# Extreme Weather Forecasts Using Real-Time Buoy Data

Jiayi Liang, Lumie Yang, Xin Ling, Yamini Muthyala, Ying Liu

**Abstract**—This study focuses on deploying a big data system to enhance the forecasting of large storms and weather patterns along the California coast, including dangerous phenomena such as enormous waves, typhoons, and hurricanes. We aim to improve forecast accuracy and deliver timely alerts for these events by leveraging modern machine learning algorithms and big data analytics. The core inquiry explores the extent to which machine learning can utilize extensive buoy data from the National Data Buoy Center (NDBC), incorporating parameters such as wave patterns, wind speeds, air pressure, and water temperature, to improve storm intensity predictions and occurrence. Employing a robust dataset spanning one decade, we developed a predictive model by comparing the performance of various machine learning models such as Random Forest, Gradient Boosting, and XGBoosting. The model training process results in the corresponding accuracy percentages of 92.01, 88.43, and 89.61 and we chose the one with the highest accuracy in our prediction model. The integration of Kafka and Spark streaming technologies enabled real-time data processing. The results reveal that the model, when applied to live data streams, can successfully predict severe weather events, thereby enhancing our capacity to issue timely warnings and maintain vigilant monitoring. The study concludes that integrating machine learning with comprehensive environmental data sets can significantly improve the forecasting of coastal weather phenomena, potentially saving lives and mitigating economic losses.

**Index Terms**—Weather Forecasting, Kafka, Spark, Real-Time Processing, Machine Learning

## 1 Introduction

### 1.1 Background Information

The challenge of predicting severe weather is growing as our climate changes. Along the California coast, the weather can cause big waves, strong winds, and hurricanes more often than before. Climate change has made these dangerous events more frequent, and it's changing the sea's temperature and wind patterns. These changes in the ocean are signs of how the weather is shifting. Even minor fluctuations in ocean temperatures can presage extreme weather patterns. Human civilizations can be greatly damaged due to extreme weather. The National Data Buoy Center (NDBC) has been instrumental in collecting marine and atmospheric data crucial for understanding these weather patterns. Despite the availability of this data, there has been a gap in effectively leveraging it to its full potential for predictive analytics. Historically, forecasts have relied heavily on models that, while sophisticated, could benefit from the more nuanced analysis provided by recent advances in machine learning and big data technologies. Recognizing this, our research seeks to tap into these advances, using the extensive data available from strategically positioned buoy stations along the California coast. By harnessing this data, we aim to refine the accuracy of weather forecasting models, offering critical advancements in the early warning systems that coastal communities depend on for preparedness and response.

### 1.2 Problem Statement and Objective

This study addresses the critical challenge of forecasting large storms and weather patterns along the California coast, including potentially devastating extreme weather events like storms. Traditional forecasting methods may have limitations in predicting the intensity and timing of these events with the necessary accuracy. This gap in meteorological forecasting poses significant risks to public safety and infrastructure.

The primary objective of this study is to deploy a big data system that enhances the forecasting of large-scale coastal weather phenomena, leveraging the power of modern machine learning algorithms and big data analytics. The primary investigation revolves around the potential of machine learning to utilize extensive buoy data from the National Data Buoy Center (NDBC). The dataset includes crucial features such as wind speed, air pressure, and water temperature from different stations to help improve the predictions regarding the intensity and occurrence of storms. By utilizing a robust historical dataset from the past ten years, the project developed a predictive model by assessing the performance of different machine learning models such as Random Forest, Gradient Boosting, and XGBoosting. The model with the highest accuracy will be selected for the final predictive system. Also, the integration of Apache Kafka and Spark Streaming will be implemented in the project as a key step to enable the processing of real-time data. By leveraging the big data stream techniques, the system will be able to consume real-time data from NDBC every five mines and show the predictions of the possibilities of severe weather.

The project ideally will demonstrate that when applying the live data stream to the predictive model, the model will successfully forecast severe weather events. It concludes that the integration of machine learning with comprehensive environmental datasets can markedly improve the forecasting of coastal weather phenomena.

## 1.3 Data Source

### 1.3.1 The Buoy Database

This study relies on data acquired from the National Data Buoy Center (NDBC), a part of the National Oceanic and Atmospheric Administration's (NOAA) National Weather Service (NWS). The NDBC operates a network of buoy stations that provide critical real-time observations essential for meteorological research and forecasting. It is a comprehensive source known for its precise and continuous marine data collection. The Standard Meteorological Data will be used for this project within a number of NDBC datasets that are publicly accessible.

The historical data of these stations used for this study spans a period of nearly a decade, from 2014 to 2023. This rich dataset includes a variety of meteorological and oceanographic measurements crucial for our analysis, such as wind direction, wind speed, peak gust speed, sea level pressure, and air temperature. It serves as the foundation for training our predictive model. By analyzing patterns and correlations within these historical data, the model learns to recognize the signs of severe weather phenomena, thereby enhancing its forecasting capabilities.

The NDBC's real-time data streams are utilized to operationalize our models for current forecasts. The real-time data file is updated every 5 minutes. This file has the same data features as the Standard Meteorological Data file, but instead of containing observations from a single station, it has the most recent observations from all stations available on the NDBC website. The real-time data is validated and prepared to match the structure of historical data in model training, ensuring consistency in the predictive model application. This enables our system to apply the pre-trained model to current conditions, providing timely and accurate weather event predictions for selected stations.

