Improving a shoreline forecasting model with Symbolic Regression

Mahmoud Al Najar, Rafael Almar, Erwin W J Bergsma, Jean-Marc Delvit, Dennis G Wilson

IMPROVING A SHORELINE FORECASTING MODEL WITH SYMBOLIC REGRESSION

Mahmoud Al Najar
Laboratory of Spatial Geophysics and Oceanography Studies (CNES/CNRS/IRD/UPS)
University of Toulouse
mahmoud.al.najar@legos.obs-mip.fr

Rafael Almar
Laboratory of Spatial Geophysics and Oceanography Studies (CNES/CNRS/IRD/UPS)
University of Toulouse

Erwin W. J. Bergsma
Earth Observation Lab, The French Space Agency (CNES)

Jean-Marc Delvit
Earth Observation Lab, The French Space Agency (CNES)

Dennis G. Wilson
ISAE-SUPAERO, University of Toulouse

ABSTRACT

Given the current context of climate change and the increasing population densities at coastal zones around the globe, there is an increasing need to be able to predict the development of our coasts. Recent advances in artificial intelligence allow for automatic analysis of observational data. Symbolic Regression (SR) is a type of Machine Learning algorithm that aims to find interpretable symbolic expressions that can explain relations in the data. In this work, we aim to study the problem of forecasting shoreline change using SR. We make use of Cartesian Genetic Programming (CGP) in order to encode and improve upon ShoreFor, a physical shoreline prediction model. During training, CGP individuals are evaluated and selected according to their predictive score at five different coastal sites. This work presents a comparison between a CGP-evolved model and the base ShoreFor model. In addition to evolution's ability to produce well-performing models, it demonstrates the usefulness of SR as a research tool to gain insight into the behaviors of shorelines in various geographical zones.

1 INTRODUCTION

Coasts around the globe are continuously facing natural and anthropogenic pressures. Our knowledge and understanding of the evolution of the coastal zone over time is crucial for a large variety of applications including coastal risk monitoring and management. Shoreline evolution forecasting is an important element in coastal studies that aims to better understand and predict the occurrence and intensity of erosive and accretive forces. Recently, large efforts have been made to understand and predict shoreline evolution due to the rising social, economic and natural pressures such as climate change [1, 2, 3, 4].

Three main types of methods have been proposed and discussed in the literature on the topic of forecasting shoreline change [1]. Process-based models simulate multiple physical processes which contribute to shoreline change. The simulations are coupled through mass and momentum conservation laws. Such models include MIKE 21 [5], Delft3D [6], ROMS [7] and CROCO [8]. In general, these models are used to model short-term local events in the nearshore zone and are not considered applicable over larger spatio-temporal scales [1, 9]. Hybrid models are mixed approaches to modelling shoreline change that incorporate general physical principles and which are calibrated

using data-driven approaches (e.g. least-squares-fit). These models can be used to predict shoreline position over much longer time scales, however they are generally unable to generalize to previously-unseen areas and conditions and require site-specific field data for model calibration. Finally, data-driven techniques range from simple regression methods to modern deep learning models, which have demonstrated impressive capabilities over a wide variety of applications [10, 11, 12]. Artificial Neural Networks (ANN's) are the most commonly used models in similar works employing AI for shoreline forecasting [13, 14]. While ANN's have helped achieve significant advances in many domains, explaining their predictions remains relatively difficult due to their black-box nature [15, 16].

Symbolic Regression (SR) is a domain of Machine Learning (ML) algorithms that search for symbolic representations of the relationships embedded in the data. Evolutionary algorithms, and specifically Genetic Programming (GP), are often used for SR. GP operates by composing a predefined set of functions in a tree, graph, or other structure; the composition of functions is determined by an evolutionary algorithm. As the optimized model is a functional graph or tree, GP is considered an interpretable ML technique that can be used to derive simple symbolic forms of relationships in data. GP has been demonstrated to be competitive with machine learning approaches such as gradient boosting [17] and has been applied to a wide variety of problems [18, 19, 20, 21, 22, 23, 24].

This work frames the problem of forecasting shoreline change as a data-driven symbolic regression task. We make use of Cartesian Genetic Programming (CGP) [25] and the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [26] to encode and evolve interpretable shoreline change models. To promote model generalization, the evolved models are optimized to maximize prediction accuracy at five different coastal sites from around the globe. Evolution discovers new shoreline forecasting models which are competitive with existing physical models across the five studied sites.

