ENVIRONMENTAL MODEL ACCURACY IMPROVEMENT FRAMEWORK USING STATISTICAL TECHNIQUES AND A NOVEL TRAINING APPROACH

by

Rakesh Matta

May, 2020

Director of Thesis: Rui Wu, PhD

Major Department: Computer Science

It is challenging to predict environmental behaviors because of extreme events, such as heatwaves, typhoons, droughts, tsunamis, torrential downpour, wind ramps, or hurricanes. In this thesis, we proposed a novel framework to improve environmental model accuracy with a novel training approach. Extreme event detection algorithms are surveyed, selected, and applied in our proposed framework. The application of statistics in extreme events detection is quite diverse and leads to diverse formulations, which need to be designed for a specific problem. Each formula needs to be tailored specifically to work with the available data in the given situation. This diversity is one of the driving forces of this research towards identifying the most common mixture of components utilized in the analysis of extreme events detection. Besides the extreme event detection algorithm, we also integrated the sliding window approach to see how well our models predict future events. To test the proposed framework, we collected coastal data from various sources and obtained the results; we improved the predictive accuracy of various machine learning models by 20% to 25% increase in R^2 value using our approach. Apart from that, we organized the discussion along with different extreme event detection types, presented a few outlier definitions, and briefly introduced their techniques. We also summarized the statistical methods involved in the detection of environmental extremes, such as wind ramps and climatic events.

ENVIRONMENTAL MODEL ACCURACY IMPROVEMENT FRAMEWORK USING STATISTICAL TECHNIQUES AND A NOVEL TRAINING APPROACH

A Thesis

Presented to The Faculty of the Department of Computer Science

East Carolina University

In Partial Fulfillment of the Requirements for the Degree

Master of Science in Computer Science

by

Rakesh Matta

May, 2020

ENVIRONMENTAL MODEL ACCURACY IMPROVEMENT FRAMEWORK USING STATISTICAL TECHNIQUES AND A NOVEL TRAINING APPROACH

by

Rakesh Matta

APPROVED BY:

DIRECTOR OF THESIS:

Rui Wu, PhD

COMMITTEE MEMBER:

Nic Herndon, PhD

COMMITTEE MEMBER:

Randall Etheridge, PhD

CHAIR OF THE DEPARTMENT OF COMPUTER SCIENCE:

Venkat N. Gudivada, PhD

DEAN OF THE GRADUATE SCHOOL:

Paul J. Gemperline, PhD

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Most of the existing environmental methods deal with spatial and temporal variables on hard and subjective thresholds. To eliminate the idea of subjective threshold, DSOPT and scikit-extremes methods are used in our framework to compute the threshold values. We proposed a novel approach of training separate machine learning models for normal data and abnormal data to increase the prediction accuracy of the trained models. A sliding window approach can be used to consider the mid-term seasonality and to identify the local undetectable outliers. We also used it to see how well our machine learning models perform in predicting the future events. Handling incomplete, missing, and delayed information is a difficult task. Noise, skewed distributions, and redundant values also hinder the progress. In traditional offline analysis, data preprocessing is usually done manually by a human expert before modeling. In the streaming scenario, manual processing is not feasible, as new data arrives continuously. Streaming data needs fully automated preprocessing methods, that can optimize the parameters and operate autonomously. Since new data arrives at every time instant, the scale of the data is immense. So preprocessing techniques like data cleansing, formatting, feature selection are required to solve the processing and resource-constraint challenges like energy, memory, computational capacity, and communication bandwidth usage [2, 3]. Our novel approach of splitting the data into

normal data and abnormal data using the threshold values is a key factor in obtaining a better accuracy in predicting future events.

A time-series outlier is an observation that significantly differs from other observations of the same feature. When a time series data is plotted, outliers are usually the unexpected spikes or dips of observations at a given point in time. An outlier may exist due to a rare event, incorrect values as a result of errors or breakdowns, or corrupt recording practices [4]. Outliers can be identified using a model by learning all the time series sequences in the database. The outlier score can be computed for each sequence using a scoring function based on the model, which can be either supervised or unsupervised depending on the availability of training data [5]. We can also detect an outlier based on frequent patterns. A data instance is likely to be an outlier if it does not contain frequent patterns [6]. We can identify outliers by computing the outlier score, which can be the deviation of the actual value from the predicted value or the distance to the centroid of the closest data cluster. In [7], they considered the outlier score to be proportional to the density of its k-nearest neighbors over the local density of the data point [8]. In the case of the sliding window approach, the time series sequence can be broken into multiple overlapping windows of fixed length, an outlier score is computed for each window and aggregated later to identify anomalies. The sliding window approach can also be used to compute the model performance in predicting future events. In the local outlier factor (LOF) method, anomalous data points are identified by measuring the local deviation of a given data point with respect to its neighbors. A better approach is proposed in [9], where the incremental LOF algorithm computes LOF value for each data record inserted into the data set and instantly determines whether the inserted data record is an outlier. In [10], they proposed a solution for the distributed local outlier factor approach, which is inherently parallel and deployable on virtually any distributed infrastructure. They

also designed a multi-step pipeline framework called distributed local outlier factor, which leverages the invariant observation and computes LOF scores in a highly distributed fashion. For a survey of outlier detection methods, see [1]. Apart from that, there are several outlier detection methods to identify the outliers based on various scenarios and available data. Both the climatic and the wind ramps events deal with the temporal data. Modeling temporal data is a challenging task due to the dynamic nature and complex evolutionary patterns in the data. [1].

In chapter 2, we present all the background research work related to anomaly detection and various statistical models. It includes the taxonomy of our extreme environmental events and a brief introduction to climatic and wind ramp events along with their statistical methods. These statistical methods are used to obtain a threshold value and to identify the outliers. We discuss all the statistical methods and machine learning models used in this research project. In chapter 3, we detail our proposed framework and the workflow, including data preprocessing techniques like data cleansing, formatting, feature engineering, and the training process. We also discuss on how we planned to split the data into training and test data, the idea of eliminating subjective threshold by computing a threshold value using DSOPT and scikit-extremes to split the training and test data into normal and abnormal data, training separate machine models for normal data and abnormal data. In chapter 4, we discuss the coastal data sets used to validate our model, how we perform our workflow on those data sets along with their performance and improved results. Finally, in chapter 5, we conclude and summarize our thesis in brief.

Chapter 2

Related work

2.1 Taxonomy of extreme weather events

Extreme weather events have generated interest world over, because of their potential for high impacts on ecological, technical, and social systems. There arises a need for an organized and detailed study of the work done in extreme events detection using various statistical approaches. Out of all the extreme environmental events, we briefly surveyed and illustrated the statistical methods for detecting climatic and wind ramp events, as shown in figure 2.1.

Climatic events are occasional variations producing extreme values of climate indicators, such as temperature and precipitation. Climate change can potentially change the intensity, frequency, timing, and duration of these events. Detection of climatic extremes mostly deals with extreme value theory involving Peak Over Threshold (POT), Generalized Pareto Distribution (GPD), and Maximum Likelihood Estimation (MLE) approaches, discussed in section 2.3. In contrast, wind ramp events are unforeseen increases or decreases in wind speed or detection along a short time. Changing wind turbines design based on wind ramp detection can increase the electricity generation rate. Wind ramp domain shall specifically cover the detection of wind ramps and increase wind farm's efficiency. Sliding window and dynamic programming approaches are often utilized by several researchers to detect wind ramps efficiently. For wind

ramp events, we summarized the data preprocessing and feature extraction methods for analyzing the data to obtain useful patterns. We also illustrated various ramp detection techniques to categorize future ramp events based on the available data.

2.1.1 Climatic events

Over the years, changes in the variability of climatic extremes had more impacts compared to the changes in the mean climate [11]. Simulations of various models are analyzed and compared with each other, and those simulations can have statistically significant links. For example, precipitation quantity can vary depending on the frequency or intensity of each precipitation event, or a combination of both factors. The intensity of precipitation refers to the amount of precipitation associated with specific quantities of the precipitation distribution. It is possible to estimate the proportion of any trend in total precipitation that is attributable to changes in frequency versus changes in precipitation intensity. Precipitation patterns may also be derived for particular quantities. Few statistical techniques are used more frequently than others, such as the maximum daily quantity of precipitation. This statistic is obtained by identifying the maximum quantities of precipitation per month throughout all the years of recorded data, then by computing the pattern through the obtained values. Alternatively, the ratio of precipitation quantity in a particular area may be matched up to the mean precipitation quantity in the entire area to see the variations of precipitation. This statistic also reveals variations in the precipitation distribution that are independent of the variations in the mean quantities [12, 13, 14, 1].

2.1.2 Wind ramps

Extreme fluctuations in winds can result in damages to wind turbines. Available historical data are used to train various models to classify future ramp events using

various statistical approaches, discussed in 2.3.11, like trend fitting, dynamic programming, sliding window, mutual information, etc. Features are extracted with the help of these methods to categorize the wind ramp events based on a set of predefined thresholds [15]. In [16], Raffi Sevlian and Ram Rajagopal described an optimal ramp detection technique for identifying wind ramp events of varying lengths under any arbitrary rule set for large time series. A dynamic programming recursion with trend fitting is used to segment wind power data to ramp and non-ramp events optimally. An optimized swinging door algorithm (OpSDA) by utilizing the swinging door algorithm and a dynamic programming algorithm to handle the wind power bumps were developed in [17, 18]. The data is segregated through a piecewise linear approximation, and the segments are optimized by merging adjacent segments with the same ramp changing direction. This optimized swinging door algorithm (OpSDA) was adopted in [19] and extended to detect wind ramps events, by merging bumps having a different changing direction, into adjacent ramping segments to improve the performance of the OpSDA method [1].

