8)	0.5				
	Number of K-fold	(4–10)	5				
KNR	Distance metrics	Euclidean, Seuclidean, Cityblock, cosine	Cityblock	2.18	0.83	2.48	0.78
	Number of Neighbors	(4–8)	4				
	Number of K-fold	(4–10)	5				

Table 5
NCA feature selection initialization parameters.

Initial Parameters	Value
Number of iterations	50
Fitting Method	Exact
Type of Solver algorithm	Limited Broyden-Fletcher-Goldfarb-Shanno (LBFGS)
Dimension of Search Space	Number of Features
A regularization Parameter (λ)	$\frac{1}{n}$
Fitness function	KNR
Cost Function	MSE
Data Used	Training/Validation dataset

sections. Then the best performance with associated hyperparameters for each algorithm is presented in Table 4. Additionally, for validating the stability of the utilized ML models via the K-fold technique, another hold-out cross-validation technique is performed and its output is presented in Table Ain the supplementary material. It can be seen that the results of the two techniques are close, which reflects the stability of the models' performance.

In comparison to all ML Algorithms, GBTR and SVR show poor generalization performance with CC = 0.51 and 0.54 respectively for testing data. Also, it is evident that the KNNR algorithm is slightly better than the GBR algorithm. The best performance for the KNR algorithm is obtained with Distance metrics (Cityblock) and four neighbors with CC equals 0.78 for the testing data. Thus, it's inferred that the KNR model outperformed other ML algorithms. In general, it can be inferred that conventional ML can offer a little precision in large and complex datasets based on the preliminary results of the applied ML models. To overcome these issues, FS is utilized to solve the complexity of the dataset by reducing data dimensionality and hence increase the prediction capability. To this end, the superior KNR function with associated hyperparameters which resulted from this stage will be utilized as the fitness function during the FS process.

Table 6
Fine tuning of (λ) for improving NCA performance.

Run No.	K	Lambda (λ)	Average Loss (l_{avg})
1	4	0,0011	14.34222
2	5	0.00221	14.28872
3	6	0.00442	9.828932
4	7	0.00663	13.81533
5	8	0.008841	13.28255
6	9	0.011051	11.43779
7	4	0.013261	10.51073
8	5	0.015471	11.96259
9	6	0.017681	12.13465
10	7	0.019891	11.23715
11	8	0.022101	11.28845
12	9	0.024312	11.21878
13	4	0.026522	11.26239
14	5	0.028732	9.393789
15	6	0.030942	11.87378
16	7	0.033152	10.45427
17	8	0.035362	10.2462
18	9	0.037572	15.87254
19	4	0.039782	11.18401
20	5	0.041993	10.44958

4.4. Feature selection of the proposed model

In this section, the results of the FS process using the NCA algorithm are presented. The initialization parameters of NCA utilized in the current analysis are outlined in Table 5. The initial regularization parameter λ (lambda) is first set to a default value of $\frac{1}{n}$ where n is a total number of observations in the training set. As shown in Fig. 6, the feature weights of the 11 input parameters are determined. If its weight is greater than the tolerance value = 0,002, a parameter is considered significant. As seen from Fig. 6, only one parameter with irrelevant weight can be omitted from data space which seems not the significant effect on data dimensionality.

To improve the performance of the proposed NCA algorithm, the regularization parameter λ is tuned using different K- values via cross-validation. Numerous values of λ are investigated by including the standard deviation of the response ($std(y_{train})$) in the λ values which balances the default loss function ('mse', mean squared error) term and the regularization term in the objective function. Thus, the λ values are assumed and randomized as ($\lambda = \frac{\text{linspace}(0,50,20)*\text{std}(\text{ytrain})}{n}$). In addition, the dataset is divided into partitions equal to K. One partition is used for the testing process, while the remaining partitions are used for the training.

Throughout 20 independent runs, the NCA algorithm is applied as a function of (k, λ) and the average results of the loss (cost) function

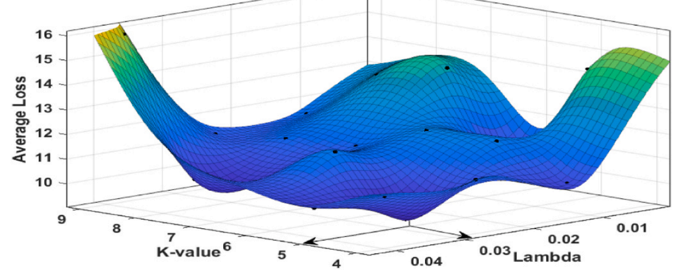

Fig. 7. 3D view of optimum hyperparameters for NCA algorithm.