2 METHODS

2.1 SHOREFOR

ShoreFor [27] is a shoreline change forecasting model that is built upon the principle of shoreline equilibrium [28], where shorelines continuously evolve towards a time-varying equilibrium condition. ShoreFor can be formulated according to: $\frac{dx}{dt} = c(F^+ + rF^-) + b$, where dx/dt is the rate of shoreline change, F is the magnitude of wave forcing, c and b are model free parameters that are optimized using a least-squares-fit minimizing the root-mean-squared-error (RMSE) of the model. The wave forcing term $F = P^{0.5}\frac{\Delta\Omega}{\sigma_{\Delta\Omega}}$ is expressed in terms of the wave energy flux P and the normalized disequilibrium term $\Delta\Omega/\sigma_{\Delta\Omega}$ using Ω_{eq} (Equation 1) to represent beach equilibrium as a weighted average of antecedent dimensionless fall velocities where ϕ is a model-free parameter which controls the number of days in the series used to estimate the current equilibrium state.

$$\Omega_{eq} = \frac{\sum_{i=1}^{2\phi} \Omega_i 10^{-i/\phi}}{\sum_{i=1}^{2\phi} 10^{-i/\phi}} \tag{1}$$

Ω, defined as $\Omega = \frac{H_{s,b}}{wT_p}$, represents the rate of sedimentation and is a function of the sediment grain settling velocity w, the breaking wave height $H_{s,b}$ and wave period T_p. Disequilibrium, $\Delta\Omega = \Omega_{eq} - \Omega$, is used to partition forcing F into accretion and erosion (F^+, F^-) according to the sign of $\Delta\Omega$. The erosion ratio $r = \left| \frac{\sum_{i=0}^{N}\langle F_i^+\rangle}{\sum_{i=0}^{N}\langle F_i^-\rangle} \right|$ is defined as a ratio between the detrended accretive and erosive wave forcing. It is calculated over the full wave forcing time series and treated as a constant to balance the accretion and erosion terms within the ShoreFor model. ShoreFor has been used in multiple shoreline prediction studies and a number of extensions have been proposed to improve its performance by accounting for shoreline change over different time-scales [29] as well as alongshore sediment transport processes [30, 31]. We make use of the ShoreFor model as a base for our experiments on the use of CGP for shoreline forecasting in a GI setting, and we highlight the possibility of extending the base CGP-ShoreFor implementation to account for these additional processes. The implementation of ShoreFor as a CGP individual is detailed in Appendix A

2.2 CARTESIAN GENETIC PROGRAMMING

Cartesian Genetic Programming (CGP) [25] is a form of genetic programming that encodes programs as directed acyclic graphs. An individual CGP graph is composed of input nodes, output

nodes and computation nodes. To represent a program as a genome, each node in the graph is associated with integers corresponding to the function of the node and its inputs; two-arity functions like $x + y$ are most common and used here, so each node is represented by three integers. These integers are optimized by evolution by constructing a graph which connects output nodes to input nodes and which is then evaluated using an objective function. A CGP genome is of fixed size, however the program graph can be of variable size as only a subset of nodes are connected to the output nodes. This allows for flexibility in the number of nodes an individual can use, where nodes that do not contribute to the output are simply ignored during evaluation. CGP has been successfully applied to a wide range of problems [32, 33, 34, 35, 36].
In this work, we use a mostly standard CGP representation with modifications to optimization for NSGA-II and modifications to the function set in order to represent ShoreFor. Other modifications to the CGP algorithm are described in Appendix B. We make use of NSGA-II with five fitness dimensions in order to rank our CGP individuals during evolution according to their predictive skills at five different coastal sites. A brief explanation of NSGA-II is presented in Appendix C. We invite the reader to refer to the original work in [26] for further details on the NSGA-II algorithm.

3 Experimental setup & results

3.1 Study Sites

Shoreline datasets from five different sites around the globe covering different coastal settings are used in this work. As shown in Figure 1, these sites include the Grand Popo beach in Benin, Gulf of Guinea, in West Africa. Truc Vert beach in the Aquitaine region of France. Narrabeen beach on the coastline of the Sydney metropolitan area, on the eastern coast of Australia. In addtion to two different sites from the USA, Duck NC on the eastern coast, and Torrey Pines on the western coast.