### 1.3.2 The Storm Events Database

The Storm Events Database from the National Oceanic and Atmospheric Administration's (NOAA) National Weather Service (NWS) was used in this study as the information on extreme weather events. This database collected information about the storm events from weather reports, emergency management agencies, and other reliable resources, and provides the historical storm events data which were geographically organized by County or by NWS Forecast Zone from January 1950 to July 2023. Large-scale events such as heat were gathered by zones, whereas small-scale events such as tornadoes were gathered by counties. Users can filter the data by selecting state, county, beginning date, end date, and event types. The event types included tornadoes, thunderstorm winds, flash floods, tropical storms, and so on. Upon accessing the selected event data, a summary of information, such as the "Number of County Zone areas affected" was presented.

In this study, we focused on the historical storm events data of the California coast in the last 10 years. The selected Storm Events Dataset provided detailed information about all the storm events happening in California from 2014 to 2023, including but not limited to the event occurring date and time, location, affected areas, the type of event, casualties, injuries, and property damage. The database updated regularly to record the information when a new storm event occurs, which ensured it provided the most accurate and recent information about the storm events. The Storm Events Database was a good resource for researching and analyzing the historical storm patterns, growing trends, and the impact of severe storm events over time. It could be used with the buoy database to predict and forecast the occurrence of extreme weather events.

## 1.4 Literature Survey

B. Zhou et al., in their March 2018 paper "Online Internet Traffic Monitoring System Using Spark Streaming" published in Big Data Mining and Analytics explore the application of Spark Streaming in real-time Internet traffic analysis [6]. Instead of the traditional server-based system, the paper addressed a more efficient and distributed approach to monitor real-time traffic data. The authors describe an architecture comprising collectors, a messaging system, and a stream processor together as a system to accomplish the goal of monitoring online traffic. Network packets captured by collectors are transmitted via the Kafka messaging system to the stream processor and at the same time, Spark Streaming processes the data in real-time. The focus is on monitoring TCP performance which involves steps like preprocessing, throughput calculation, retransmission, out-of-order statistics, RTT calculation, and result summarization. The final experiment results show the system's high efficiency in processing high-speed data streams with a notable throughput and minimal processing delay. The system's resilience to failures such as slave crashes, was also tested to show the robustness of the system. The paper concludes with five key learnings: the role of micro-batching in stream data processing, strategic architectural decisions, failure handling mechanisms, algorithm efficiency in managing data volume, and the impact of packet collector design and batch intervals on system performance. The methodologies and outcomes of this study are particularly relevant to our project on deploying a big data system for enhancing weather forecasting along the California coast. The parallels drawn between real-time Internet traffic analysis and real-time environmental data processing are significant. The insights from Zhou et al.'s study, especially regarding system architecture, data processing strategies, and management of high-speed data streams, are invaluable for our approach to integrating machine learning and big data analytics for extreme weather predictive modeling.

A deep Convolutional Neural Network (CNN) classification technique was developed to detect extreme weather by Liu et al. (2016) [2]. The study selected climate simulations and reanalysis products as datasets and used them in the classification models to classify tropical cyclones, atmospheric rivers, and weather fronts. This technique could be applied to quantify trends in climate extreme events in both current-day conditions and future climate scenarios. In 2017, Racah et al. introduced a semi-supervised CNN architecture to identify extreme weather patterns and improve the localization of extreme weather events by leveraging temporal information and unlabeled data [5]. They also provided a new dataset to encourage further studies on extreme weather prediction and climate change.

In 2019, Nitu et al. published a paper detailing the development of a multi-channel decision-making system, designed as an expert system for automating decision-making in various applied areas [4]. The study is particularly relevant to our project, as one of its use cases includes the detection of the onset of tropical cyclones using data from the National Data Buoy Center (NDBC), and predicting the hurricane motion direction. The paper presents an innovative approach to enhance the decision-making process in weather forecasting, specifically in the context of tropical cyclone detection. The system developed utilizes a multi-channel framework that processes atmospheric and oceanic data from the NDBC to detect early signs of tropical cyclones and predict their path. This expert system employs advanced algorithms to analyze various environmental parameters, thereby facilitating more accurate and timely predictions of hurricane movements. The research demonstrates the system's effectiveness in identifying the initial stages of cyclones and accurately predicting their trajectory. This capability is crucial for early warning systems, enabling authorities to make informed decisions and take preventive measures to mitigate the impact of such potentially devastating weather events.

Chattopadhyay et al. made a noteworthy contribution in 2020 with their research on a data-driven framework for predicting extreme weather events [1]. Their study, which employs deep learning architectures, specifically capsule neural networks (CapsNets), represents a significant advancement in the field of meteorological forecasting. The paper focuses on the development of a data-driven framework that utilizes historical weather patterns for prediction purposes. Chattopadhyay and colleagues implemented capsule neural networks (CapsNets), a cutting-edge deep learning technology, to analyze and predict extreme weather conditions. The innovative use of CapsNets, known for their efficiency in capturing spatial hierarchies and relationships in data, marked a significant shift from traditional neural network models. Their findings indicate that this approach is particularly effective in forecasting extreme weather events. The predictive model developed was able to provide early warnings by accurately identifying potential weather anomalies based on historical data patterns. This capability is crucial in enhancing preparedness and response strategies for extreme weather phenomena.