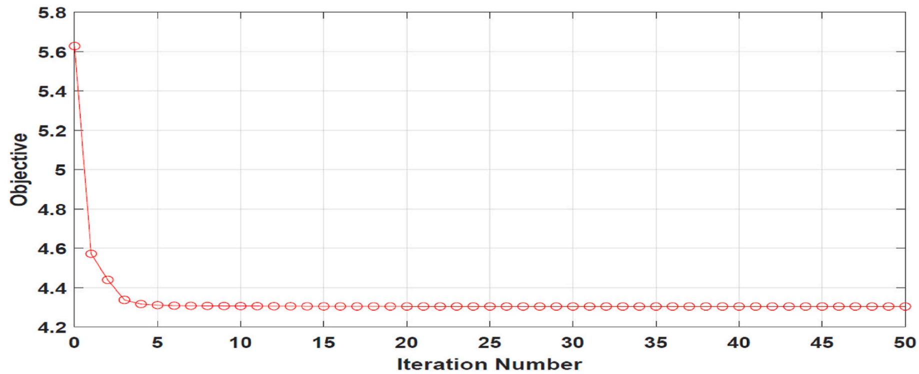

Fig. 8. Feature selection convergence process using the NCA algorithm.

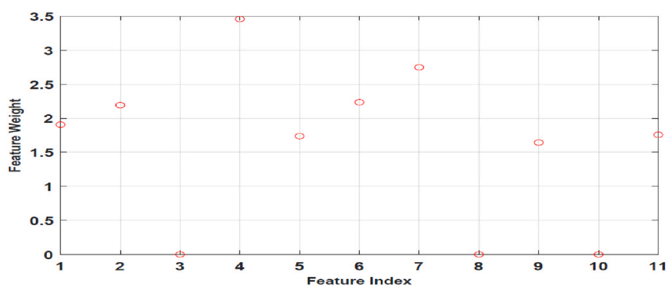

Fig. 9. Input features weights with optimum lambda values.

measurements represent final results are obtained as outlined in Table 6. As shown in Fig. 7, the NCA function included many points of local minima, but the unique global optimum value located at K = 5 and λ = 0.028.

Using the obtained optimum hyper parameters to refit the NCA regression algorithm, the evolution of MSE values after 50 iterations is shown in Fig. 8. The MSE value can be seen to have gradually decreased. At the first iteration, the lowest MSE is 5.62 Cm and decreased to 4.31 at the 5th iteration and remained unchanged till the end of the run. It's also visible from Fig. 9 that the number of irrelevant parameters has increased from 1 to 3, indicating that only the most significant features are chosen. As a result, eight input variables only will be included for a compact dataset.

For further investigation, the performance of NCA is validated by comparing its results with the embedded VIRF technique. It can be seen from Fig. 10 that the redundant variables are the third, eighth, and tenth ones as the same results obtained from NCA which reflect the accuracy obtained results from NCA.

4.5. DLNN prediction capability

In this section, the evolutionary results of the NTSLV prediction Via the DLNN model are assessed. To emphasize the role of the FS process in NTSLV modelling, two different scenarios are performed for comparison purposes. The first one uses the initial 11-input data set space, denoted as 11-input model DLNN. The second one uses the reduced dataset with 8-input, denoted with 8-input model DLNN.

The ideal DLNN architecture with the right number of hidden layers, neurons' number in each hidden layer and neurons' connected functions are universally difficult to identify. Intensive research has been conducted on a large and continuous discussion of this issue. Given that different possibilities for constructing the final network structure must be assembled in the DLNN algorithm, the process of proper selection of combined parameters is difficult to achieve. Therefore, in the current study, a total of 100 runs takes numerous DLNN parameters into account for tunning its performance as outlined in Table 7 will be investigated. The summary of the DLNN predictive capabilities model corresponding to 8 and 11 input parameters is shown in Table 8, it can be noticed that after the reduction of input space, the output of the DLNN model is enhanced clearly. The average CC value increased from 0.97 to 0.98