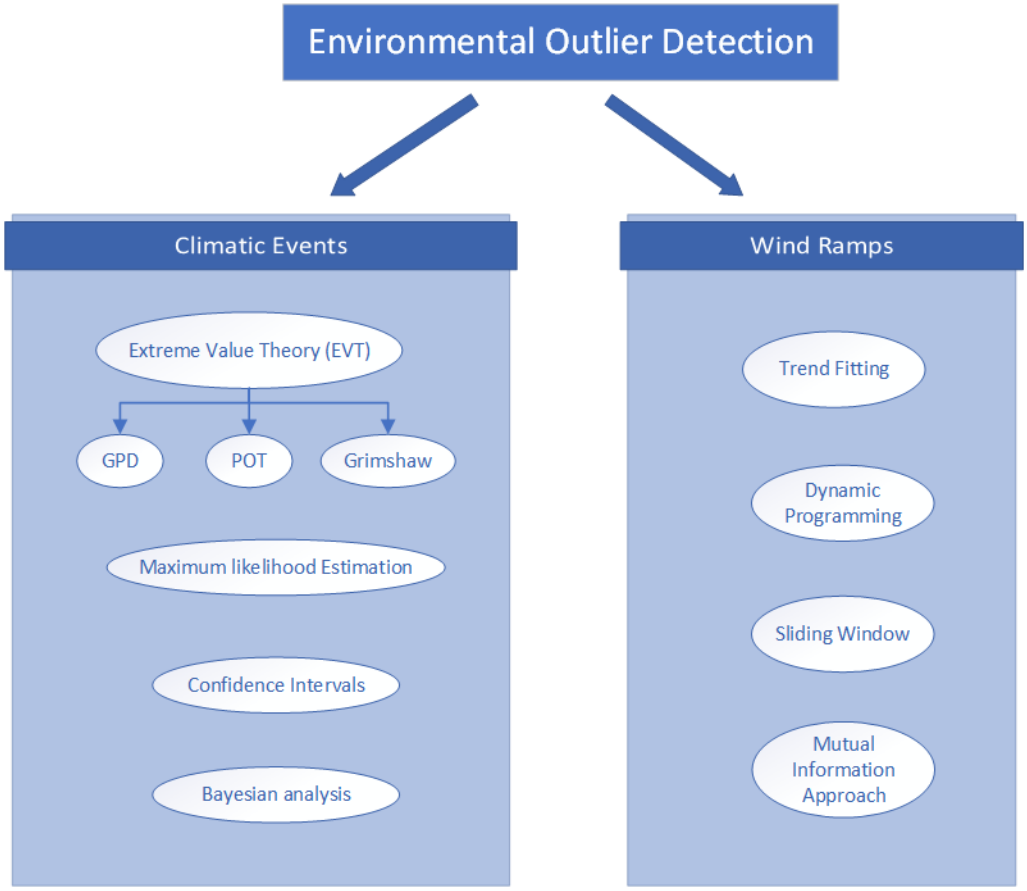

Figure 2.1: Commonly used statistical methods related to climatic and wind ramp events, from [1]

2.2 Post-processor framework to improve the accuracy of hydrologic models

It is often challenging to develop an accurate hydrologic model due to the time-consuming model calibration procedure and the fluctuation of hydrologic data. In [20], the authors introduced some techniques to improve the accuracy and flexibility of hydrologic models in terms of model simulation and development using machine-learning models (like support vector regression, gradient boosted tree and random forest) as

post processors. The workload is reduced significantly by creating an accurate hydrologic model without the calibration step. A moving window-based machine-learning approach is proposed by them to enhance the machine-learning error predictions.

Based on their findings, the errors of hydrologic models are correlated with model inputs. So, they proposed this framework by leveraging this correlation to improve the accuracy of a hydrologic model. The key idea was to predict the differences between the observed values and the hydrologic model predictions, the errors of the model by using machine-learning techniques. A machine-learning-based post-processor is introduced, which can capture and characterize model errors to improve hydrologic model predictions. It significantly simplifies the parameter tuning processes by learning and calibrating the modeling error using machine-learning techniques. A moving window-based approach is proposed to tackle the fluctuating issue, which identifies the local stationarity regions using a stationarity measure. Their idea was to first find all possible window sizes by using data autocorrelation and then select the best window size, which contains stationary data. The stationarity measure is proposed to calculate the data stationarity within a window. The distribution of hydrologic data changes over time, and the data exhibit seasonality. This approach can characterize the time-varying relationship between model inputs and model output errors [20]. The idea of sliding window in our framework for evaluating the models ability to predict future events is gathered from this work.

2.3 Statistical methods for extreme events detection

2.3.1 Extreme Value Theory (EVT)

Extreme value theory is a powerful statistical tool developed to study the laws of extreme events i.e., Extreme Value Distributions (EVD). Using these EVDs, we can

analyze the input time series using normal distribution. When the data doesn't fit a normal distribution, we can use log-normal distribution, which transforms not-normal distribution into normal [21]. By fitting an EVD to the unknown distribution, it is possible to evaluate the probability of potential extreme events. Extreme events have the same kind of distributions, based on the results from Fisher, Tippett [22] and Gnedenko [23]. These extreme value distributions have the following form:

$$G_\gamma : x \mapsto exp\left(-(1+\gamma x)^{-\frac{1}{\gamma}}\right), \gamma \in R, (1+\gamma x) \geq 0 \tag{2.1}$$

Extremes (x values) of common standard distributions follow the above distribution, and the extreme value index, γ, depends on it. It may not be reasonable to apply a method directly to detect extreme climatic events if the method assumes data follow a specific distribution because climatic data may not follow any data distribution. The extreme value theory, through the POT approach, introduced in subsection 2.3.2, gives us a way to estimate threshold without any strong assumption and precise knowledge about the distribution. We have to detect abnormal events in a stream in a blind way without any knowledge about the distribution [24, 25, 1].

2.3.2 Peak Over Threshold (POT)

Climatic data is usually temporal data. POT is one of the efficient methods to analyze peak values in the time series. Peak values in the time series can be analyzed using the POT method. Initially, we set a subjective threshold to obtain all the data points which exceed the threshold (the peaks) and then fit a Generalized Pareto Distribution, discussed in subsection 2.3.5, to those peaks. The estimation of GPD parameters, i.e., shape and scale parameters, can be done using various methods, among which maximum likelihood function, discussed in subsection 2.3.6, is widely

preferred, especially when the data sample is large [21].

Many applications rely on high throughput streaming data. In [25], the authors proposed two streaming algorithms: streaming peak over threshold (SPOT) for stationary cases and SPOT with drift, discussed in subsection 2.3.3, which consider the drift component. Firstly, a POT estimation is performed on the first few observations, and a threshold is initialized. Then for all the following observed values, they flagged all the abnormal values and updated the threshold. They built a streaming outlier detector, which uses the next observations to both detect the outliers and refine the threshold value. SPOT assumes that the distribution does not change over time, but it might be conservative. They proposed DSPOT, where SPOT runs on the relative values by considering the variable changes and local behavior at every moment.

Parameter estimates, scale (σ) and shape (γ) parameters, can be computed through:

$$z_q = t + \frac{\hat{\sigma}}{\hat{\gamma}} \left(\left(\frac{qn}{N_t} \right)^{-\hat{\gamma}} - 1 \right) \tag{2.2}$$

Where z_q is updated threshold, t is threshold value, q is the desired probability, n is the total number of observations, N_t is the number of peaks i.e., observations greater than threshold values. Estimation of parameters are done using appraches like maximum likelihood estimation, grimshaw, etc [1].

2.3.3 Anomaly detection in streams with EVT

The extreme value theory, through the POT approach, gives us a way to estimate z_q such that $P(X > z_q) < q$ without any strong assumption on the distribution of X and any explicit knowledge about its distribution.

A streaming outlier detector was built to identify all the outliers dynamically in a streaming scenario. First, the initialization step is introduced, which computes a

threshold z_q from n observations $(X_1, X_2....., X_n)$ with a risk q. Then, two streaming algorithms are introduced, which updates z_q with the incoming data and use it as a decision bound. They proposed SPOT, which works in stationary cases and DSPOT, which takes into account a drift component. The idea is to set a high threshold t, retrieve the peaks (the excesses over t) and fit a GPD to them. So that the distribution of the extreme values is inferred, and the threshold value z_q is computed. The only necessary condition is to ensure that t is lower than z_q, meaning that the probability associated with t must be lower than 1 - q. The set Y_t is the peaks set where the observed excesses over t are stored. The streaming anomaly detector uses the next observations to both detect anomalies and refine the anomaly threshold z_q [25].

Streaming Peak Over Threshold (SPOT)

The way how the POT estimate is built is stream-ready. As it is not necessary to store the whole time series (only the peaks are required), this approach requires low memory. However, the stream must contain values from the same distribution, so this distribution cannot be time-dependent. In the case of time-dependency, the algorithm can be adapted to drifting cases. The principle of the SPOT algorithm is to detect abnormal events in streams without knowledge about the distribution. Firstly, a POT estimate is performed on the n first values ($n \sim 1000$), and an initial threshold of z_q (initialization) is computed. Then for all the following observed values, the extreme events are flagged, or the threshold is updated. If a value exceeds the threshold z_q, then it is considered as abnormal (retrieved the anomaly in a list). The anomalies are not taken into account for the model update. In other cases, either X_i is greater than the initial threshold (peak case) or is a common value (normal case). In the peak case, the values are added to the peaks set, and then the threshold z_q is updated. In this algorithm, a maximum number of threshold updates are performed, but it is

possible to do it off-line at a fixed time interval [25].