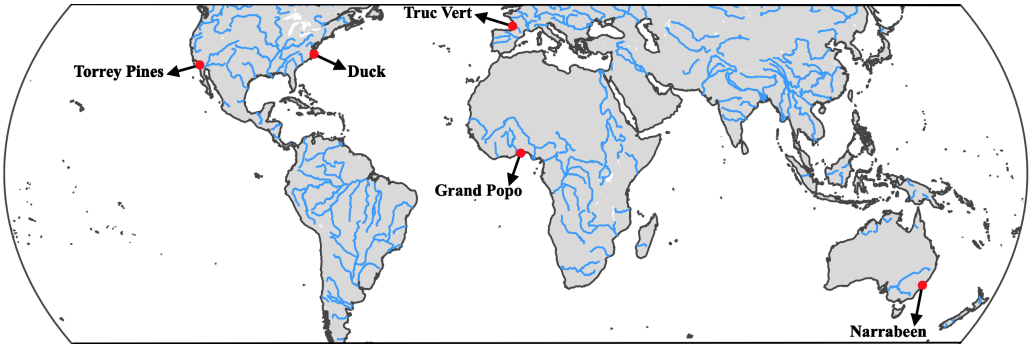

Figure 1: World map highlighting the locations of the five sites included in this study.

A mixture of techniques were used to gather these datasets. The shoreline time series are constructed using both in-situ GPS surveys and image processing to derive shoreline positions from video. The input time series were gathered from wave-bouys, satellite altimetry, as well as global model hindcasts (ISBA-CTRIP). The Ω and P input time series were computed according to [27].

3.2 Fitness Function

In this work, we make use of a modified version of the Mielke skill test as proposed in [37] in order to evaluate the performances of the models. The modified version of the Mielke skill test is formulated as $\lambda = 1 - \frac{N^{-1}\sum_{i=1}^{N}(o_i - m_i)^2}{\sigma_o^2 + \sigma_m^2 + (\hat{o} - \hat{m})^2}$, where o and m are the target and modelled shoreline location values respectively, $\hat{o}$ and $\hat{m}$ their mean values, N the sample size and σ the variance. This metric is an extension of the Pearson correlation coefficient (r) that downgrades the value of r according to the bias between the two datasets. This score is used in order to evaluate the fitness of our CGP individuals over the calibration period during evolution, where the objective is to maximize their Mielke score. We also use it to evaluate the forecast performances of the individuals after evolution as presented in Section 3.3.

3.3 Model Evaluation

This section presents and analyzes the performance of the top performing generalist model produced by CGP and NSGA-II over the test period (i.e. the model with the highest mean Mielke skill accross

the five datasets over the test period). Figure 3 in Appendix E presents and compares the predicted time series produced by ShoreFor and CGP over the full target time series. Table 1 presents a numerical comparison of the Mielke skill of both models over the train and test periods.

	λ_{train}		λ_{test}	
Site	ShoreFor	CGP	ShoreFor	CGP
Grand Popo	0.65	**0.73**	0.51	0.51
Narrabeen	0.44	**0.56**	-0.04	**0.11**
Duck	0.22	**0.49**	0.56	**0.67**
Torrey Pines	0.36	**0.48**	0.14	**0.23**
Truc Vert	0.05	0.05	**0.34**	0.06

Table 1: Comparison of the Mielke skill scores of ShoreFor and CGP-evolved model over the training and test periods at the five coastal sites.

Compared to ShoreFor, the CGP-evolved model is able to produce more accurate predictions of the seasonal to long-term trends in shoreline behavior, while ShoreFor produces better forecasts of the shorter-term shifts in shoreline positions at some of the sites. In general, the model produced by CGP achieves higher Mielke skill score over the training period over most sites, while achieving similar or higher test skill over almost all sites.
Overall, these results showcase CGP's ability to evolve an existing shoreline prediction model to improve its prediction performance over multiple sites. They also demonstrate the usefulness of CGP as a research tool to gain insight into the behaviors of shorelines at different areas of the world.

3.4 Evolved Model

Figure 4 presents the graph representation of the generalist model produced by CGP. The model can also be represented by a system of a equations as presented in Equation 2.

$$\frac{dx}{dt} = \begin{cases} \frac{1}{2}\bar{P}^{0.5}\frac{d}{dt}\sqrt{\frac{\phi^2}{2} + \frac{1}{4}(R-\Omega)^2 + S^2}, & \text{if } S \geq P^{0.5} + \Omega \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

We first note that the evolved model differs largely from the original ShoreFor model. While the fall-velocity Ω is used, there is no calculation of an equilibrium term Ω_{eq}, and the wave energy flux P is used directly to calculate $\frac{dx}{dt}$, as opposed to calculating accretive and erosive wave forcing terms F. The combination of information not used in ShoreFor, sea level anomaly S and river discharge R, instead determines the influence of wave power on shoreline change. While it is notable that wave power is the primary driver in both the evolved model and in ShoreFor, the evolved model offers a different perspective on the relationship between sea level, river discharge, wave power, and shoreline change.