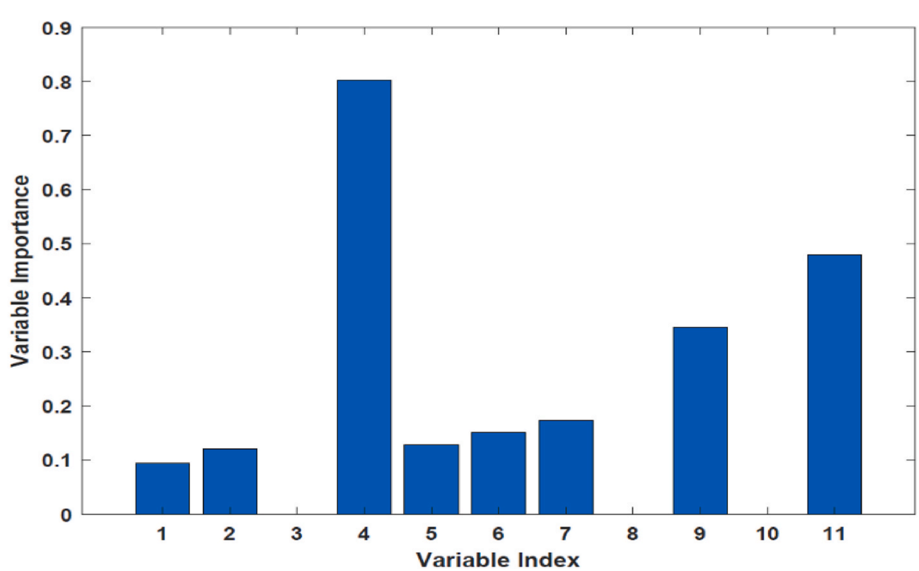

Fig. 10. Input variables importance with VIRF method.

Table 7
DLNN Parameters with their tuning ranges/options.

DLNN Parameters	Value
Optimizer algorithm	Quasi-Newton, Stochastic gradient descent, Adam
Activation function of hidden layers	Identity, Logistic, Tanh, Relu
Number of hidden layers	(2–10)
Number neurons in first hidden layer	(8–128)
Number neurons in second hidden layer	(8–128)
–	–
Number neurons in last hidden layer	(8–128)
Cost Function	MSE
Data Used	Training/Validation dataset

Table 8
DLNN Parameters with their tuning ranges/options.

Model	Dataset	Criteria	Min	Average	MAX	SD
11 -Input DLNN	Training	RMSE	0.21	0.43	0.59	0.095
		CC	0.90	0.93	0.97	0.0185
	Testing	RMSE	0.38	0.48	0.68	0.068
		CC	0.88	0.91	0.94	0.0113
8- Input DLNN	Training	RMSE	0.14	0.38	0.54	0.081
		CC	0.93	0.94	0.98	0.0161
	Testing	RMSE	0.22	0.38	0.52	0.065
		CC	0.91	0.94	0.97	0.0107

during the training phase. The most significant improvement can be observed in the testing set in which the average CC increased from 0.91 to 0.94. Moreover, for checking the stability of the proposed model, statistical indicators such as min, average, and standard deviation (SD) are calculated for CC and RMSE values. It can be observed on the testing stage, the 8-input DLNN with (SD = 0.0107) smaller than the 11-input DLNN model (SD = 0.0113) for CC values, indicating a more stable network for the 8-input DLNN model. Scatter plots are also provided to compare the performance of both DLNN models (Fig. 11). The figure shows that 8-input DLNN has a higher agreement between measured and predicted values than 11-input DLNN during the testing phase.

The results of the best models' predictability with associated optimal hyperparameters are summarized in Table 9. Also, Fig. 12 illustrates the evolution of the DLNN model through 5000 iterations for the best model performance in both scenarios. It can be seen that the 8-input DLNN model has better precision with (RMSE = 0.52 cm and CC = 0.97) on the testing phase. Additionally, Table 9 demonstrates that both models choose the same activation function (Relu) and the number of hidden layers varies between 4 and 10 with the associated number of neurons range between 9 and 80. Meanwhile, each model chooses a different type of algorithm for optimizing the network. It is also worth pointing

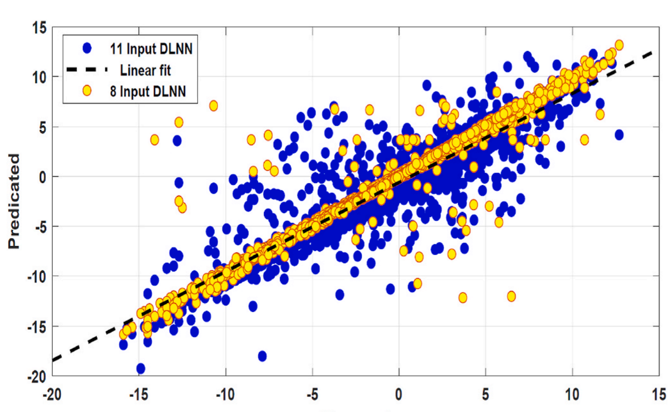

Fig. 11. Performance of the best DLNN models during testing stage.

Table 9
The optimum parameter of models with associated results.