Streaming Peak Over Threshold with Drift (DSPOT)

SPOT assumes that the distribution of the X_i does not change over time, but it might be restrictive. For instance, a mid-term seasonality cannot be taken into account, making local peaks undetectable. This issue is overcome by modeling an average local behavior and applying SPOT on relative gaps. DSPOT approach is proposed, which makes SPOT run not on the absolute values X_i but the relative ones. Variable change is used, $X'_i = X_i - M_i$, where M_i models the local behavior at time i. A moving average approach is implemented where $M_i = \left(\frac{1}{d}\right) \sum_{k=1}^{d} X^*_{i-k}$ with, $X^*_{i-1}, X^*_{i-2}, ..., X^*_{i-d}$, the last d normal observations (where d is a window parameter). This variant uses an additional parameter of d, which can be viewed as the window size. The distinctive features of this window are: it might be non-continuous, and it does not contain abnormal values [25].

2.3.4 scikit-extremes

scikit-extremes is a python library to perform univariate extreme value calculations. It requires Numpy, Scipy, Matplotlib, and Numdifftools to work with scikit-extremes. It has two main classical approaches to calculate extreme values.

Gumbel/Generalised extreme value distribution + Block maxima

A classical approach that takes the maxima of long data blocks like annual maxima and reduces the data by a significant amount. The generalized extreme value (GEV) distribution function has theoretical justification for fitting to block the maxima of data.

Generalised Pareto Distribution + Peak-Over-Threshold

This approach is to analyze excesses over a high threshold. Generalized Pareto Distribution function has a similar justification for fitting to excesses over a high threshold.

2.3.5 Generalized Pareto Distribution (GPD)

POT data consist of selecting all the events exceeding a high threshold. When the threshold value increases, the limit distribution of a POT series can be approximated with a Generalized Pareto Distribution. Rather than fitting an EVD to the extreme values of x, the POT approach tries to fit a GPD to the excesses over threshold values $(x - t)$. The excess over a threshold, written $(x - t)$ are likely to follow a GPD with scale (β) and shape (γ) parameters [26]. The location parameter μ is null in our case [27, 28, 25, 29].

$$G_{\gamma,\beta}(x) = \begin{cases} 1 - \left(1 + \frac{\gamma x}{\beta}\right), \text{ if } \gamma \neq 0 \\ 1 - exp\left(-\frac{x}{\beta}\right), \text{ if } \gamma = 0 \end{cases} \tag{2.3}$$

$$\text{where } \beta > 0 \text{ , } x \geq 0 \rightarrow \gamma \geq 0, \text{ and } 0 \leq x \leq -\frac{\beta}{\gamma}$$

2.3.6 Maximum Likelihood Estimation (MLE)

Some classical methods, like the method of moments, least-squares method, and probability-weighted moments can be used to estimate shape and scale parameters. However, they are less efficient and not robust compared to maximum likelihood estimation. In the method of moments, the first m statistical moments of the target distribution are compared to the moments derived from the observation, whereas in the least-squares method, a flexible method is used for fitting any data sets and distribution functions. Parameters are estimated using linear and non-linear regression

methods. Linear regression fits the distributions with two parameters, and non-linear regression fits the distributions having three or more parameters [4].

The MLE helps us to evaluate the parameters using the observations. If $x_1, x_2, ..., x_n$ are n independent observations of a random variable x, where density is parameterized by θ [25], the likelihood function is defined by:

$$\mathcal{L}(x_1, ...x_n; \theta) \ = \ \prod_{i=1}^{n} f_\theta(x_i) \tag{2.4}$$

This equation represents the joint density of these n observations. The parameter θ is estimated by maximizing the likelihood, and f is the probability of observing a data point as an outlier. Maximum likelihood can be used to estimate the parameters of GPD by maximizing the $log\mathcal{L}$.

$$log\mathcal{L}(\gamma, \sigma) \ - N_t \ log\sigma \left(1 + \frac{1}{\gamma}\right) \sum_{i=1}^{N_t} log\left(1 + \frac{\gamma}{\sigma} Y_i\right) \tag{2.5}$$

This method performs well for estimating location and scale parameters, especially when the sample size is large [30, 27]. It is common for climatic data, which usually contains many features and are collected for decades. In MLE, the parameters of the probability density function are estimated by maximizing the likelihood function. In practice, it is more convenient to work with the logarithm of likelihood.

2.3.7 Grimshaw

The trick of the Grimshaw's procedure is to reduce the two variables optimization problem to a single variable equation [25]. Let us write $l(\gamma, \sigma) = \log L(\gamma, \sigma)$. As we find an extremum of l, we look for solutions of the system $\nabla l(\gamma, \sigma) = 0$. Grimshaw has shown that if we get a solution (γ^*, σ^*) of this system then the variable $x^* = \frac{\gamma^*}{\sigma^*}$ is the solution of the scalar equation $u(x)v(x) = 1$ where:

$$u(x) = \frac{1}{N_t} \sum_{i=1}^{N_t} \frac{1}{1 + xY_i} \tag{2.6}$$

$$v(x) = 1 + \frac{1}{N_t} \sum_{i=1}^{N_t} \log(1 + xY_i) \tag{2.7}$$

Moreover, by finding a solution x^* of this equation, we can retrieve $\gamma^* = v(x^*)$ and $\sigma^* = \frac{\gamma^*}{x^*}$. Nevertheless, the solutions of this equation give only possible candidates for the maximum of l, so we have to get all the roots, to calculate the corresponding likelihood and keep the best tuple $(\hat{\gamma}, \hat{\sigma})$ as our final estimates. In fact, the values $1 + xY_i$ must be strictly positives. As the Y_i are positive, we must find x^* on $(-\frac{1}{Y^M}, +\infty)$ where Y^M is the maximum value among Y_i. Grimshaw calculates also an upper-bound x^*_{max} for this root search:

$$x^*_{max} = 2\frac{\bar{Y} - Y^m}{(Y^m)^2} \tag{2.8}$$

where Y^m is the mimimum value among Y_i and $\bar{Y}$ is the mean of the Y_i. Finally, the number of roots is not known, and 0 is always a solution, so the implementation must find all the solutions and pick up those which maximize the likelihood [25].

2.3.8 Confidence intervals

In order to identify an area of unpredictability or variability, confidence intervals may be utilized. These intervals may also be used for testing theories or suppositions of significant deviations that fall in the category of the theory of temporal clustering, or cases where there is no apparent pattern in climatic data. The confidence interval identifies the boundaries of expected variability, and all regions that are outside these boundaries are deemed to be statistically significant. Consequently, all regions

that are within the boundaries are statistically insignificant, i.e., the hypothesis is accepted. Following this standard, the statistical significance of any particular theory can easily be determined. By using the method of parametric bootstrapping, a set of extreme values is placed in distribution. Then random values are created, and finally, confidence intervals are approximated. Initially, this method was used only for independent values, but with the development of block bootstrapping, it is expanded to cover the scrutiny of dependent values. Block bootstrapping groups data in blocks from which the resampling is made and thus, preserving the time within the series [31, 4]. Nonetheless, the data is still considered to be independent since the POT data is taken out only after applying independence measures.

2.3.9 Bayesian analysis

By using Bayes theorem and applying specific conjectures, values may be neutrally designated to a variety of climatic models that results in a probability distribution of climate change in the future [32]. Bayesian analysis has concerns about the parameters vector as it is viewed as a constant quantity, which must be estimated. Therefore, there has to be a prior distribution that is supposed to reveal earlier awareness of the parameters and can be evaluated without relying solely on observations. Supposing that there is a set of existing data, considering the probability and the prior distribution, the parameters of posterior distribution can be obtained through Bayes theorem. This posterior distribution is now considered as the updated version of earlier data. It has a more widely general estimation compared to the traditional point estimation. Nevertheless, as the coverage increases, the posterior distribution can no longer be used in simple applications. Hence more complicated computational approaches are necessary [33]. Researchers have suggested the use of a Bayesian predictive technique to the peak over threshold method [34]. In this approach, values in the higher-order

statistics are treated as separate parameters, along with a suitable prior distribution. A weighted average is also derived from several potential threshold values by utilizing the predictive distribution, which eliminates the difficulties introduced from using small sizes of samples. Given the probability and the prior distributions, the Bayes theorem can thus be used to acquire the posterior distribution [35].

2.3.10 NEAT-python

Neuroevolution of augmenting topologies (NEAT), artificial neural networks with genetic algorithms deal with a crossover of different topologies. They grow incrementally from a minimal structure by protecting structural innovation, developed for evolving arbitrary neural networks. NEAT-python is a pure python implementation of NEAT with the Python standard library and has no other dependencies [36].

2.3.11 Ramp detection techniques

In the case of wind ramp events, the time scale of interest is on the order of minutes to hours. First, we must bring all the data to a consistent format by normalizing them to nameplate capacity (energy produced by a turbine running at optimal wind speeds) of the system and by filling all the missing values. We can also ignore some missing values and neglect them in the statistical analysis. Finally, preprocessed data can be used to model the ramp events using the statistical framework [37]. We can consider an event as a ramp, if a large increase or decrease in wind power within a given time span i.e, the difference between the magnitudes of power generation in a time interval is greater than a specific threshold value $\mid \Delta p_{(t)} \mid = p_{(t)} - p_{(t-\Delta t)}$ and $\mid \Delta p_{(t)} \mid > \mu$, where μ is the threshold value, $\Delta p_{(t)}$ is the power at time t, $p_{(t-\Delta t)}$ is power at time $(t - \Delta t)$, and Δt is short time span [38, 16, 15]. There is a chance of missing a few ramp events occurring between the endpoints of an interval. Hence, if

the difference between the maximum and minimum values of the wind power within a time span is greater than a specified threshold i.e, $p_{max_{t_1}} - p_{min_{t_2}} > \mu_0$, then it can be a ramp event [38, 16]. The rate of increase (or decrease) in wind power within a time span is greater than a specified threshold value, i.e., $\frac{p_{(t)} - p_{(t-\Delta t)}}{\Delta t} > \mu_1$, is also considered as a ramp [16].