4 Discussion & Conclusion

In this work, we have presented our work on the use of CGP and NSGA-II in order to evolve a pre-established shoreline forecasting model, ShoreFor. CGP was used in order to encode the ShoreFor system of equations into a format that can be evolved using evolutionary algorithms. During evolution, NSGA-II is employed in order to maintain a pareto-front of optimal solutions according to their performances at five different coastal points from around the globe. A generalist model was selected according to its average forecast skill score over the five different objectives, the CGP-produced model achieved superior predictive skill than that of ShoreFor over four of the five sites used in this work.
Future directions of this work would include the expansion of the training and testing datasets to cover more areas around the globe. Using satellite imagery, shoreline data could be obtained from around the world. Coupled with global atmospheric and wave models for the input parameters, similar methodologies could be used to evolve a global shoreline forecasting model. Furthermore, the incusion of a wider range of shorelines and coastal environments in the training data could expose the method to different areas in the search space that could bring the algorithm closer towards finding a globally-applicable model. In addition, the use of a larger training dataset would allow for further development of the algorithm using a noisy fitness evaluation scheme [38, 39, 40, 41, 42] which should aid in reducing overfitting and maximizing test performance.

Overall, the results presented in this work are a strong motivation for further study on the use of genetic programming and multi-objective genetic algorithms in shoreline forecasting studies due to the wide potential applicability of the evolved models and their interpretability as ordinary systems of equations.

REFERENCES

[1] Jennifer Montaño, Giovanni Coco, Jose AA Antolínez, Tomas Beuzen, Karin R Bryan, Laura Cagigal, Bruno Castelle, Mark A Davidson, Evan B Goldstein, Raimundo Ibaceta, et al. Blind testing of shoreline evolution models. *Scientific reports*, 10(1):1–10, 2020.

[2] John A Church and Neil J White. A 20th century acceleration in global sea-level rise. *Geophysical research letters*, 33(1), 2006.

[3] Robert J Nicholls, Susan E Hanson, Jason A Lowe, Richard A Warrick, Xianfu Lu, and Antony J Long. Sea-level scenarios for evaluating coastal impacts. *Wiley Interdisciplinary Reviews: Climate Change*, 5(1):129–150, 2014.

[4] Borja G Reguero, Iñigo J Losada, and Fernando J Méndez. A recent increase in global wave power as a consequence of oceanic warming. *Nature communications*, 10(1):1–14, 2019.

[5] IR Warren and H_K_ Bach. Mike 21: a modelling system for estuaries, coastal waters and seas. *Environmental Software*, 7(4):229–240, 1992.

[6] Giles R Lesser, JA v Roelvink, JA Th M van Kester, and GS Stelling. Development and validation of a three-dimensional morphological model. *Coastal engineering*, 51(8-9):883–915, 2004.

[7] John C Warner, Christopher R Sherwood, Richard P Signell, Courtney K Harris, and Hernan G Arango. Development of a three-dimensional, regional, coupled wave, current, and sediment-transport model. *Computers & geosciences*, 34(10):1284–1306, 2008.

[8] Patrick Marchesiello, Julien Chauchat, Hassan Shafiei, Rafael Almar, Rachid Benshila, Franck Dumas, and Laurent Debreu. 3d wave-resolving simulation of sandbar migration. *Ocean Modelling*, page 102127, 2022.

[9] Mark Davidson. Forecasting coastal evolution on time-scales of days to decades. *Coastal Engineering*, 168:103928, 2021.

[10] Evan B Goldstein, Giovanni Coco, and Nathaniel G Plant. A review of machine learning applications to coastal sediment transport and morphodynamics. *Earth-science reviews*, 194: 97–108, 2019.

[11] GS Dwarakish, Rakshith Shetty, and Usha Natesan. Review on applications of neural network in coastal engineering. *Artificial Intelligent Systems and Machine Learning*, 5(7):324–331, 2013.

[12] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.

[13] Saeed Zeinali, Maryam Dehghani, and Nasser Talebbeydokhti. Artificial neural network for the prediction of shoreline changes in narrabeen, australia. *Applied Ocean Research*, 107: 102362, 2021.

[14] Floris Calkoen, Arjen Luijendijk, Cristian Rodriguez Rivero, Etienne Kras, and Fedor Baart. Traditional vs. machine-learning methods for forecasting sandy shoreline evolution using historic satellite-derived shorelines. *Remote Sensing*, 13(5):934, 2021.

[15] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, 32(11):4793–4813, 2020.