In 2022, Lou et al. introduced a novel approach in the realm of maritime navigation with their design of an automatic ship-driving scheme [3]. Their research, focusing on wave height prediction models using Long Short-Term Memory (LSTM) networks, offers significant insights for applications requiring accurate oceanic data analysis. Lou et al.'s paper centers on developing two distinct wave height prediction models for open sea and offshore conditions, utilizing LSTM networks. The ocean data for this study were sourced from the National Data Buoy Center (NDBC), ensuring a robust and reliable dataset for analysis. The models employed evaluation metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the correlation coefficient to assess their performance. The results of this study are particularly groundbreaking for maritime operations. By leveraging the predictive models developed, ships can adjust their routes in real-time, responding to the predicted wave heights. This ability to anticipate high-wave conditions and adjust accordingly significantly enhances maritime safety, helping vessels avoid areas with potentially hazardous wave conditions.

## 2 Architecture of the System

The architecture of the system is specifically built for the efficient and speedy handling of massive amounts of data, smoothly integrating raw observations into actionable weather predictions. The architecture was designed with a data collection module that interacts directly with the National Data Buoy Center, retrieving the latest data every five minutes. This data is immediately processed by a Kafka Producer to ensure that the raw data is cleaned and processed to match the format required by the machine learning models. The Kafka Consumer receives the preprocessed data while it is passed to the topic. Within the Consumer, the machine learning component, powered by PySpark, is at the core of the system, analyzing incoming data and producing forecasts using advanced algorithms. Kafka, which manages the data flow, and Spark Streaming, which applies predictive models to the live data stream, enable real-time prediction capabilities. The resulting forecasts are transformed into a user-friendly output, giving critical weather alerts to users as soon as possible. The overflow is shown below in Fig1.

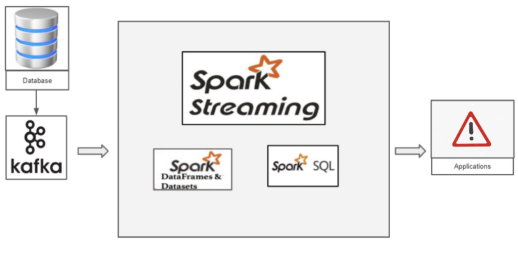


Fig. 1. The Flowchart

## 3 Methodology

### 3.1 Data Collection

The data collection process for this research was a crucial step in ensuring the accuracy and reliability of our predictive models. We focused on selecting 13 buoy stations along the California coast, a decision driven by several factors. These included the geographical coverage of the stations, ensuring a comprehensive representation of the diverse coastal conditions of California. Additionally, we considered the historical reliability and consistency of data from these stations, prioritizing those with the most complete and uninterrupted data records. The types of data collected from these buoy stations were comprehensive, encompassing a range of oceanic and atmospheric parameters. Key data points included wave patterns, wind speeds, air pressure, and water temperature. These elements were critical in understanding and predicting the various aspects of weather patterns, especially those leading to extreme weather events. The historical data used in our study is from 2014 to 2023.

By using this ten-year range robust dataset, an in-depth analysis of trends and patterns over time can be created. For real-time data access, we use Apache Kafka to retrieve updates from the buoy stations at regular intervals. The data was updated every 5 minutes, ensuring that our models had access to the most current information available. This frequent update rate was essential for the real-time aspect of our predictive models, enabling them to respond quickly to changing weather conditions. Integrating this real-time data with our historical analysis was a key factor in developing models that could provide timely and accurate weather predictions.

#### 3.1.1 The Buoy Database

For this study, thirteen buoy stations along the California coast were chosen based on their geographical locations. As part of the NDBC's network, these stations have a track record of providing consistent and reliable data. The Standard Meteorological Data of buoys from NDBC contains critical meteorological and oceanographic data for analyzing and forecasting weather events, available in both historical and real-time data. The dataset is presented as follows:

| #YY | MM | DD | hh | mm | WDIR | WSPD | GST | WVHT | DPD | APD | MWD | PRES | ATMP | WTMP | DEWP | VIS | PTDY | TIDE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| #yr | mo | dy | hr | mn | degT | m/s | m/s | m | sec | sec | degT | hPa | degC | degC | degC | nmi | hPa | ft |
| 2014 | 09 | 11 | 16 | 50 | 120 | 5.0 | 6.0 | 0.6 | 6 | 4.2 | 134 | 1016.5 | 29.3 | 30.5 | 24.4 | MM | +0.3 | MM |

Fig. 2. The Dataset

Here's a breakdown of what each column represents and their units.

| | |
|---|---|
| YY | Year |
| MM | Month |
| DD | Day |
| hh | Hour |
| mm | Minute |
| WDIR | Wind Direction (degrees from true north) |
| WSPD | Wind Speed (meters per second) |
| GST | Gust (meters per second) |
| WVHT | Wave Height (meters) |
| DPD | Dominant Wave Period (seconds) |
| APD | Average Wave Period (seconds) |
| MWD | Mean Wave Direction (degrees from true north) |
| PRES | Atmospheric Pressure (hectopascals) |
| ATMP | Air Temperature (degrees Celsius) |
| WTMP | Water Temperature (degrees Celsius) |
| DEWP | Dew Point (degrees Celsius) |
| VIS | Visibility (nautical miles) |
| PTDY | Pressure Tendency (hectopascals) |
| TIDE | Tide (feet) |