DLNN Parameters	11 -Input DLNN	8- Input DLNN
RMSE	0.71	0.903
CC	0.66	0.95
Average Computational Time	52 min per run	35 min per run
Activation function of hidden layers	Relu	Relu
Number of hidden layers	6	4
Number neurons in each hidden layer	(128,64,64,64,64,32)	(128,64,64,32)
Network optimizer algorithm	Stochastic gradient descent	Adam

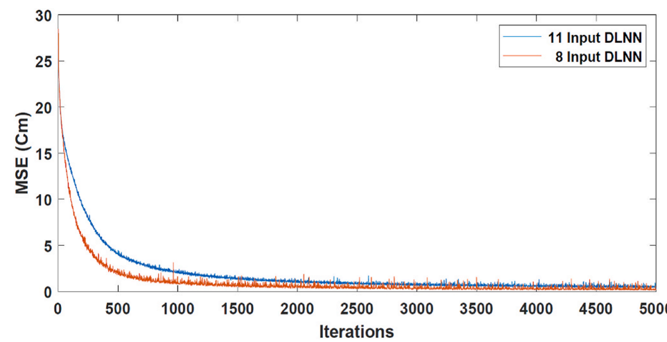

Fig. 12. Convergence curves for the best DLNN algorithms with 5000 iterations.

out that the average computational time for the 8-input DLNN models is much lower than the 10-input DLNN with a 35% lesser time.

For further investigation, the prediction capability for the proposed model is examined for prediction of NTSLV using new metrological dataset and compared with results of previous studies using CC indicator as presented in Table 10. The results of the current study as well as previous studies confirmed the effectiveness of ML techniques for predicating NTSLV for short periods. However, the current study outperforms other previous studies for long term prediction of NTSLV. Based on CC values, it should be noted that the proposed DLNN model (with CC = 0.96) enhances the prediction capability significantly by 23% in comparison with the initial attempt with the best ML model (with CC = 0.78) at 1 h lead time prediction. On the other hand, for long-term predication (e.g., 72h), the developed model in the current study (with CC = 0.91) is also improves the prediction capability by 19% in comparison with previous studies (with max CC = 0.76).

Table 10
Comparison between the current study with other studies.

Type of AI Approaches	Ref	Input Metrological Variables	Lead Time (hr)	Model Performance (CC)
ANN	Zubier and Eyouni (2020)	WD, WS, P, T, E, NTSLV	24	0.95
ANN, ARMA, ANFIS	Karimi et al. (2013)	MSL(t)	1, 24, 48, 72	0.99,0.96,0.88,0.76 Respectively
ANN	Tsai and You (2014)	WD, WS, P, MSL(t)	45	0.91
KNR (ML)	Current Study	WS, WD, WR, SST, T, RH, P, R, SR, DP, WP	1	0.78
DLNN	Current Study	WS, WD, SST, T, RH, P, R, SR, DP, WP	1, 24, 48, 72	0.96,0.95,0.92,0.91 Respectively

5. Conclusions

SLR as a result of climate change is a worldwide issue. Furthermore, on the Egyptian north-west coast, the SLR issue is becoming increasingly severe especially with the accelerating urbanization there. Thus, an accurate and reliable SLR model must be designed for early warning to help coastal authorities before hazardous occurrences.

Recently, numerous SLR forecasting models have been produced. Nonetheless, these models suffer from many deficiencies due to the sophisticated nature of the sea trend, these deficiencies including the simplicity of time series models based on historical recorded mean sea level data only that ignoring the importance of climatic variables on SLV. Also, for few studies that take climatic variables into account, those studies don't include all possible climatic variables that may affect NTSLV. Meanwhile, these models don't investigate deeply the behavior of SLV by decomposing measured sea level into tidal and nontidal components to precisely model SLV. So, the aforementioned models contain weak initial basics that have a huge impact on their predictive stability. Thus, in the current research, an integrated approach is proposed to improve the accuracy and stability of SLR prediction models.