[16] Plamen P Angelov, Eduardo A Soares, Richard Jiang, Nicholas I Arnold, and Peter M Atkinson. Explainable artificial intelligence: an analytical review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(5):e1424, 2021.

[17] William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabricio Olivetti de Franca, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H Moore. Contemporary symbolic regression methods and their relative performance. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[18] Thomas Uriot, Marco Virgolin, Tanja Alderliesten, and Peter AN Bosman. On genetic programming representations and fitness functions for interpretable dimensionality reduction. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 458–466, 2022.

[19] Kyle Cranmer and R Sean Bowman. Physicsgp: A genetic programming approach to event selection. *Computer Physics Communications*, 167(3):165–176, 2005.

[20] JM Link, PM Yager, JC Anjos, I Bediaga, C Castromonte, C Göbel, AA Machado, J Magnin, A Massafferri, JM De Miranda, et al. Application of genetic programming to high energy physics event selection. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 551(2-3):504–527, 2005.

[21] Lee Spector, Howard Barnum, Herbert J Bernstein, and Nikhil Swamy. Quantum computing applications of genetic programming. *Advances in genetic programming*, 3:135–160, 1999.

[22] Lee Spector and Jon Klein. Machine invention of quantum computing circuits by means of genetic programming. *AI EDAM*, 22(3):275–283, 2008.

[23] Evan B Goldstein, Giovanni Coco, and A Brad Murray. Prediction of wave ripple characteristics using genetic programming. *Continental Shelf Research*, 71:1–15, 2013.

[24] A Makkeasorn, Ni-Bin Chang, and Xiaobing Zhou. Short-term streamflow forecasting with global climate change implications–a comparative study between genetic programming and neural network models. *Journal of Hydrology*, 352(3-4):336–354, 2008.

[25] Julian F Miller. Cartesian genetic programming. In *Cartesian Genetic Programming*, pages 17–34. Springer, 2011.

[26] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6 (2):182–197, 2002.

[27] MA Davidson, KD Splinter, and IL Turner. A simple equilibrium model for predicting shoreline change. *Coastal Engineering*, 73:191–202, 2013.

[28] Lynn D Wright, Andrew D Short, and MO Green. Short-term changes in the morphodynamic states of beaches and surf zones: an empirical predictive model. *Marine geology*, 62(3-4): 339–364, 1985.

[29] Rob Schepper, Rafael Almar, Erwin Bergsma, Sierd de Vries, Ad Reniers, Mark Davidson, and Kristen Splinter. Modelling cross-shore shoreline change on multiple timescales and their interactions. *Journal of Marine Science and Engineering*, 9(6):582, 2021.

[30] Yen Hai Tran and Eric Barthélemy. Combined longshore and cross-shore shoreline model for closed embayed beaches. *Coastal Engineering*, 158:103692, 2020.

[31] Yen Hai Tran, Patrick Marchesiello, Rafael Almar, Duc Tuan Ho, Thong Nguyen, Duong Hai Thuan, and Eric Barthélemy. Combined longshore and cross-shore modeling for low-energy embayed sandy beaches. *Journal of Marine Science and Engineering*, 9(9):979, 2021.

[32] Milan Češka, Jiří Matyáš, Vojtech Mrazek, Lukas Sekanina, Zdenek Vasicek, and Tomas Vojnar. Approximating complex arithmetic circuits with formal error guarantees: 32-bit multipliers accomplished. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 416–423. IEEE, 2017.

[33] Lukas Sekanina, Simon L Harding, Wolfgang Banzhaf, and Taras Kowaliw. Image processing and cgp. In *Cartesian genetic programming*, pages 181–215. Springer, 2011.

[34] Simon Harding, Jürgen Leitner, and Juergen Schmidhuber. Cartesian genetic programming for image processing. In *Genetic programming theory and practice X*, pages 31–44. Springer, 2013.

[35] Dennis G Wilson, Sylvain Cussat-Blanc, Hervé Luga, and Julian F Miller. Evolving simple programs for playing atari games. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 229–236, 2018.

[36] Julian Francis Miller. Cartesian genetic programming: its status and future. *Genetic Programming and Evolvable Machines*, 21(1):129–168, 2020.

[37] Gregory Duveiller, Dominique Fasbender, and Michele Meroni. Revisiting the concept of a symmetric index of agreement for continuous datasets. *Scientific reports*, 6(1):1–14, 2016.

[38] W Langdon. Minimising testing in genetic programming. *RN*, 11(10):1, 2011.

[39] Yi Liu and Taghi Khoshgoftaar. Reducing overfitting in genetic programming models for software quality classification. In *Eighth IEEE International Symposium on High Assurance Systems Engineering, 2004. Proceedings.*, pages 56–65. IEEE Computer Society, 2004.