Thirteen buoy stations strategically located around the California coast were chosen for our special attention in this area, ranging from the north to the south: ST Georges (station ID: 46027), Eel River (station ID: 46022), Arena (station ID: 46014), Bodega Bay (station ID: 46013), San Francisco (station ID: 46026), Monterey (station ID: 46042), Cape San Martin (station ID: 46028), San Maria (station ID: 46011), West Santa Barbara (station ID: 46054), South Santa Rosa (station ID: 46069), East Santa Barbara (station ID: 46053), Santa Monica Basin (station ID: 46025) and San Clemente Basin (station ID: 46086). These stations were chosen based on their geographical dispersion, data consistency and completeness, and relevance to the coastal areas most affected by high-impact meteorological events.

The timeframe for the historical Standard Meteorological Data spans from 2014 to 2023. This duration was selected to ensure a comprehensive understanding of weather patterns, encompassing seasonal changes and rare meteorological events. These historical data can be directly downloaded from the NDBC website. The extensive historical data serves as a solid base for developing machine learning models capable of accurately predicting weather. For real-time forecasting, access to real-time data is made possible by the NDBC's open data link, which ensures that forecasts are as up-to-date and accurate as possible. The Latest Observations File of Standard Meteorological Data offers updates every five minutes, supplying a continuous stream of each station's data. This high-frequency updating allows for the incorporation of the most current conditions into the predictive models, enhancing their relevance and timeliness.

#### 3.1.2 The Storm Events Database

The extreme weather events dataset was collected through the Storm Events Database in NOAA. All storm events data of California from 2014 to 2023 was downloaded from the database. The Storm Events Database is a comprehensive repository for recording various storm events. The storm events contain various types, including but not limited to tornadoes, thunderstorm winds, flash floods, heatwaves, and tropical storms. Storm events are geographically organized based on either the County or NWS Forecast Zone, depending on the event scale and impact. The information for each storm event data record consists of the event date, time, location, affected areas, and the loss caused by the event. The database is updated when there is a storm event occurring in the United States, to ensure that it provides the most recent and accurate information of storm events.

In this project, all storm events data of California from 2014 to 2023 was downloaded from the database. We selected the storm events that occurred in the Counties where the thirteen buoy stations described in part 3.1.1 are located. The selected event data is downloaded as a CSV file. There are 39 columns in this database, including the columns representing the storm event's location, date, time, and event type. The begin and end times are represented as xxyy or xyy where x represents the hour and yy represents the minute. Since the occurrence of each type of storm event in the past 10 years on the California coast is very rare, the count of each type of storm event in this dataset is too small to precisely predict. We decided to predict the occurrence of extreme weather events rather than a specific storm event. We gathered the buoy data for every storm event and used the buoy data as feature value and the occurrence of storm events as a target for training the classifier models to predict whether there will be an extreme weather event occurring.

### 3.2 Exploratory data analysis

Our EDA highlighted daily variations in parameters such as wind speed, peak gust speed, sea-level pressure, wave height, dominant wave period, average wave period, and

dominant wave direction. By comparing the daily average wind speed in September 2023, the lowest wind speed of these 13 stations is around 1.8 m/s which is from Santa Monica Basin (station ID: 46025), and the highest wind speed is around 12.5 m/s from ST Georges (station ID: 46027). Overall, the wind speed at ST Georges is the highest in all observation stations, while the wind speed at Santa Monica Basin is the lowest. We found that wind speed and peak gust speed are highly correlated. The frequency and degree of the wind speed changes have a deep influence on the sea level pressure. There is also a strong correlation between the dominant wave period and the average wave period. The shape of trend lines for both the average daily dominant wave period and average daily average wave period are very similar, and they are also analogous to the trend line for average daily wave height. The trends in the data could be indicative of bigger weather patterns or seasonality that should be investigated further with larger datasets. We also found anomaly observations revealed several outliers in the dataset. We used a 7-day moving average line to detect the unusual daily average wind speed for each station. For example, two anomalies of daily average wind speed were detected in September 2023 at ST Georges (station ID: 46027), which occurred on 2023-9-13 and 2023-9-20. We used the same method to detect the anomalies of daily average wave height. For example, one anomaly of daily average wave height occurred in September 2023 at ST Georges, which occurred on 2023-9-16. In September 2023, no anomaly was detected in PT Arena (station ID: 46014), Bodega Bay (station ID: 46013), San Francisco (station ID: 46026), Cape San Martin (station ID: 46028), San Maria (station ID: 46011), West Santa Barbara (station ID: 46054), and East Santa Barbara (station ID: 46053). The detected anomalies are critical as they might correspond to extreme weather conditions or data-collection anomalies.

The heatmap visualization of the correlation matrix was particularly important, providing a clear picture of how variables are related to each other as shown in the Fig3. Important findings included strong correlations between wind speed and peak gust speed, as well as complex relationships among wave characteristics. This visual exploration helped uncover crucial patterns and connections within the dataset, enhancing our grasp of its intricate interactions.

After examining the correlations using a heatmap, we carried out a time series analysis to uncover changes in various weather elements over time. Using a line plot, we visualized the trends of important parameters throughout the years, revealing their yearly patterns. This graphical representation helps us identify any noticeable trends, seasonal variations, or unusual behavior in the dataset, assisting in spotting patterns that could impact extreme weather events. The labels on the plot distinguish each feature and provide a clear picture of how these meteorological factors change over the span of ten years.This time series analysis complements our previous work with correlation heatmaps by providing an insightful view into temporal fluctuations within selected weather features in Fig4.