Specifically, powerful data pre-processing techniques are applied, including WT time-frequency analysis alongside t-tide harmonic analysis is performed to decompose SLV into TSLV and NTSLV. Also, an EDA is performed to investigate the correlation and interdependence of the system variables. Next, four ML algorithms namely GBTR, SVR, GPR, and K-NNR are examined initially to investigate their capability to build an efficient NTSLV model, but those algorithms don't give the desired output. Thus, the NCA is designed for FS purposes to reduce data dimensionality by tuning its hyperparameters with the hope of achieving high accuracy. Finally, an efficient DLNN model is developed based on the compacted dataset from the FS process to enhance NTSLV predictive model. Furthermore, different statistical criteria and visual evaluations are performed to validate the proposed WT-DLNN model. The results confirm the superiority of the developed WT-DLNN model models as it enhances the prediction capability by 19% in comparison with previous studies especially for long-term forecasting of NTSLV (e.g., 72h). Therefore, integrated NCA and WT-DLNN can be used as a potential soft computing technique for modelling NTSLV. Whilst current research made substantial improvements in forecasting and modelling the SLV process, it is important not to overlook the limitations. The major limitation of this research is that it is only relevant to marine study locations in the Mediterranean Sea with similar climatic and hydrodynamic conditions. However, the approach adopted here can be applied to various research locations with varying environmental conditions. For future work, the used NCA as an FS tool in the proposed model needs to be tested with other metaheuristic algorithms like Simulated Annealing, Genetic Algorithms, and Particle Swarm Optimization to evaluate its performance.

Authorship contribution

Ahmed Alshouny: Investigation, Validation, Writing – review & editing, Funding acquisition. Mohamed T. Elnabwy: Conceptualization, Resources, Data curation (data assessment), Investigation, Methodology, Software, Formal analysis, Validation, Writing – original draft, Writing – review & editing. Mosbeh R. Kaloop: Data curation (data assessment), Investigation, Validation, review &editing. Ahmad Baik Validation, review. Yehia Miky: Investigation, Validation, review.

Software availability

The ML models' training is implemented through MATLAB Environment and Regression Learner of MATLAB R2020B. Also, the EDA and DLNN Python codes are available via the GitHub link: https://github.com/AHMEDALSHOUNY/The-EDA-and-DLNN-Python-codes.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project No. (IFPHI-234-137-2020) and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2022.105399.

References

Abubakar, A.G., Mahmud, M.R., Tang, K.K.W., Hussaini, A., Md Yusuf, N.H., 2019. A REVIEW of MODELLING APPROACHES on TIDAL ANALYSIS and PREDICTION. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. International Society for Photogrammetry and Remote Sensing, pp. 23–34. https://doi.org/10.5194/isprs-archives-XLII-4-W16-23-2019.

Afshar-Kaveh, N., Nazarali, M., Pattiaratchi, C., 2020. Relationship between the Persian gulf sea-level fluctuations and meteorological forcing. J. Mar. Sci. Eng. 8 https://doi.org/10.3390/JMSE8040285.

Ali Ghorbani, M., Khatibi, R., Aytek, A., Makarynskyy, O., Shiri, J., 2010. Sea water level forecasting using genetic programming and comparing the performance with Artificial Neural Networks. Comput. Geosci. 36, 620–627. https://doi.org/10.1016/j.cageo.2009.09.014.

Breiman, L., 2001. Random forests. Mach. Learn. 451 45, 5–32. https://doi.org/10.1023/A:1010933404324, 2001.

Cervantes, J., Li, X., Yu, W., Li, K., 2008. Support vector machine classification for large data sets via minimum enclosing ball clustering. Neurocomputing 71, 611–619. https://doi.org/10.1016/j.neucom.2007.07.028.

Chao, W.T., Young, C.C., Hsu, T.W., Liu, W.C., Liu, C.Y., 2020. Long-lead-time prediction of storm surge using artificial neural networks and effective typhoon parameters: revisit and deeper insight. Water (Switzerland) 12. https://doi.org/10.3390/W12092394.

Church, J.A., Clark, P.U., Cazenave, A., Gregory, J.M., Jevrejeva, S., Levermann, A., Merrifield, M.A., Milne, G.A., Nerem, R.S., Nunn, P.D., Payne, A.J., Pfeffer, W.T., Stammer, D., Unnikrishnan, A.S., 2013. Sea Level Change.

Das, S., Roy, S., Sambasivan, R., 2018. Fast Gaussian process regression for big data. Big Data Res 14, 12–26. https://doi.org/10.1016/j.bdr.2018.06.002.

El-Diasty, M., Al-Harbi, S., Pagiatakis, S., 2018. Hybrid harmonic analysis and wavelet network model for sea water level prediction. Appl. Ocean Res. 70, 14–21. https://doi.org/10.1016/j.apor.2017.11.007.

Enríquez, A.R., Marcos, M., Álvarez-Ellacuría, A., Orfila, A., Gomis, D., 2017. Changes in beach shoreline due to sea level rise and waves under climate change scenarios: application to the Balearic Islands (western Mediterranean). Nat. Hazards Earth Syst. Sci. 17, 1075–1089. https://doi.org/10.5194/nhess-17-1075-2017.

Filippo, A., Rebelo Torres, A., Kjerfve, B., Monat, A., 2012. Application of Artificial Neural Network (ANN) to improve forecasting of sea level. Ocean Coast Manag. 55, 101–110. https://doi.org/10.1016/j.ocecoaman.2011.09.007.