[40] Ivo Gonçalves, Sara Silva, Joana B Melo, and João Carreiras. Random sampling technique for overfitting control in genetic programming. In *European Conference on Genetic Programming*, pages 218–229. Springer, 2012.

[41] Yasuhito Sano and Hajime Kita. Optimization of noisy fitness functions by means of genetic algorithms using history of search with test of estimation. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02Th8600)*, volume 1, pages 360–365. IEEE, 2002.

[42] Magnus Rattray and Jon Shapiro. Noisy fitness evaluation in genetic algorithms and the dynamics of learning. 1998.

APPENDIX A SHOREFOR IN CGP

We encode the ShoreFor model as a CGP individual in order to improve it using evolution. The first population of models is created by randomly mutating the ShoreFor individual, and random genes are used to fill in the inactive nodes. The encoding also informs the inputs available and the function set, as we use functions during search which are necessary to encode the ShoreFor individual.

We first implement equation 1, the Ω_eq time series, by calculating the weight vector $W = 10^{-i/\phi}$, which is computed such that the weighting factor decreases per day i over the number of days ϕ. In order to encode the computation of this weight vector, two inputs and five different functions are required; the inputs are the calibrated ϕ constant and the value of 2ϕ, which specifies the size of the moving window. Given these inputs, we decompose the calculation of W as follows. First, a vector of length 2ϕ with values ranging from 1 to 2ϕ is computed using the $irange$ function. This vector is then simply flipped using the $reverse$ function, to represent the number of days back in history each point in the time series represents. At this point, the vector of i in $10^{-i/\phi}$ is computed. Then, this vector is negated using the $negate$ function and divided by the input constant phi using div, resulting in a vector of values representing $-i/\phi$. This vector is passed to the $tpow$ function ($tpow(x) = 10^x$), obtaining $W = 10^{-i/\phi}$. In order to compute the moving average over the full time series, we make use of the convolution function. We therefore modify the computation of Ω_{eq} as follows: $\Omega_{eq} = conv(\Omega, \frac{10^{-i/\phi}}{\sum_{i=1}^{2\phi} 10^{-i/\phi}})$. The next step in the computational graph is to divide the weight vector by the sum of the vector itself to be used as the convolution filter. The graph-form of Equation 1 is shown in Figure 2(left).

The final set of inputs used in our CGP-ShoreFor implementation is described in Table 2. We note that not all inputs included in our implementation are used by the ShoreFor model. These additional inputs are included so that evolution can integrate them into the evolved models.

Input	Description
Ω	Dimensionless fall-velocity time series
P	Wave power time series
ϕ	Pre-calibrated number of days used for the initial ShoreFor model
2ϕ	Used to indicate the size of the weight vector in Equation 1
D*	Wave direction time series
$H_{s,b}$*	Peak breaking wave height
T_p*	Peak wave period
S*	Sea level anomaly
R*	Regional river discharge

Table 2: Inputs to the CGP-ShoreFor model. *Additional inputs that are not used by ShoreFor.

Generally speaking, implementations of CGP require that all input and computed variables are bound to a range of -1 to 1 in order to prevent various computational issues such as the existance of NaN's or infinities in the computational graph. However, this requirement is difficult to achieve in the case of GI of a physical system of equations due to the lack of true maxima for each input and the use of unbounded functions in the original model. Therefore, we instead choose to handle out-of-bounds computation by penalizing all such individuals by assigning them a fitness value of negative infinity, essentially discarding them from future generations.

After encoding the ShoreFor system of equations as a single CGP genome, the ShoreFor individual can be represented as a graph structure as shown in Figure 2(right).

Currently, the implementation of ShoreFor as a CGP individual assumes that the equations to calculate P and Ω are physical facts and are therefore not included in the evolvable CGP-ShoreFor implementation, but rather passed as pre-calculated time series.