### 3.3 Historical Data Processing

In this project, the data collected from the Buoy Database was used as the weather characteristics which were regarded as the feature values, and the data collected from the Storm Events Database was used as the target for the prediction models.

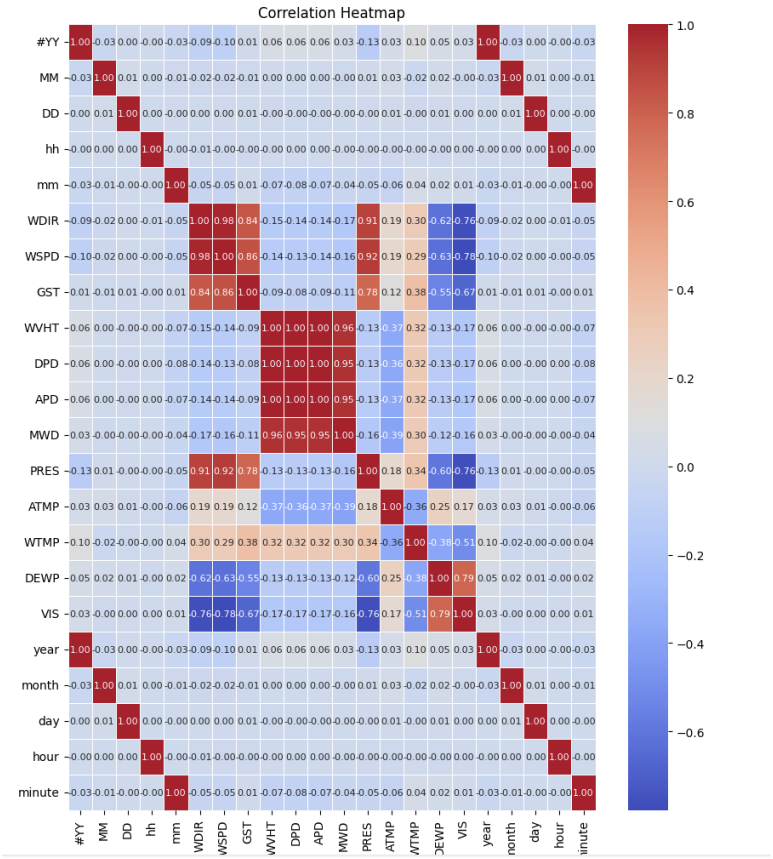


Fig. 3. The Correlation Heatmap

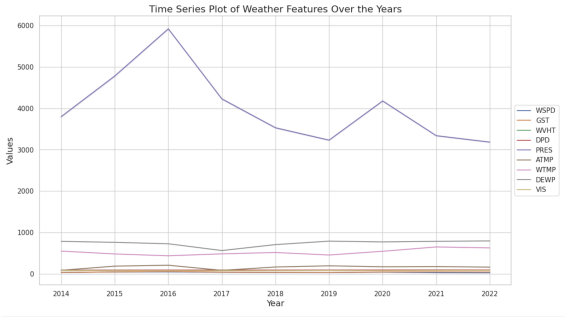


Fig. 4. The Lineplot

By checking the data quality of the buoy dataset, there were a large number of missing values in several columns. For instance, 2.83 million out of 3.11 million values in the ”WVHT” column are missing, with similar situations observed in columns ”DPD,” ”APD,” and ”MWD.” Some columns exhibit missing value patterns such as 99.00, 999, 999.0, or 99.0. To handle these missing value patterns, each column in the dataset underwent examination, and missing value patterns were replaced with ”NaN.” The number of missing values in each column was counted, setting a threshold percentage of 60 for excessive missing values. Columns exceeding the threshold were dropped, and subsequently, rows containing any missing values were also

eliminated. By checking the data quality of the storm events dataset, we did not find any missing values in the remaining columns.

The most challenging part of historical data processing was merging the buoy dataset and the storm events dataset to generate a large combined dataset to train, test, and evaluate the prediction models. Since the buoy dataset was collected by buoy stations while the storm events dataset was geographically organized by the County, we needed to match the locations for these two datasets manually. Another piece of information we needed to transform is date and time. Since the buoy data was updated every 5 minutes, the time values in the buoy dataset are every hour at 5 minutes, 10 minutes, 15 minutes, and so on. While the time in the storm events dataset was the exact time of the event occurrence, the majority of them were not at 5 minutes. So we transformed the time in the storm events dataset to the nearest hour (the transformed time should be earlier than the actual occurrence time) to match the time in the buoy dataset. Finally, the buoy dataset and the storm events dataset were merged by location and time.