Franco-Arcega, A., Carrasco-Ochoa, J.A., Sánchez-Díaz, G., Fco Martínez-Trinidad, J., 2012. Building fast decision trees from large training sets. Intell. Data Anal. 16, 649–664. https://doi.org/10.3233/IDA-2012-0542.

Gregorutti, B., Michel, B., Saint-Pierre, P., 2017. Correlation and variable importance in random forests. Stat. Comput. 27, 659–678. https://doi.org/10.1007/s11222-016-9646-1.

Grinsted, A., Moore, J.C., Jevrejeva, S., 2004. Application of the cross wavelet transform and wavelet coherence to geophysical time series. Nonlinear Process Geophys. 11, 561–566. https://doi.org/10.5194/npg-11-561-2004.

Guo, Z., Pan, H., Cao, A., Lv, X., 2018. A harmonic analysis method adapted to capturing slow variations of tidal amplitudes and phases. Continent. Shelf Res. 164, 37–44. https://doi.org/10.1016/j.csr.2018.06.005.

Huang, D., Wu, Z., 2017. Forecasting outpatient visits using empirical mode decomposition coupled with backpropagation artificial neural networks optimized by particle swarm optimization. PLoS One 12, 1–17. https://doi.org/10.1371/journal.pone.0172539.

Hzami, A., Heggy, E., Amrouni, O., Mahé, G., Maanan, M., Abdeljaouad, S., 2021. Alarming coastal vulnerability of the deltaic and sandy beaches of North Africa. Sci. Rep. 11 https://doi.org/10.1038/s41598-020-77926-x.

Imani, M., Kao, H.C., Lan, W.H., Kuo, C.Y., 2018. Daily sea level prediction at Chiayi coast, Taiwan using extreme learning machine and relevance vector machine. Global Planet. Change 161, 211–221. https://doi.org/10.1016/j.gloplacha.2017.12.018.