APPENDIX B CGP

The definition of CGP in this work uses standard uniform mutations where functional genes and connection genes are modified randomly. However, the following mutation-level constraints are also applied in order to discard invalid individuals before evaluation: 1) We discard all mutated graphs with direct input-output connections. 2) We ensure that for the same set of random inputs, the outputs produced by parent graph and the mutated graph are different in order to minimze the chances of having behaviorally identical individuals in the population. 3) Since we use a mixed-type

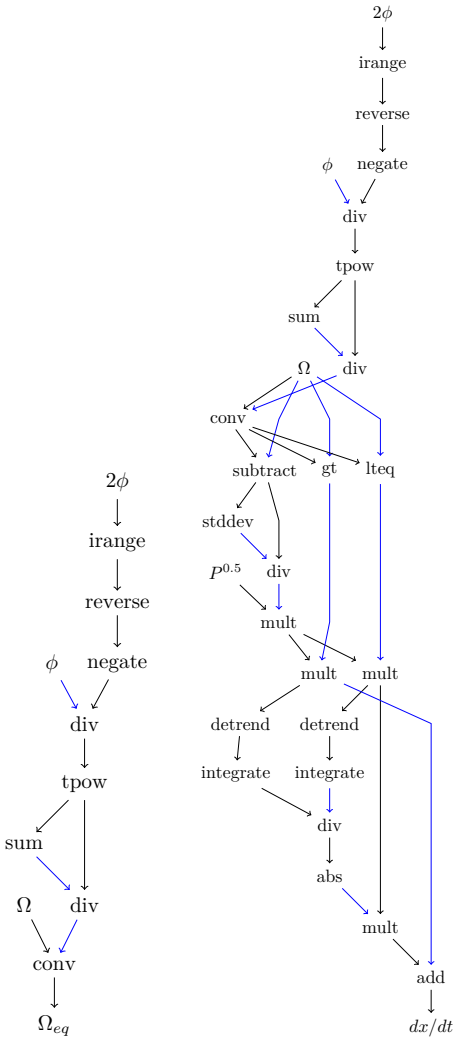

Figure 2: Graph representations of the Ωeq equation (left) and the full ShoreFor system of equations (right).

version of CGP where both scalar and vector values exist within the computational graph, we add a constraint that discards any mutated graph that outputs scalar values. 4) Finally, we ensure that the size of the output vector is equal to the size of the input time series.

APPENDIX C NSGA-II

NSGA-II is a well-known and widely used multi-objective genetic algorithm that was proposed in [26]. It makes use of the concept of pareto dominance in order to split a population of models into different performance-based ranks. A crowding distance measure is also used in order to maintain the diversity of population during evolution. At generation G_t, parents (P_t) are selected from the current population using tournament selection and are mutated in order to generate a population of offspring models Q_t. Individuals in Q_t are evaluated according to the user-defined fitness function, then a combined population R_t is created by merging both P_t and Q_t. R_t is then sorted according based on pareto dominance, as well as the crowding distance in lower-ranks, and the N top-ranking individuals are chosen as the upcoming population. The algorithm is run in a loop until a certain threshold is reached, such as the number of evaluations executed or a predefined fitness threshold. Due to the elitist nature of NSGA-II, top-ranking models are guaranteed to be conserved through the different generations until they are replaced by higher-ranking models.

APPENDIX D EVOLUTIONARY CONFIGURATION

In this work, NSGA-II is used to evolve the CGP-encoded individuals. The algorithm is configured according to Table 3 and run for 50 thousand generations. This setup was executed for 50 runs using different random seeds. All runs were found to converge very early on during evolution (within hundreds of generations). After each run, 200 different CGP individuals are recorded representing the final generation from that run. The final generations from all runs are grouped into a single merged population of 10000 individuals and evaluated using the calibration and forecast datasets.

Parameter	Value
Population size	200
N offsprings	200
Mutation rate	0.1
Output mutation rate	0.3
Rows	1
Columns	50

Table 3: Evolutionary configuration used throughout this work.

Tables 4 and 5 document the different functions used in this work, representing the function set used by the genetic algorithm during evolution.

Function	Operation
abs	$\lvert x \rvert$
sqrt	$\sqrt{x}$
sin	$\sin x$
cos	$\cos x$
negate	$-x$
tpow	10^x
nop	x
add	$x + y$
subtract	$x - y$
mult	$x * y$
div	$x \div y$
pow	x^y
sqrt_xy	$\sqrt{x^2 + y^2}$
lteq	$x <= y$
gt	$x > y$

Table 4: The set of scalar operations included in the GA's function set.

Function	Operation
diff	$x_i = x_i - x_{i-1}$
sum	$\sum_{i=1}^{n} x$
stddev	σx
detrend	$\langle x \rangle$
integrate*	$\int_a^b f(x)dx \approx \sum_{i=1}^{n} \frac{f(x_{i-1})+f(x_i)}{2} \Delta x_i$
mean	$\frac{1}{n} \sum_{i=1}^{n} x$
reverse	$reverse(x)$
irange	$[1, 2, 3...n], n = length(x)$
conv	$conv(x, y)$

Table 5: The set of vector operations included in the GA's function set. * Assuming a constant time step in the time series.