After the historical data processing, the generated dataset contained 1,678,318 rows and 6 columns, with five features and a target. The figure below shows the processed data.

| | WDIR | WSPD | GST | PRES | ATMP | EVENT_TYPE |
|---|---|---|---|---|---|---|
| **0** | 157.0 | 5.2 | 9.4 | 1017.2 | 13.2 | yes |
| **1** | 160.0 | 5.6 | 8.5 | 1018.0 | 13.3 | yes |
| **2** | 192.0 | 6.3 | 9.0 | 1020.9 | 13.9 | yes |
| **3** | 206.0 | 1.5 | 1.7 | 1004.6 | 14.2 | yes |
| **4** | 134.0 | 2.4 | 3.1 | 1003.3 | 15.8 | yes |
| **...** | ... | ... | ... | ... | ... | ... |
| **1677863** | 317.0 | 5.8 | 8.6 | 1002.8 | 10.9 | no |
| **1677864** | 320.0 | 7.1 | 8.8 | 1002.9 | 10.8 | no |
| **1677865** | 312.0 | 7.2 | 10.6 | 1003.0 | 10.7 | no |
| **1677866** | 317.0 | 7.9 | 11.1 | 1003.1 | 10.9 | no |
| **1677867** | 187.0 | 2.2 | 2.2 | 1018.0 | 13.6 | no |

1678318 rows × 6 columns

Fig. 5. The Processed dataset

### 3.4 Model Development

Three machine learning algorithms, including Random Forest, Gradient Boosting, and XGBoost, were used for training models to predict the occurrence of extreme weather events. The chosen algorithms used ensemble methods, which were suitable for this project due to their nature of combining individual models to increase accuracy and reduce overfitting. Since the number of records has decreased to over one thousand after undersampling, it was essential to choose an algorithm that helps reduce overfitting.

Random Forest Classification is an ensemble machine learning algorithm that uses a majority vote of the predictions of multiple individual decision trees to improve the accuracy and other performance of the overall prediction. Random Forest model builds several decision trees (bootstrap samples) during training, each based on a random subset of features and a random subset of training data. The algorithm employs a bagging technique in which each decision tree is trained on a different subset of training data, drawn with replacement. Each decision tree randomly selects a subset of features to split the data. The randomness for selecting features and training data enhances the diversity among trees and reduces overfitting, which is more capable of generalizing well to unseen data.

Gradient Boosting Classification is another ensemble machine learning algorithm that can handle outliers and noisy data by its sequential learning structure. The gradient Boosting model is an additive model, which builds a series of decision trees sequentially and each tree corrects the errors of the previous tree. Each tree built by the Gradient Boosting model is a weak learner that does not need to have high accuracy. The learning rate is an important parameter of the Gradient gradient-boosting algorithm that controls the contribution of each tree to the final overall prediction. A gradient-boosting model with a low learning rate needs more trees but usually leads to better performance and generalization. It often employs regularization techniques such as tree depth limitation and shrinkage to prevent overfitting.

XGBoost (Extreme Gradient Boosting) is an advanced implementation of Gradient Boosting Classification which optimizes the model efficiency, scalability, and performance. XGBoost employs regularization techniques in its objective function to avoid overfitting, including using a learning rate to control the contribution of each tree to the overall prediction and limiting the tree depth by pruning the unnecessary subtrees. An important feature of XGBoost is its ability to parallelize the training process, which improves the efficiency of the model compared with traditional gradient-boosting models. Another improvement of XGBoost is that it can handle missing values by a built-in mechanism without data preprocessing operation.

The feature variables of the prepared dataset include `WDIR`, `WSPD`, `GST`, `PRES`, and `ATMP`, while the target variable is `EVENT_TYPE`. The data was divided into training and testing sets using an 80:20 ratio. After the splitting, undersampling was performed due to an imbalance of the target class. The number of records in the `yes` class in the target variable was 900, which was significantly lower than the number of records in the "no" class, which had over one hundred thousand records. Undersampling reduced the number of records in the majority class and helped balance the data.

PySpark Machine Learning Library (MLlib) was used to implement the modeling for streaming data. The resampled feature and target columns were concatenated back to prepare for Spark implementation, and training data and test data were converted into PySpark dataframe respectively. A PySpark feature transformer VectorAssembler was used to create feature vectors. After that, the StringIndexer function was used to encode the categorical values in the target column into numerical values, with 0 representing "no" and 1 representing "yes". The figure below shows the PySpark processed dataframe. The features were combined into the "features" column and the `labelIndex` column was the

target.

PySpark classification algorithms RandomForestClassifier, GBTClassifier, and SparkXGBClassifier were used to build random forest, gradient boosting, and XGBoost models respectively. After random forest, gradient boosting, and XGBoost models were developed, the models were trained using the training dataset and evaluated using the testing dataset. By tuning the hyperparameters of each model, the model's accuracy and performance were improved. For the random forest model, the hyperparameters were the number of decision trees and the maximum depth of each tree. We found the value of the decision tree counts to be 40 and the value of the maximum depth to be 3 which optimized the performance of the random forest model on our dataset. For the gradient boosting model, the hyperparameter was the number of iterations. We found the value of the number of iterations to be 50 which optimized the performance of the Gradient Boosting model in this project. For the XGBoost model, since we had preprocessed the data, with no missing values remaining, we could ignore the missing value hyperparameter.