Trend fitting

The time-series wind power data contain uncertainty and underlying structure at multiple time scales. Investigating phenomenon at appropriate time scales requires eliminating noise or trends outside the time scale of interest. It is essential to reduce data size while still preserving appropriate trends in the data set. Trend fitting can be done as a preprocessing step to remove short term fluctuations in wind power. It also eliminates the noises and trends outside the time scale [16].

Dynamic programming

Optimal ramp intervals mean that a ramp event refers to the longest series of points that comply with the predetermined ramp rules. A ramp event has a start and an endpoint, both of which define the boundaries of the ramp interval. The objective is to find the longest intervals, and it can be done by analyzing each subsequence of the original data. The best approach for this is through dynamic programming. If an interval complies with ramp rules, all its subintervals also comply with the ramp rules. Each of these intervals is assigned a score that is a function corresponding to the interval length. Thus, finding a set of intervals that maximizes the score is all needed. Given an appropriate scoring function on intervals of the time series, the optimal ramp start times and stop times are recovered by maximizing the objective function J according to the dynamic program.

$$J(i,j) = \max_{i<k\leq j} W(i,k) + J(k+1,j) \tag{2.9}$$

$J(i,j)$ is the maximum attained score in the interval, i.e., it is computed as the maximum over $j-i$ subproblems. i, j and k are temporary variables to represent the time interval. Maximizing overall subproblems yields an optimal solution since each subproblem is in terms of a maximum score in the interval $(k+1,j)$. W(i,k) is the positive weight given to the interval (i,k). The dynamic programming requires a proper cost function that evaluates the cost of each subsequence, and the ramp score can be computed as the maximum over subproblems [16].

Sliding window

To efficiently detect data trends within a large temporal wind dataset, input signals are split into several overlapping sections parameterized by window length. Wind ramps can be identified by processing these overlapping sections individually or in a parallel fashion, which significantly reduces the computation time. The advantage of this technique can be maintained by maximizing the ramp duration parameter, i.e., if the window length is greater than the most prolonged ramp duration, all the wind ramps are easily detected. It ensures that no ramp event is missing, and the detected ramps are aggregated later. The window-based method can perform better localization of outliers [16].

Mutual information

Essential features are required to build accurate wind ramp models. One of the commonly used methods is the mutual information approach, where the mutual independence of two random variables is measured. A feature is considered significant

if the mutual information value between the candidate feature and target class is high. Also, the mutual information value between the candidate feature and selected features need to be low, making the features independent of each other. Considering only the essential features can result in the improvement of classification accuracy and computational costs can be reduced. Hence, for the classification of ramp-up/down events, mutual information is useful for filtering out the less essential inputs of the classification engine [39, 15].

Chapter 3

Prediction Accuracy Improvement Framework

Environmental extreme events such as heatwaves, typhoons, tsunamis, torrential downpour, wind ramps, or hurricanes can have profound impacts and cause the loss of human, animal, and plant lives. For example, floods can cause famines; high temperatures and high-speed wind can lead to forest fires or health problems in humans, like heat stroke. Low precipitation and heat waves leading to forest fires have occurred in different areas of the world. Therefore, it is essential to study and understand extreme environmental events in order to predict them so that we can prevent or minimize the loss [1, 11]. We proposed this framework to predict extreme environmental events using historical data. We improved the predictive accuracy of machine learning models using extreme event detection algorithm, various other statistical methods, and a novel approach to the training process.

3.1 Proposed method

The prediction accuracy of machine learning models (like gradient boosting regressor, random forest regressor, and support vector regressor) can be increased by training separate models, one for normal data and one for abnormal data. DSPOT and scikit-extremes are used to eliminate the concept of a subjective threshold. High threshold values of the target variable are obtained to split the data into normal training,

abnormal training, normal test, and abnormal test data. Then predictive models can be trained and tested separately for normal and anomaly data. The sliding window approach is integrated to see how well the models perform in predicting future events.

3.2 Statistical methods implemented

3.2.1 Streaming Peak Over Threshold with Drift

This method performs SPOT thresholding on local variations. It contains some additional steps to compute variable changes. For these stages, we principally use a sliding window over normal observation to calculate a local normal behavior M_i through averaging. We logically update the local behavior only in normal or peak cases. We can retrieve the real extreme quantiles sequentially by adding M_i to the calculated z_q. Such a choice to model the local behavior is a very efficient way to adapt SPOT to drifting contexts. We implemented an algorithm in python from a research article [25] to identify outliers in streaming data considering the variable change.

3.2.2 scikit-extremes venv

We created a scikit-extremes virtual environment to perform univariate extreme value calculations. Numpy, Scipy, Matplotlib, and Numdifftools are required to work with scikit-extremes. It has two main classical approaches to calculate extreme values. They are Gumbel/Generalised Extreme Value Distribution + Block Maxima and Generalised Pareto Distribution + Peak-Over-Threshold.

3.2.3 Sliding window

To efficiently detect anomalies and data trends within a large time-series dataset, input signals are split into several overlapping sections parameterized by window length.

Anomalies can be identified by processing these overlapping sections individually or in a parallel fashion, which significantly reduces the computation time. It ensures that no anomaly is missed, and the detected anomalies are aggregated later. The window-based method can perform better localization of anomalies. We also integrated this approach with predictive models to see how well they are performing in predicting future events [16].

3.3 Work flow

First, we gathered data from various sources and did data preprocessing techniques like data cleansing, formatting, feature selection, and training data - test data separation. We then trained sklearn's gradient boosting regression, random forest regression, and support vector regression models on the training data. We made predictions on the test data and obtained the accuracy results. Then we used our novel approach of training models where we separated the training data into two parts and test data into two parts using high threshold values obtained with statistical techniques like DSPOT and scikit-extremes. Now we have normal training data, normal test data, abnormal training data, and abnormal test data. We trained two models (i.e., one for normal data and one for abnormal data). We then used these models to test separately on their respective test datasets. Further, we used the R^2 scoring method to evaluate the performance of our models. The results of both approaches (without data seperation and with data seperation) are compared and visualized. The R^2 value lies in the range of 0 and 1, the models having R^2 values closer to 1 perform well and are efficient. So, if the model performance is good, we can use it for predicting future occurrences in a data stream. The workflow of our proposed method is illustrated in figure 3.1.

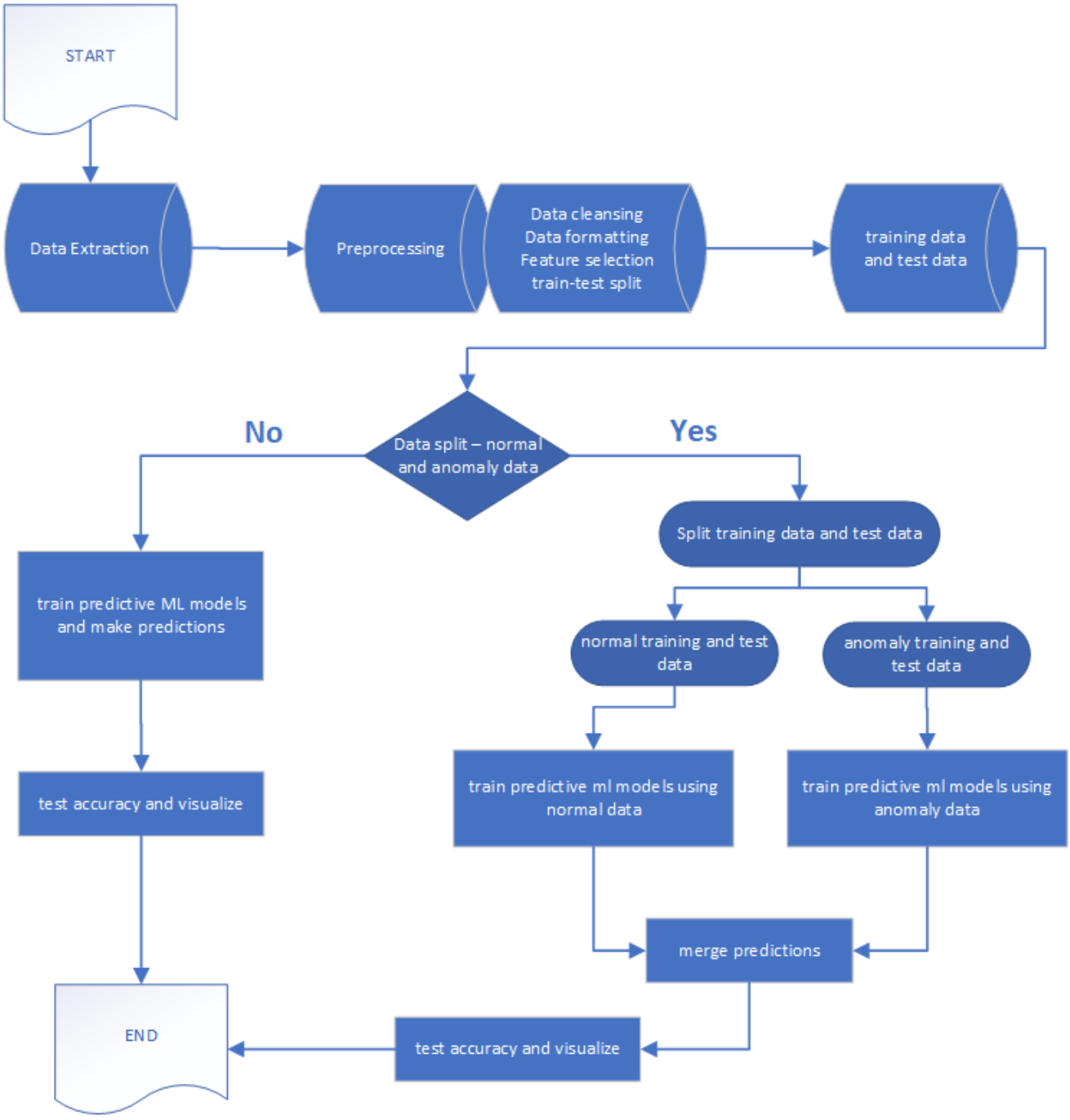

Figure 3.1: A flowchart describing the steps involved in our proposed framework

3.3.1 Data preprocessing

Data cleansing

Some missing and null values can tangibly reduce the prediction accuracy. To convert the raw data set, which is not feasible for analysis, into a clean data set, we identify incomplete, incorrect, and unnecessary parts and modified or deleted them to improve the quality of data, thereby increasing the overall productivity.