Ishida, K., Tsujimoto, G., Ercan, A., Tu, T., Kiyama, M., Amagasaki, M., 2020. Hourly-scale coastal sea level modeling in a changing climate using long short-term memory neural network. Sci. Total Environ. 720, 137613. https://doi.org/10.1016/j.scitotenv.2020.137613.
Jan, C.D., Tseng, C.M., Wang, J.S., Cheng, Y.H., 2006. Empirical relation between the typhoon surge deviation and the corresponding typhoon characteristics: a case study in Taiwan. J. Mar. Sci. Technol. 11, 193–200. https://doi.org/10.1007/s00773-006-0220-y.
Jiang, S.L., Shen, X., Zheng, Z., 2019. Gaussian process-based hybrid model for predicting oxygen consumption in the converter steelmaking process. Processes 7. https://doi.org/10.3390/pr7060352.
Karimi, S., Kisi, O., Shiri, J., Makarynskyy, O., 2013. Neuro-fuzzy and neural network techniques for forecasting sea level in Darwin Harbor, Australia. Comput. Geosci. 52, 50–59. https://doi.org/10.1016/j.cageo.2012.09.015.
Kim, S., Matsumi, Y., Pan, S., Mase, H., 2016. A real-time forecast model using artificial neural network for after-runner storm surges on the Tottori coast, Japan. Ocean. Eng. 122, 44–53. https://doi.org/10.1016/j.oceaneng.2016.06.017.
Kurniawan, A., Tay, S.H.X., Ooi, S.K., Babovic, V., Gerritsen, H., 2015. Analyzing the physics of non-tidal barotropic sea level anomaly events using multi-scale numerical modelling in Singapore regional waters. J. Hydro-Environment Res. 9, 404–419. https://doi.org/10.1016/j.jher.2014.10.005.
Lange, W.P. De, Carter, R.M., Peiser, B., 2014. Living with Uncertainty SEA-LEVEL CHANGE.
Li, Y., Han, W., 2015. Decadal sea level variations in the Indian Ocean investigated with HYCOM: roles of climate modes, ocean internal variability, and stochastic wind forcing. J. Clim. 28, 9143–9165. https://doi.org/10.1175/JCLI-D-15-0252.1.
Majdzadeh Moghadam, F., 2017. Neural network-based approach for identification of meteorological factors affecting regional sea-level anomalies. J. Hydrol. Eng. 22, 04016058 https://doi.org/10.1061/(asce)he.1943-5584.0001472.
Makarynska, D., Makarynskyy, O., 2008. Predicting sea-level variations at the Cocos (Keeling) Islands with artificial neural networks. Comput. Geosci. 34, 1910–1917. https://doi.org/10.1016/j.cageo.2007.12.004.
Meena, B.L., Agrawal, J.D., 2015. Tidal level forecasting using ANN. In: Procedia Engineering. Elsevier Ltd, pp. 607–614. https://doi.org/10.1016/j.proeng.2015.08.332.
Muslim, T.O., Ahmed, A.N., Malek, M.A., Abdulmohsin Afan, H., Khaleel Ibrahim, R., El-Shafie, Amr, Sapitang, M., Sherif, M., Sefelnasr, A., El-Shafie, Ahmed, 2020a. Investigating the influence of meteorological parameters on the accuracy of sea-level prediction models in sabah, Malaysia. Sustainability 12, 1193. https://doi.org/10.3390/su12031193.
Muslim, T.O., Ahmed, A.N., Malek, M.A., Afan, H.A., Ibrahim, R.K., El-Shafie, Amr, Sapitang, M., Sherif, M., Sefelnasr, A., El-Shafie, Ahmed, 2020b. Investigating the influence of meteorological parameters on the accuracy of sea-level prediction models in Sabah, Malaysia. Sustain. Times 12. https://doi.org/10.3390/su12031193.
Ou, T.-C., Su, W.-F., Liu, X.-Z., Huang, S.-J., Tai, T.-Y., 2016. A modified bird-mating optimization with hill-climbing for connection decisions of transformers. Energies 9, 671. https://doi.org/10.3390/en9090671.
Ou, T.C., Hong, C.M., 2014. Dynamic operation and control of microgrid hybrid power systems. Energy 66, 314–323. https://doi.org/10.1016/j.energy.2014.01.042.
Ou, T.C., Lu, K.H., Huang, C.J., 2017. Improvement of transient stability in a hybrid power multi-system using a designed NIDC (Novel Intelligent Damping Controller). Energies 10. https://doi.org/10.3390/en10040488.
Panagopoulos, A., 2021. Water-energy nexus: desalination technologies and renewable energy sources. Environ. Sci. Pollut. Res. 28, 21009–21022. https://doi.org/10.1007/s11356-021-13332-8.
Panagopoulos, A., Haralambous, K.J., 2020. Environmental impacts of desalination and brine treatment - challenges and mitigation measures. Mar. Pollut. Bull. 161, 111773. https://doi.org/10.1016/j.marpolbul.2020.111773.
Pancheva, D., Mukhtarov, P., 2000. Wavelet analysis on transient behaviour of tidal amplitude fluctuations observed by meteor radar in the lower thermosphere above Bulgaria. Ann. Geophys. 18, 316–331. https://doi.org/10.1007/s00585-000-0316-3.
Pasarić, M., Pasarić, Z., Orlić, M., 2000. Response of the Adriatic sea level to the air pressure and wind forcing at low frequencies (0.01-0.1 cpd). J. Geophys. Res. Ocean. 105, 11423–11439. https://doi.org/10.1029/2000JC900023.
Patil, S.G., Mandal, S., Hegde, A.V., 2012. Genetic algorithm based support vector machine regression in predicting wave transmission of horizontally interlaced multi-layer moored floating pipe breakwater. Adv. Eng. Software 45, 203–212. https://doi.org/10.1016/j.advengsoft.2011.09.026.