| WDIR | WSPD | GST | PRES | ATMP | EVENT_TYPE | features | labelIndex |
|---|---|---|---|---|---|---|---|
| 144.0 | 1.4 | 3.8 | 1013.3 | 14.6 | no | [144.0,1.4,3.8,10... | 0.0 |
| 227.0 | 0.8 | 1.1 | 1017.6 | 10.8 | no | [227.0,0.8,1.1,10... | 0.0 |
| 215.0 | 2.1 | 3.4 | 1011.7 | 15.5 | no | [215.0,2.1,3,4,10... | 0.0 |
| 289.0 | 1.9 | 5.7 | 1018.6 | 11.4 | no | [289.0,1.9,5.7,10... | 0.0 |
| 325.0 | 4.9 | 6.6 | 1018.1 | 12.9 | no | [325.0,4.9,6.6,10... | 0.0 |
| 198.0 | 1.5 | 1.7 | 1021.2 | 10.8 | no | [198.0,1.5,1.7,10... | 0.0 |
| 349.0 | 3.4 | 4.5 | 1023.6 | 12.0 | no | [349.0,3.4,4.5,10... | 0.0 |
| 29.0 | 5.2 | 6.3 | 1025.4 | 12.2 | no | [29.0,5.2,6.3,102... | 0.0 |
| 263.0 | 1.9 | 4.4 | 1017.8 | 14.6 | no | [263.0,1.9,4.4,10... | 0.0 |
| 279.0 | 1.5 | 2.9 | 1015.6 | 13.5 | no | [279.0,1.5,2.9,10... | 0.0 |
| 260.0 | 3.4 | 5.0 | 1013.5 | 18.8 | no | [260.0,3.4,5.0,10... | 0.0 |
| 256.0 | 2.9 | 4.3 | 1012.2 | 14.8 | no | [256.0,2.9,4.3,10... | 0.0 |
| 312.0 | 8.6 | 10.6 | 9999.0 | 11.2 | no | [312.0,8.6,10.6,9... | 0.0 |
| 319.0 | 12.2 | 14.7 | 1014.4 | 10.7 | no | [319.0,12.2,14.7,... | 0.0 |
| 248.0 | 3.9 | 6.4 | 1008.9 | 11.9 | no | [248.0,3.9,6.4,10... | 0.0 |
| 144.0 | 0.7 | 2.3 | 1011.9 | 22.1 | no | [144.0,0.7,2.3,10... | 0.0 |
| 191.0 | 2.2 | 3.1 | 1019.5 | 14.4 | no | [191.0,2.2,3,1,10... | 0.0 |
| 254.0 | 4.6 | 8.6 | 1015.6 | 14.9 | no | [254.0,4.6,8.6,10... | 0.0 |
| 242.0 | 2.9 | 4.6 | 1017.6 | 14.1 | no | [242.0,2.9,4.6,10... | 0.0 |
| 321.0 | 2.1 | 3.4 | 1019.9 | 10.3 | no | [321.0,2.1,3,4,10... | 0.0 |

Fig. 6. The PySpark processed dataframe

### 3.5 Implementation of Real-Time Prediction

Upon completion of the training phase, the team evaluated the performance of each model. The model with the highest accuracy was then chosen for deployment. In order to utilize the model in a real-time environment, this model was saved to a local directory using the 'save' API offered by 'pyspark.ml.Pipeline'. This procedure ensured the model could be easily accessible and used for prediction.

The consistency of data structure is critical for applying a pre-trained model to real-time prediction. Hence, the format and schema of the real-time streams should match the historical data used in model training. A Kafka producer is in charge of cleaning and preprocessing real-time streams during the data ingestion phase. It transforms and filters the massive stream of data, handling the missing value, and keeping only the necessary data we are interested in. This filtered and cleaned data is then sent to a Kafka topic as a JSON string, which serves as a messaging queue.

A Kafka Consumer receives and loads the JSON messages once they reach the Kafka topic, and converts the JSON objects into batch DataFrames, which is the pre-trained model's needed input format. This conversion is crucial because it ensures that the data is ready for prediction analysis.

With the real-time data structured appropriately, the system proceeds to load the pre-trained model from the local path utilizing the 'load' API from the 'pyspark.ml.Pipeline'. The model is then applied to the real-time data. This application is the final step in the prediction pipeline, where the model processes the live data and outputs predictions. These predictions are then used to inform about the likelihood of upcoming severe weather events, completing the real-time forecasting cycle.

### 3.6 Evaluation Metrics

All the models are evaluated with metrics including accuracy score, recall, precision, and F1 score. The accuracy score is the ratio of the number of correct predictions over the total number of predictions. In general, an accuracy score that is greater than 70 is the threshold to be considered a good model. Precision is the ratio of the number of true positives over the total number of positives. Recall is the ratio of the number of true positives over the sum of true positives and false negatives. The F1 score is the ratio of 2 times the multiplication of precision and recall over the summation of precision and recall. The F1 score is an important metric in this project, where the incidence of extreme weat

## 4 Experimental Result

### 4.1 Performance of the Predictive Models

The team used a historical testing dataset to evaluate the models. Random forest has an accuracy score of 0.9169, recall of 0.8474, precision of 0.9992, and f1 score of 0.9169. Gradient boosting has an accuracy score of 0.8911, recall of 0.8044, precision of 0.9994, and f1 score of 0.8911. XGBoost has an accuracy score of 0.9060, recall of 0.8290, precision of 0.9994, and f1 score of 0.9060. The results are shown in the chart below.

| Model | Random Forest | Gradient Boost | XGBoost |
|---|---|---|---|
| Accuracy | 0.9169 | 0.8911 | 0.906 |
| Recall | 0.8474 | 0.8044 | 0.829 |
| Precision | 0.9992 | 0.9994 | 0.9994 |
| F1 Score | 0.9169 | 0.8911 | 0.906 |

TABLE 1
Table 1. Model Metrics Evaluation

### 4.2 Comparison of Model Results

Out of the three models, the random forest has the highest accuracy score, recall, and F1 score. Overall, the random forest has the best performance when the depth of the tree is set to 3 and the number of trees is set to 40. With these hyperparameter settings, a random forest is not sensitive to noise, and it can generalize unseen data. Giandent Boosting has the lowest accuracy score, recall, and F1 score, and this may be due to the fact that the sequential trees are correcting the previous one, making it more prone to overfitting. It is interesting to find that all models have similar high precision, indicating that all models can make accurate predictions about positive events. Since the goal of the project is to predict whether there will be an extreme weather event, it is important to have high performance in classifying the positive events.