Data formatting

The data is formatted using different approaches, like converting raw data into various formats like CSV files, make sure all the variables within a given attribute are consistent, indexing, rescaling the data, and formatting the date based on our requirements.

Feature selection

Having irrelevant features in our data set can hugely impact the performance of the model, so feature selection is essential to obtain good results. Articulating the problem and considering the more valuable features based on the value to be predicted is a crucial factor. It might be tempting to include as many features as possible, but including only the critical features is a key factor in reducing complexity and computational costs.

Separating the data set into training and test data

Mostly we worked on time series data, so we preferred to split the data based on the time scale. For example, when we have data from 2012 to 2019, we considered the training data from the year 2012 to 2017 (around 70% of data) and test data from the

year 2018 to 2019 (around 30% of data). Later, we used our proposed approach to separate the training data into normal training and abnormal training data, test data into normal test data, and abnormal test data by obtaining high threshold values.

3.3.2 Training models - without data split

We trained the sklearn's gradient boosting regression, random forest regression, and support vector regression models using the training data and made predictions on the test data. Further, we used the R^2 scoring method to evaluate the performance.

3.3.3 Training models - with data split

Split training data into two parts

A high threshold value of the target variable in the training data is computed using two methods, i.e., scikit-extremes or DSPOT algorithm. Then we separated the training data based on those threshold values (i.e., normal training data and abnormal training data) to further train the models based on our approach.

Split test data into two parts

Similarly, a high threshold value of the target variable in the test data is computed using the scikit-extremes or DSPOT. Then we separated the training data based on those threshold values (i.e., normal test data and abnormal test data) to further test the trained models.

Training and evaluating the model performance

Based on our approach, we divide the training and test data into two parts each. First, we determine a high threshold value (say 95 percentile) of the target variable; we then

update that threshold value using the DSPOT algorithm or scikit-extremes. Using the updated threshold, we separate the data set into normal data and abnormal data. Now we will build and train two models, one for normal data and one for abnormal data. Then we use these models to test separately for the normal test data and abnormal test data to obtain the R^2 value for evaluating the model performance. So if the model performance is good, we can use it for the incoming observations of data streams.

3.4 Predictive models used in scikit-learn

scikit-learn is an open-source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection, prediction, and evaluation. We mainly focused on three machine learning algorithms with their default parameter values; they are gradient boosting regression, random forest regression, and support vector regression.

3.4.1 Gradient Boosting Regression (GBR)

In this approach, an additive model is built in a forward stage-wise fashion. In each stage, a regression tree is fit on the negative gradient of the given loss function. It also allows for the optimization of arbitrary differentiable loss functions.

3.4.2 Random Forest Regression (RFR)

A random forest is a meta estimator that fits multiple classifying decision trees on various sub-samples of the data set. It also used averaging to improve the accuracy of predictions and controls the over-fit scenario. The samples are drawn with replacement in a bootstrap scenario. The size of the sub-sample is always the same as the

size of the original input sample.

3.4.3 Support Vector Regression (SVR)

SVR gives us the flexibility to define how much error is acceptable in our model and finds an appropriate hyperplane in higher dimensions to fit the data. In simple regression, we try to minimize the error rate, while in SVR, we try to fit the error within a certain threshold.

Chapter 4

Case study

4.1 Acoustic Doppler Current Profiler (ADCP) - ocean surface currents

Ocean surface currents have been measured for around 15+ years. The hourly measurements are gathered in a 6 km resolution of 2 to 3 meter water columns, see figure 4.1, taken from Computer Science seminar by Dr. Mike Muglia. Wave height, period, direction, and water surface temperatures are also measured. ADCP measures the currents over the water column, temperature, and salinity are measured by conductivity, temperature, and depth (CTD).

This data is gathered from U.S. Army Engineer Research and Development Center, Coastal and Hydraulics Laboratory, Field Research Facility, Duck, North Carolina. It contains 4 - meter bins starting above the ADCP and reaching towards the ocean's surface. Most variables are organized by BIN x ENSEMBLE (m x n matrix), where the lower bins are closer to the ADCP. As bin number goes up, the bin is farther from the ADCP and closer to the surface. The time is recorded in MATLAB/computer time, and the measurements were taken every 50 seconds by the ADCP and then averaged into 10 - minute ensembles. Each deployment was stitched together to create one single data file for the Gulf Stream location. Then the 10-minute ensembles were quality controlled into hourly averaged ensembles. Not a number (NaN) values are found between the ADCP and the first good bins of data, as well as between the

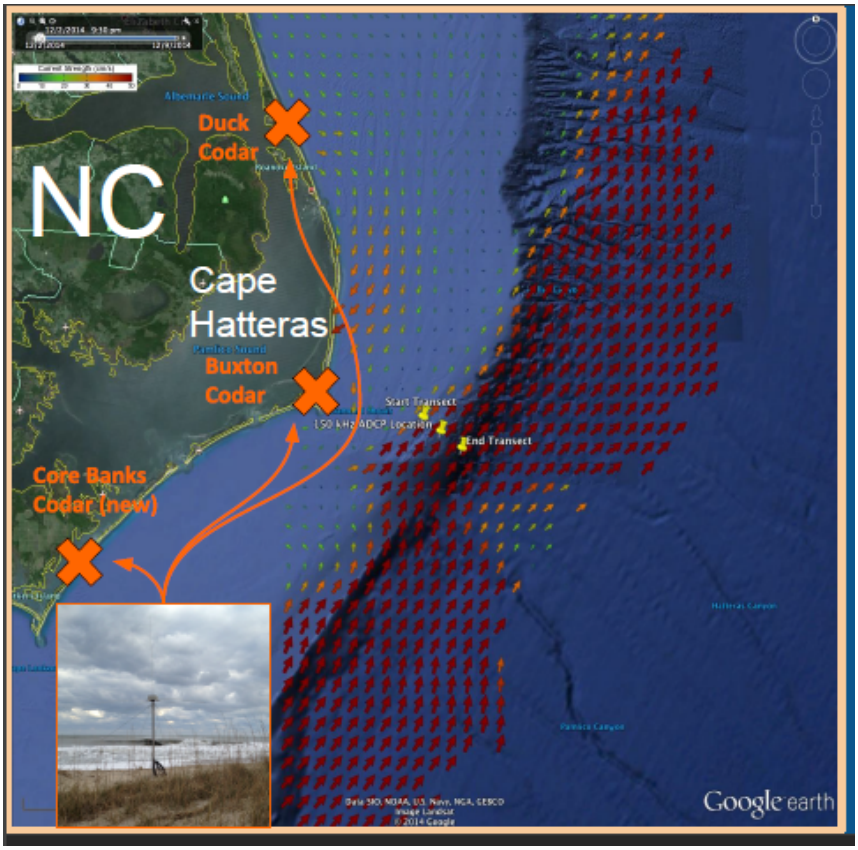

Figure 4.1: ADCP is deployed at marked yellow positions in coastal gulf stream locations of North Carolina, and measurements were recorded

last good bins of data and the ocean's surface. Due to these values, gaps exist between deployments due to the nature of deploying and recovering instruments offshore (like bad weather). The dataset contains the following variables (we used east_vel, north_vel, and vert_vel in our work) for water velocity analysis:

backscat : The amount of acoustic backscatter (ABS) in each bin. ABS measurement is a non-intrusive technique for monitoring suspended sediment particles in the water column. BINxENSEMBLE (67x33640).

east_vel : Water velocity component to the east (u), measured in m/s. Negative values mean flow is to the west. BINxENSEMBLE (67x33640)

echo_intensity : The intensity of echos measured by the ADCP per BIN. BINxENSEMBLE (67x33640)

labels : Descriptors of each variable listed in a README

mtime : computer time vector. UTC time. TIMExENSEMBLE(1x33640)

north _vel : Water velocity component to the north (v), measured in m/s. Negative values mean flow is to the south. BINxENSEMBLE (67x33640)

platform : Metadata information including id (site name), loc (location type), lat (site latitude), lon (site longitude), mvar (magnetic variation at site location), water_depth (range of water depths from all deployments)

vert_vel : Water velocity component in the vertical direction (w), measured in m/s. Negative values mean flow is downwards. BINxENSEMBLE (67x33640)

water_level : Measurement of water height above or below Mean Sea Level (MSL). Positive values indicate water above MSL, while negative values indicate water below MSL. WATERLEVELxENSEMBLE (1x33640)

z : depth vector, depth for each data bin, where each bin is vertically 4 meters and begins just above the ADCP, so lower indexed bins are deeper compared to the bins with a higher index DEPTHxBIN (1x67).

4.1.1 Results obtained

We preprocessed the dataset using data cleansing and data formatting techniques to increase the overall productivity. The target variable is a spatial vector of water velocity, and it is predicted using all the remaining spatial vectors. We separated the time-series sequence into training data (80%) and test data (20%). Then we trained sklearn machine learning models like gradient boosting regression, random forest regression, and support vector regression. First, we trained a model using the training data set and used it to make predictions on the test data, see figure 4.2. Then the model performance is computed based on the predictions and the ground truth values of the test data using the R^2 scoring method, see table 4.1. We also used sliding window approach to compute our model performance in predicting future events, see table 4.2 and figure 4.3. RFR and GBR models performed better compared to the

SVR model; there is a significant decrease in SVR's accuracy as we increased the number of hours ahead.