APPENDIX E MODEL COMPARISON

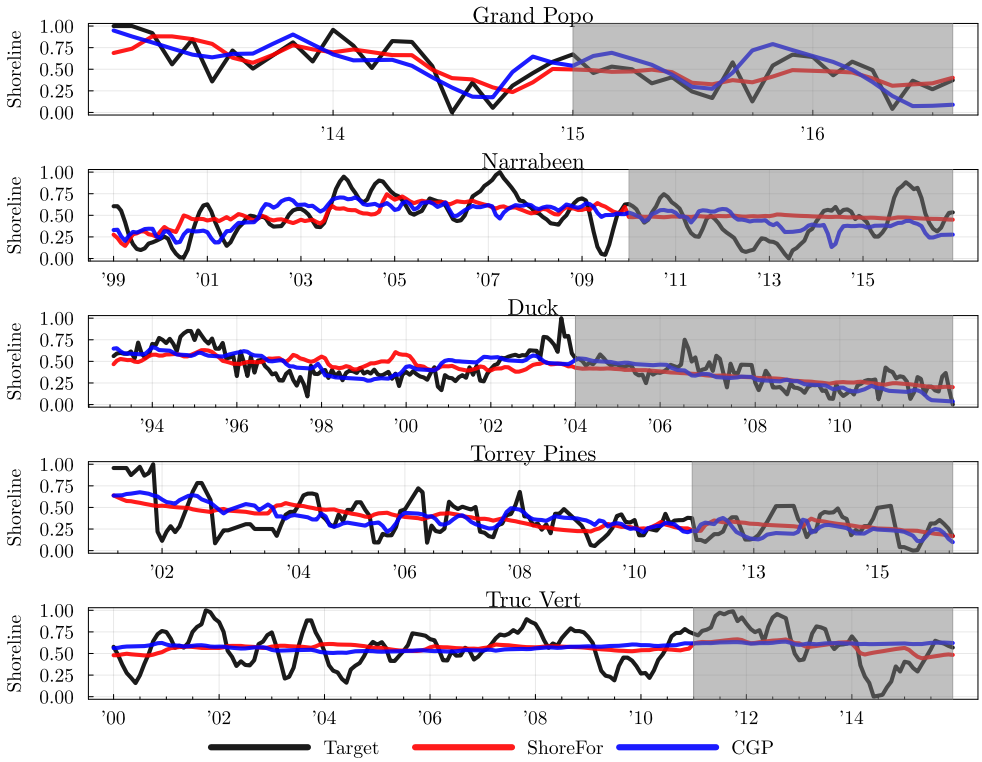

Figure 3: Performances of ShoreFor and CGP-evolved model over the training and test periods at the five coastal sites. *Shaded areas correspond to the test period at each site.

In Figure 3, we display the prediction of ShoreFor and the CGP-evolved model on the five sites. At Grand Popo, the CGP-evolved model demonstrates higher predictive skill compared to ShoreFor in predicting longer term trends in shoreline behavior, while being less accurate in predicting shorter-term variations in shoreline position, resulting in higher Mielke skill over the training period, and an identical score over the test period. At Narrabeen, the generalist model achieves only a slight improvement over the baseline ShoreFor model. Over Duck, while both models seem unable to accurately predict the shorter-term variations in shoreline position, the CGP-produced model demonstrates higher skill in predicting the longer-term trends in shoreline behavior, allowing it to have a higher Mielke skill over both the training and test periods. At Torrey Pines, the CGP model is able to better predict the strong seasonal cycle compared to ShoreFor, achieving higher training skill and slightly improved test skill. Finally, both models fail to produce a reliable forecast of the shoreline position at Truc Vert as shown in Figure 3. Both models achieve poor Mielke skill over the training period, while ShoreFor achieves a higher skill over the test period.

APPENDIX F EVOLVED GRAPH

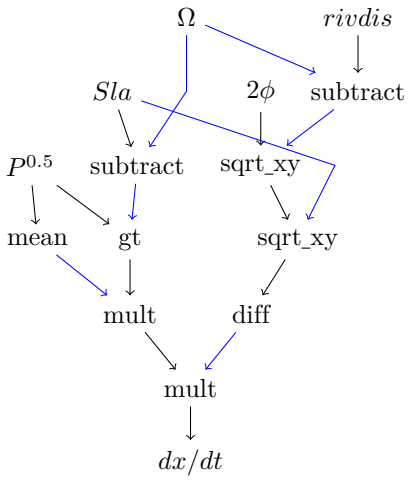

Figure 4: The generalist model produced by CGP and NSGA-II using ShoreFor as the starting point for evolution.