Pawlowicz, R., Beardsley, B., Lentz, S., 2002. Classical tidal harmonic analysis including error estimates in MATLAB using TDE. Comput. Geosci. 28, 929–937. https://doi.org/10.1016/S0098-3004(02)00013-4.
Pekel, E., 2020. Estimation of soil moisture using decision tree regression. Theor. Appl. Climatol. 139, 1111–1119. https://doi.org/10.1007/s00704-019-03048-8.
Pham, T.A., Tran, V.Q., Vu, H.L.T., Ly, H.B., 2020. Design deep neural network architecture using a genetic algorithm for estimation of pile bearing capacity. PLoS One 15. https://doi.org/10.1371/journal.pone.0243030.
Almaliki, Z.A., 2019. Do you know how to choose the right machine learning algorithm among 7 different types?. Towards Data Science [WWW Document], URL https://towardsdatascience.com/do-you-know-how-to-choose-the-right-machine-learning-algorithm-among-7-different-types-295d0b0c7f60 (accessed 11.13.20).
Raicich, F., 2010. On the contributions of atmospheric pressure and wind to daily sea level in the northern Adriatic Sea. Continent. Shelf Res. 30, 1575–1581. https://doi.org/10.1016/j.csr.2010.05.017.
Rajasekaran, S., Gayathri, S., Lee, T.L., 2008. Support vector regression methodology for storm surge predictions. Ocean. Eng. 35, 1578–1587. https://doi.org/10.1016/j.oceaneng.2008.08.004.
Salim, A.M., Dwarakish, G.S., Liju, K.V., Thomas, J., Devi, G., Rajeesh, R., 2015. Weekly prediction of tides using neural networks. In: Procedia Engineering. Elsevier Ltd, pp. 678–682. https://doi.org/10.1016/j.proeng.2015.08.351.
Sammari, C., Koutitonsky, V.G., Moussa, M., 2006. Sea level variability and tidal resonance in the Gulf of Gabes, Tunisia. Continent. Shelf Res. 26, 338–350. https://doi.org/10.1016/j.csr.2005.11.006.
Seong, H., Son, H., Kim, C., 2018. A comparative study of machine learning classification for color-based safety vest detection on construction-site images. KSCE J. Civ. Eng. 22, 4254–4262. https://doi.org/10.1007/s12205-017-1730-3.
Sharaan, M., Udo, K., 2020. Projections of future beach loss along the mediterranean coastline of Egypt due to sea-level rise. Appl. Ocean Res. 94, 101972. https://doi.org/10.1016/j.apor.2019.101972.
Shen, J., Wang, H., Sisson, M., Gong, W., 2006. Storm tide simulation in the Chesapeake Bay using an unstructured grid model. Estuar. Coast Shelf Sci. 68, 1–16. https://doi.org/10.1016/j.ecss.2005.12.018.
Tilburg, C.E., Garvine, R.W., 2004. A simple model for coastal sea level prediction. Weather Forecast. 19, 511–519. https://doi.org/10.1175/1520-0434(2004)019<0511:ASMFCS>2.0.CO;2.
Tsai, C.P., You, C.Y., 2014. Development of models for maximum and time variation of storm surges at the Tanshui estuary. Nat. Hazards Earth Syst. Sci. 14, 2313–2320. https://doi.org/10.5194/nhess-14-2313-2014.
Turki, I., Laignel, B., Kakeh, N., Chevalier, L., Costa, S., 2015. A new hybrid model for filling gaps and forecast in sea level: application to the eastern English Channel and the North Atlantic Sea (western France). Ocean Dynam. 65, 509–521. https://doi.org/10.1007/s10236-015-0824-z.
Vu, P.L., Ha, M.C., Darrozes, J., Bonneton, P., 2010. Identifying 2010 xynthia storm signature in GNSS-R-based tide records 1–16.https://doi.org/10.3390/rs11070782.
Wang, Bao, Wang, Bin, Wu, W., Xi, C., Wang, J., 2020. Sea-water-level prediction via combined wavelet decomposition, neuro-fuzzy and neural networks using SLA and wind information. Acta Oceanol. Sin. 39, 157–167. https://doi.org/10.1007/s13131-020-1569-1.
Wei, Z., Wang, W., Bradfield, J., Li, J., Cardinale, C., Frackelton, E., Kim, C., Mentch, F., Van Steen, K., Visscher, P.M., Baldassano, R.N., Hakonarson, H., 2013. Large sample size, wide variant spectrum, and advanced machine-learning technique boost risk prediction for inflammatory bowel disease. Am. J. Hum. Genet. 92, 1008–1012. https://doi.org/10.1016/j.ajhg.2013.05.002.
Yang, W., Wang, K., Zuo, W., 2012. Neighborhood component feature selection for high-dimensional data. J. Comput. 7, 162–168. https://doi.org/10.4304/jcp.7.1.161-168.
Zemouri, R., Omri, N., Fnaiech, F., Zerhouni, N., Fnaiech, N., 2020. A new growing pruning deep learning neural network algorithm (GP-DLNN). Neural Comput. Appl. 32, 18143–18159. https://doi.org/10.1007/s00521-019-04196-8.
Zhang, S., Cheng, D., Deng, Z., Zong, M., Deng, X., 2018a. A novel kNN algorithm with data-driven k parameter computation. Pattern Recogn. Lett. 109, 44–54. https://doi.org/10.1016/j.patrec.2017.09.036.
Zhang, S., Li, X., Zong, M., Zhu, X., Wang, R., 2018b. Efficient kNN classification with different numbers of nearest neighbors. IEEE Transact. Neural Networks Learn. Syst. 29, 1774–1785. https://doi.org/10.1109/TNNLS.2017.2673241.
Zubier, K.M., Eyouni, L.S., 2020. Investigating the role of atmospheric variables on sea level variations in the eastern central red sea using an artificial neural network approach. Oceanologia 62, 267–290. https://doi.org/10.1016/j.oceano.2020.02.002.