Model	R^2 **value**
gradient boosting regression	0.980
random forest regression	0.983
support vector regression	0.890

Table 4.1: R^2 values of ML models without data split (case study - 1)

Figure 4.2: GBR (top), RFR (middle), and SVR (bottom) - comparision of ground truth values (green) VS predictions (blue) without data split

Compared to other datasets having various features in predicting the target variable, the features and target variables of this dataset are spatial vectors of the water velocity. The target variable, which is a spatial vector depends only on other spatial vectors. So, we decided not to split the dataset into normal and abnormal values.

No of hours ahead	**gbr**	**rfr**	**svr**
1	0.979	0.979	0.966
3	0.866	0.868	0.859
5	0.723	0.725	0.724
7	0.581	0.575	0.575
9	0.432	0.429	0.433
11	0.328	0.318	0.269
13	0.195	0.218	0.108
15	0.0481	0.061	-0.068
17	-0.096	-0.065	-0.249
19	-0.166	-0.160	-0.368

Table 4.2: capability of ML models in predicting future events (case study - 1)

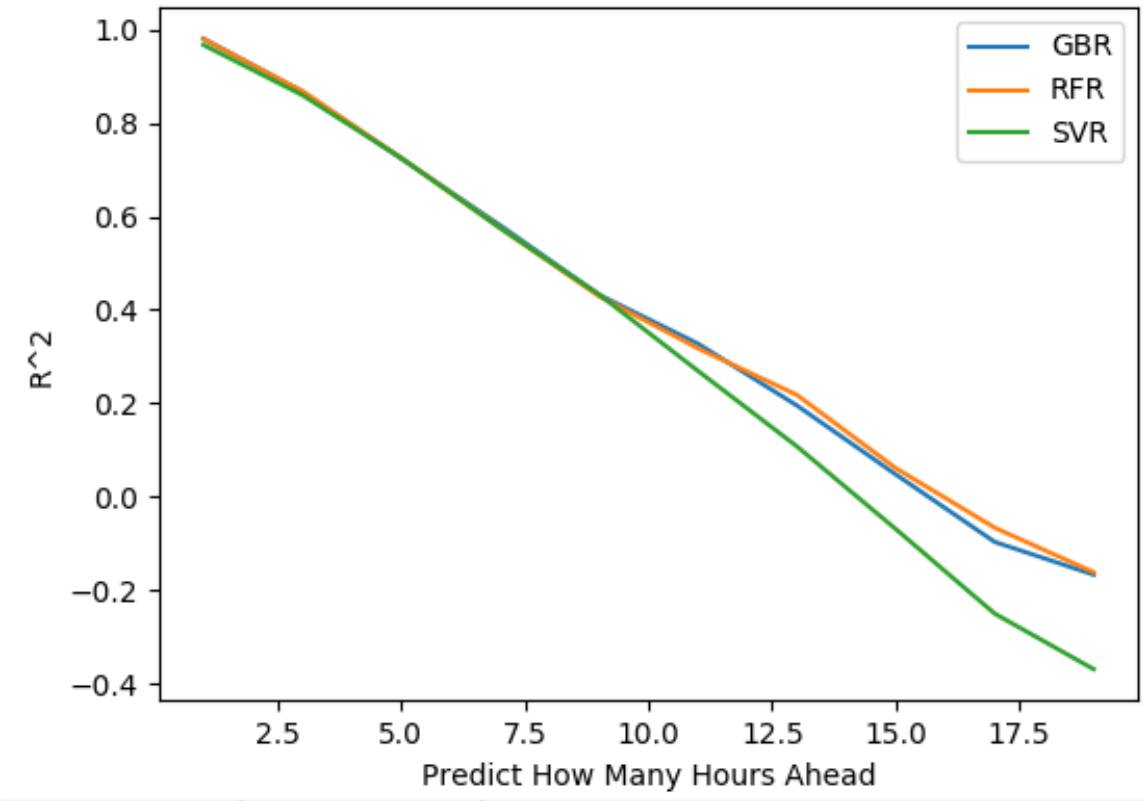

Figure 4.3: Comparision of R^2 values vs number of hours ahead (case study - 1)

4.2 CDIP - monitoring waves along the coastlines of the United States

This data is gathered from the Coastal Data Information Program (CDIP), operated by the Ocean Engineering Research Group (OERG), part of the Integrative Oceanography Division (IOD) at Scripps Institution of Oceanography (SIO). This program measures, analyzes, archives, and disseminates coastal environment data for use by coastal engineers, planners, and managers, as well as scientists and mariners. The CDIP is an extensive network for monitoring waves and beaches along the coastlines of the United States. Since its inception in 1975, the program has produced a vast database of publicly-accessible environmental data for use by coastal engineers

and planners, scientists, mariners, and marine enthusiasts. The program has also remained at the forefront of coastal monitoring, developing numerous innovations in instrumentation, system control and management, computer hardware and software, field equipment, and installation techniques. Waves, shoreline change, and sea surface temperatures are monitored and predicted. Parameters include significant wave height (Hs), peak period (Tp), peak direction (Dp), and sea surface temperature (SST).

4.2.1 Results obtained

The data set is preprocessed by some techniques like data cleansing and data formatting to increase the overall productivity. We also added features like lag (to incorporate feedback over time), time_sin (used sine function to convert time information into a floating point value), time_sin_no_year (time_sin without year) and current season. Then we separated the data into training data (observations from the year 2012 to 2017) and test data (observations from the year 2018 to 2019). Then we trained sklearn machine learning models like gradient boosting regression, random forest regression, and support vector regression. First, the training data set is used to train the model and used it to make predictions on the test data, see figure 4.4. The model performance is computed based on the predictions and the ground truth values of the test data using the R^2 scoring method, see table 4.3.

Model	R^2 **value**
gradient boosting regression	0.698
random forest regression	0.613
support vector regression	0.575

Table 4.3: R^2 values of ML models without data split (case study - 2)

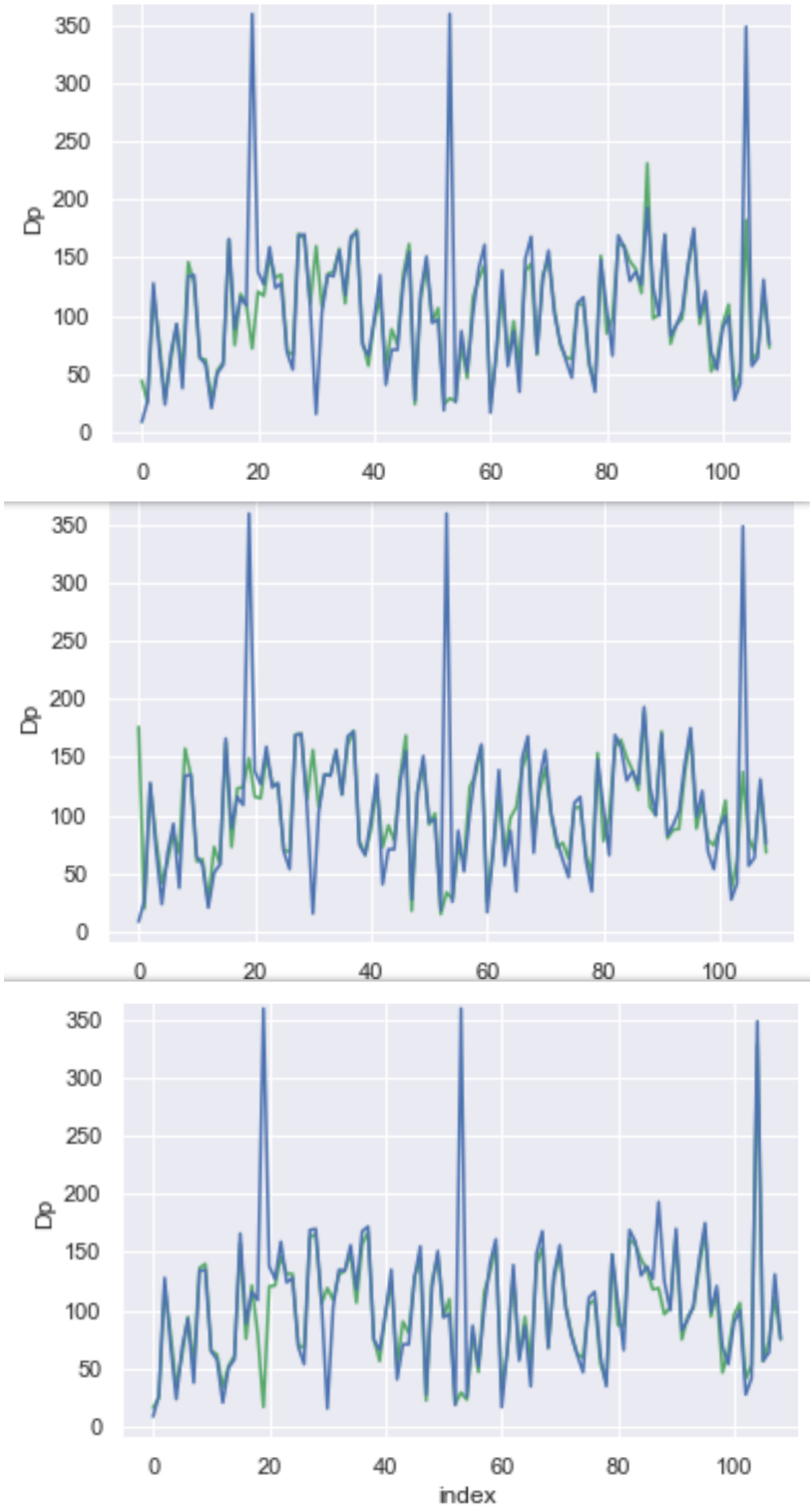

Figure 4.4: GBR (top), RFR (middle), and SVR (bottom) - comparision of ground truth values (green) VS predictions (blue) without data split

Also, we computed the threshold values of the target variable in the training dataset using DSPOT algorithm (the threshold value is 185.62) and scikit-extremes library (the threshold value is 213.73). Based on these threshold values, we separated both training data into normal training data and abnormal training data, see figure 4.5. Then we trained two models for every machine learning algorithm used, one with the normal training data and the other with the abnormal training data. Now we indexed the test data, divided it into normal test data and abnormal test data using DSPOT and scikit-extremes threshold values (183.75 and 213.61 respectively), and used them to make predictions separately. Now, these predictions are merged back based on the index values, and R^2 values are computed with the help of the prediction values and the ground truth values, see figures 4.6, 4.7. Using our framework, we improved the predictive accuracy by 20% to 25%, compare the R^2 values in the tables 4.3, 4.4, 4.5.

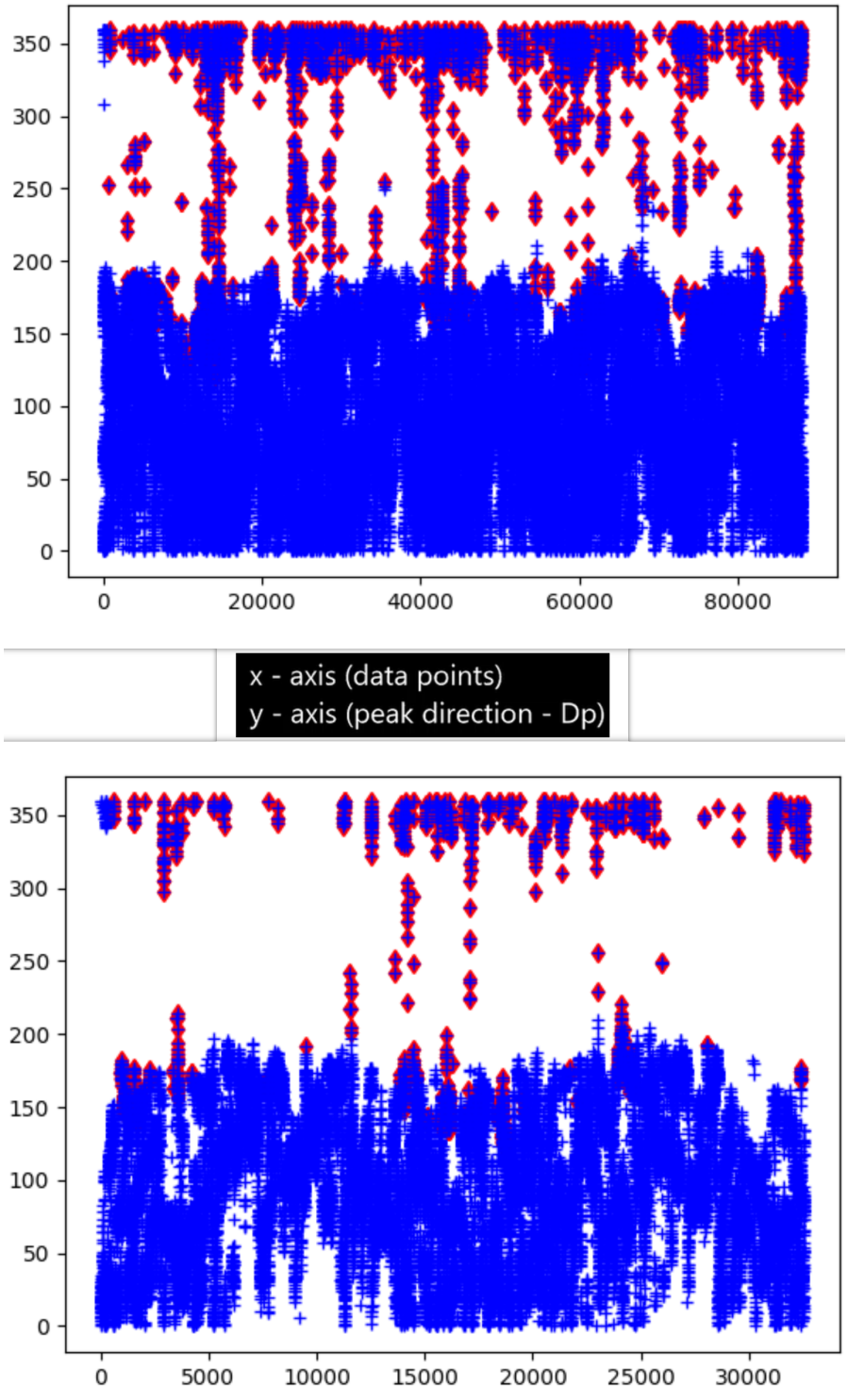

Figure 4.5: data split using DSPOT , training data around 90k observations (top) and test data around 32k observations (bottom), normal target values (blue) and abnormal target values (red)

Model	R^2 **value**
gradient boosting regression	0.916
random forest regression	0.913
support vector regression	0.701

Table 4.4: R^2 values of ML models with DSPOT data split (case study - 2)

Figure 4.6: GBR (top), RFR (middle), and SVR (bottom) - comparision of ground truth values (green) VS predictions (blue) with data split using DSPOT

Model	R^2 **value**
gradient boosting regression	0.923
random forest regression	0.894
support vector regression	0.753

Table 4.5: R^2 values of ML models with scikit-extremes data split (case study - 2)

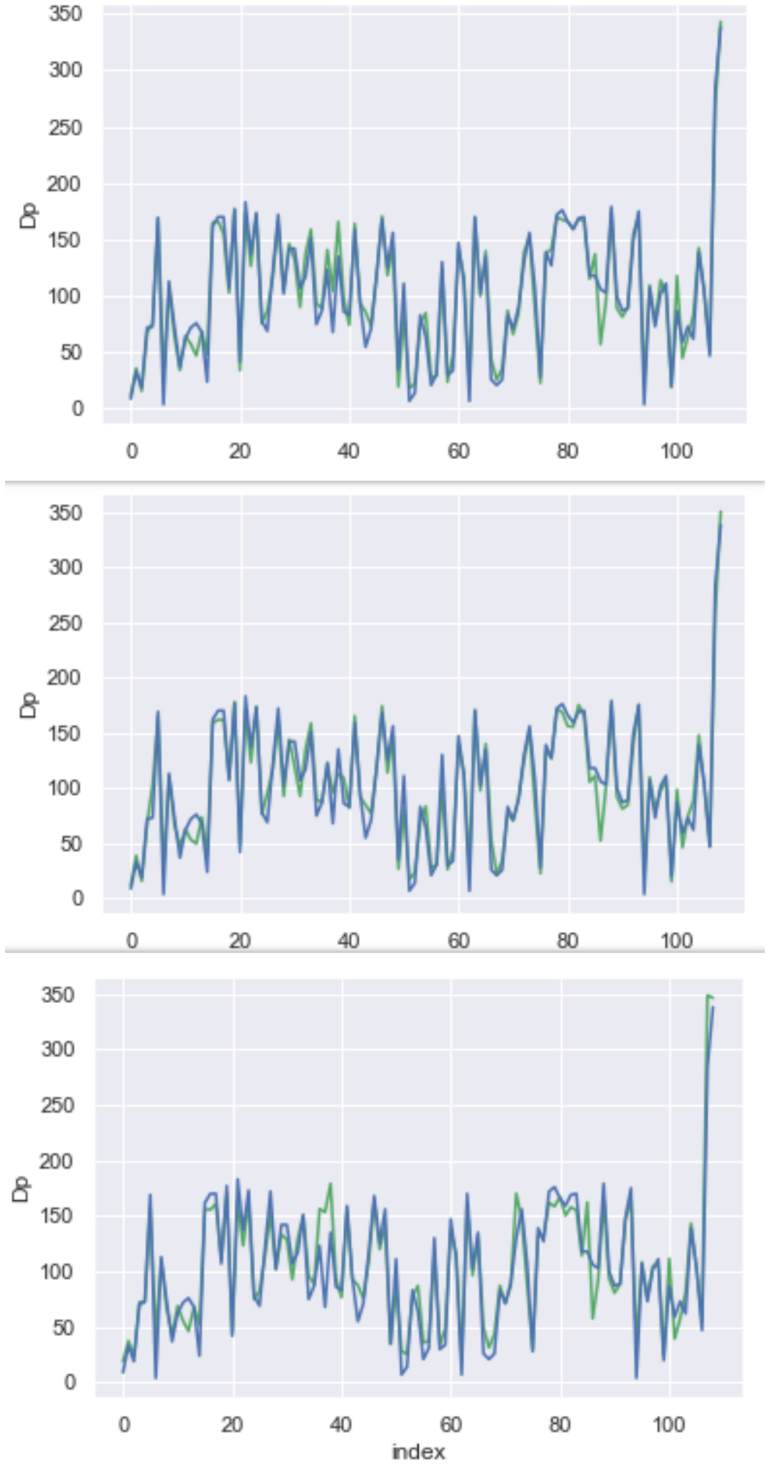

Figure 4.7: GBR (top), RFR (middle), and SVR (bottom) - comparision of ground truth values (green) VS predictions (blue) with data split using scikit-extremes

4.3 Research Contribution

Most of the existing environmental models deal with spatial and temporal variables on hard and subjective thresholds. DSOPT and scikit-extremes techniques are used to compute the threshold values, thereby eliminating the idea of a subjective threshold. Our novel approach of splitting the data into normal data and abnormal data using the threshold values is a key factor in obtaining a better accuracy in predicting future events. Models like gradient boosting regression, random forest regression, and support vector regression can be trained and tested separately using two models (one for normal data and one for abnormal data). The sliding window approach is integrated to see how well the models perform in predicting future events. By our framework, we improved the predictive accuracy of machine learning models by 20% to 25% increase in R^2 value.

Chapter 5

Conclusion

In this thesis, we proposed a framework to forecast and improve the prediction accuracy of environmental models using extreme event detection algorithm. We integrated the sliding window approach to see how well our models predict future events. In our framework, we used a novel approach of training process where we separated the data into normal data and abnormal data. Machine learning models are trained and tested separately using two models (one for normal data and one for abnormal data). Most of the existing methods deal with spatial and temporal variables on hard and subjective thresholds. So we built predictive machine learning models using DSPOT and scikit-extremes to eliminate the idea of a subjective threshold. To test the proposed framework, we collected coastal data from various sources, implemented our workflow, and obtained results to show our model's performance. The R^2 values increased by 20% to 25% for all the machine learning models trained using our approach. In addition, these results would help to study the North Carolina gulf stream as an energy source and also for determining key factors related to ocean currents, water mass exchange dynamics of coastal areas in North Carolina. This framework can be used to forecast and predict several other environmental events.

We also surveyed and summarized all the recent works on extreme environmental events, classified them into wind ramp events and climatic extremes along with their

detection techniques. Apart from that, we introduced various outlier detection methods depending on the scenario and available data. A neural network-based system might be an alternative approach, which is capable of learning a broad class of patterns from complex multi-variable data and avoiding subjective threshold for extreme events detection. Neural networks eliminate the need for feature extraction, which is one of the most critical and time-consuming parts of traditional machine learning methods. However, training a neural network model with a limited amount of training data is quite challenging as these approaches require more data than other machine learning methods. Also, their performance is strongly correlated with the amount of available training data. Data are much harder and more expensive to collect than developing and applying the algorithms for execution. Without enough data, a neural network may not be able to achieve the desired level of accuracy. Training a neural network can help us extract features and classify images to detect extreme events like tropical cyclones, atmospheric rivers, and hurricanes.

Chapter 6

Acknowledgements

This work is supported in part by the National Science Foundation IUSE/PFE: RED award #1730568.

BIBLIOGRAPHY

[1] R. Matta, R. Wu, and S. Guan, "Environmental extreme events detection: A survey," in *Proceedings of 28th International Conference on Software Engineering and Data Engineering*, vol. 64, 2019, pp. 184–193.

[2] G. Krempl, I. Žliobaite, D. Brzeziński, E. Hüllermeier, M. Last, V. Lemaire, T. Noack, A. Shaker, S. Sievi, M. Spiliopoulou *et al.*, "Open challenges for data stream mining research," *ACM SIGKDD explorations newsletter*, vol. 16, no. 1, pp. 1–10, 2014.

[3] Y. Zhang, N. Meratnia, and P. J. Havinga, "Outlier detection techniques for wireless sensor networks: A survey." *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.

[4] G. Van Vledder, Y. Goda, P. Hawkes, E. Mansard, M. J. Martin, M. Mathiesen, E. Peltier, and E. Thompson, "Case studies of extreme wave analysis: a comparative analysis," in *Proceedings of the second international symposium on ocean wave measurement and analysis*. ASCE New York, 1993, pp. 978–992.

[5] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2014.

[6] W. Zhang, J. Wu, and J. Yu, "An improved method of outlier detection based on frequent pattern," in *2010 WASE International Conference on Information Engineering*, vol. 2. IEEE, 2010, pp. 3–6.

[7] Z. Zhao, C. Mohan, and K. Mehrotra, "Adaptive sampling and learning for unsupervised outlier detection," in *The Twenty-Ninth International Flairs Conference*, 2016.

[8] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2013.

[9] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," in *2007 IEEE symposium on computational intelligence and data mining*. IEEE, 2007, pp. 504–515.

[10] Y. Yan, L. Cao, C. Kulhman, and E. Rundensteiner, "Distributed local outlier detection in big data," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2017, pp. 1225–1234.

[11] R. W. Katz and B. G. Brown, "Extreme events in a changing climate: variability is more important than averages," *Climatic change*, vol. 21, no. 3, pp. 289–302, 1992.

[12] C. Frei and C. Schär, "Detection probability of trends in rare events: Theory and application to heavy precipitation in the alpine region," *Journal of Climate*, vol. 14, no. 7, pp. 1568–1584, 2001.

[13] T. R. Karl and R. W. Knight, "Secular trends of precipitation amount, frequency, and intensity in the united states," *Bulletin of the American Meteorological society*, vol. 79, no. 2, pp. 231–242, 1998.

[14] G. A. Meehl, J. M. Arblaster, and C. Tebaldi, "Understanding future patterns of increased precipitation intensity in climate model simulations," *Geophysical Research Letters*, vol. 32, no. 18, 2005.

[15] H. Zareipour, D. Huang, and W. Rosehart, "Wind power ramp events classification and forecasting: A data mining approach," in *2011 IEEE Power and Energy Society General Meeting*. IEEE, 2011, pp. 1–3.

[16] R. Sevlian and R. Rajagopal, "Detection and statistics of wind power ramps," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3610–3620, 2013.

[17] E. Bristol, "Swinging door trending: Adaptive trend recording?" in *ISA National Conf. Proc., 1990*, 1990, pp. 749–754.

[18] M. Cui, J. Zhang, A. R. Florita, B.-M. Hodge, D. Ke, and Y. Sun, "An optimized swinging door algorithm for identifying wind ramping events," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 1, pp. 150–162, 2015.

[19] M. Cui, J. Zhang, C. Feng, A. R. Florita, Y. Sun, and B.-M. Hodge, "Characterizing and analyzing ramping events in wind power, solar power, load, and netload," *Renewable energy*, vol. 111, pp. 227–244, 2017.

[20] R. Wu, L. Yang, C. Chen, S. Ahmad, S. M. Dascalu, and F. C. Harris Jr, "Modeling error learning based post-processor framework for hydrologic models accuracy improvement," *Geoscientific Model Development Discussions*, p. 1, 2018.

[21] G. Rosso, "Extreme value theory for time series using peak-over-threshold method," *arXiv preprint arXiv:1509.01051*, 2015.

[22] R. A. Fisher and L. H. C. Tippett, "Limiting forms of the frequency distribution of the largest or smallest member of a sample," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 24, no. 2. Cambridge University Press, 1928, pp. 180–190.

[23] B. Gnedenko, "Sur la distribution limite du terme maximum d'une serie aleatoire," *Annals of mathematics*, pp. 423–453, 1943.

[24] J. Beirlant, Y. Goegebeur, J. Segers, and J. L. Teugels, *Statistics of extremes: theory and applications*. John Wiley & Sons, 2006.

[25] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1067–1075.

[26] J. R. Lanzante, "Resistant, robust and non-parametric techniques for the analysis of climate data: Theory and examples, including applications to historical radiosonde station data," *International Journal of Climatology: A Journal of the Royal Meteorological Society*, vol. 16, no. 11, pp. 1197–1226, 1996.

[27] Y. Li, W. Cai, and E. Campbell, "Statistical modeling of extreme rainfall in southwest western australia," *Journal of climate*, vol. 18, no. 6, pp. 852–863, 2005.

[28] A. J. McNeil, "Estimating the tails of loss severity distributions using extreme value theory," *ASTIN Bulletin: The Journal of the IAA*, vol. 27, no. 1, pp. 117–137, 1997.

[29] X. Zhang, F. W. Zwiers, and G. Li, "Monte carlo experiments on the detection of trends in extreme values," *Journal of Climate*, vol. 17, no. 10, pp. 1945–1952, 2004.

[30] J. R. Hosking and J. R. Wallis, "Parameter and quantile estimation for the generalized pareto distribution," *Technometrics*, vol. 29, no. 3, pp. 339–349, 1987.

[31] V. Ntegeka and P. Willems, "Trends and multidecadal oscillations in rainfall extremes, based on a more than 100-year time series of 10 min rainfall intensities at uccle, belgium," *Water Resources Research*, vol. 44, no. 7, 2008.

[32] M. E. Shongwe, G. J. van Oldenborgh, B. van den Hurk, and M. van Aalst, "Projected changes in mean and extreme precipitation in africa under global warming. part ii: East africa," *Journal of Climate*, vol. 24, no. 14, pp. 3718–3733, 2011.

[33] B. Renard, M. Lang, and P. Bois, "Statistical analysis of extreme events in a non-stationary context via a bayesian framework: case study with peak-over-threshold data," *Stochastic environmental research and risk assessment*, vol. 21, no. 2, pp. 97–112, 2006.

[34] C. N. Behrens, H. F. Lopes, and D. Gamerman, "Bayesian analysis of extreme events with threshold estimation," *Statistical Modelling*, vol. 4, no. 3, pp. 227–244, 2004.

[35] G. D'Agostini, "A multidimensional unfolding method based on bayes' theorem," P00024378, Tech. Rep., 1994.

[36] K. O. Stanley and R. Miikkulainen, "Efficient evolution of neural network topologies," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2. IEEE, 2002, pp. 1757–1762.

[37] T. Ouyang, X. Zha, and L. Qin, "A survey of wind power ramp forecasting," *Energy and Power Engineering*, vol. 5, no. 04, p. 368, 2013.

[38] C. Kamath, "Understanding wind ramp events through analysis of historical data," in *IEEE PES T&D 2010*. IEEE, 2010, pp. 1–6.

[39] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on neural networks*, vol. 5, no. 4, pp. 537–550, 1994.