### 4.3 Analysis of Real-Time Prediction Capabilities

The effectiveness of the trained models was tested by seeing how well they worked with real-time data. Every five minutes, new data was sent by the producer and immediately processed by the consumer. This quick process ensured that the predictions were based on the latest information. The models gave a forecast that showed both the likelihood of a weather event happening and a straightforward yes or no prediction.

When looking at the predictions, the models proved to be quite successful. They were particularly good at giving early warnings for severe weather events. Some of the most impressive forecasts involved detecting sudden storm surges and quick increases in wind strength, which are crucial for taking early safety measures. The high rate of successful forecasts shows that these models could be very useful for weather forecasting and helping people prepare for disasters. By providing the chance of certain weather events happening, the system allows people to understand and act on these predictions effectively. The ability to quickly turn real-time data into reliable forecasts represents a big step forward in predicting weather events.

Below is an example of the results from 13 stations:

| WDIR | WSPD | GST | PRES | ATMP | probability | prediction |
|---|---|---|---|---|---|---|
| 310.0 | 8.0 | 10.0 | 1018.2 | 15.4 | 0.28295528416307675 | 0.0 |
| 330.0 | 4.0 | 5.0 | 1019.4 | 14.4 | 0.3093080747049778 | 0.0 |
| 350.0 | 6.0 | 7.0 | 1020.2 | 14.1 | 0.29826484162227107 | 0.0 |
| 350.0 | 7.0 | 9.0 | 1022.6 | 13.2 | 0.3143281837151583 | 0.0 |
| 290.0 | 4.0 | 5.0 | 1016.2 | 17.6 | 0.28089443894267196 | 0.0 |
| 310.0 | 1.0 | 2.0 | 1019.7 | 15.4 | 0.24955256522482916 | 0.0 |
| 340.0 | 5.0 | 7.0 | 1023.0 | 12.9 | 0.31003743501318853 | 0.0 |
| 340.0 | 8.0 | 10.0 | 1017.3 | 15.4 | 0.3658174074831528 | 0.0 |
| 340.0 | 5.0 | 7.0 | 1019.1 | 14.8 | 0.3093080747049778 | 0.0 |
| 270.0 | 2.0 | 3.0 | 1014.9 | 16.9 | 0.3412338204278479 | 0.0 |
| 310.0 | 9.0 | 11.0 | 1015.5 | 16.2 | 0.4258387917944176 | 0.0 |
| 310.0 | 8.0 | 10.0 | 1015.3 | 16.2 | 0.4258387917944176 | 0.0 |
| 260.0 | 4.0 | 5.0 | 1014.6 | 18.1 | 0.31273490920463176 | 0.0 |

Fig. 7. Example of results from 13 stations

## 5 Conclusion and Future Work

### 5.1 Summary of Findings

This project successfully combines the power of big data streams, data analytics, and machine learning in the field of meteorology. It targets the accuracy of storm and weather pattern predictions along the California coasts. The primary data source provides a rich repository of oceanic and atmospheric data that is pivotal in developing our predictive system. The results and findings of this project are insightful. We successfully developed models that perform a significant accuracy of extreme weather predictions by implementing different models such as Random Forest, Gradient Boosting, and XGBoosting. With the comparisons between various models, we discovered the best performance model is Random Forest based on the ten-year historical dataset. The accuracy of the Random Forest model resulted in percentage of 92.01 which is higher than the Gradient Boosting which is in percentage of 88.43 and XGBoosting resulted in percentage of 89.61. These models can be successfully applied in the real-time extreme weather prediction system and generate reliable outcomes. The project has shown the potential to provide timely and reliable forecasts, which are crucial for effective disaster preparedness and response strategies.

### 5.2 Limitations of the Study

Despite the outstanding performance of the model, the study encountered some limitations. A significant constraint of the predictive models developed is their limited lead time in forecasting extreme weather events. The model can predict such events only about an hour in advance which may not always be able to provide sufficient time for effective emergency responses or evacuations. Another notable limitation pertains to the reliance on data from NDBC. Although NDBC provides a comprehensive dataset, the integrity of the dataset may not be guaranteed. Due to this reason, only a few stations with integrity datasets can be used in the model. To cover all the areas along the California coasts, we need more reliable data from buoy stations. This uneven distribution can affect the overall efficacy and reliability of our predictive models, especially in less monitored oceanic regions.

### 5.3 Recommendations for Future Work

Looking ahead, there are several avenues for further research that could build upon the findings of this study. Future research could explore integrating additional data sources to enrich the models. Diversifying data inputs could help mitigate some of the biases and limitations encountered in this study. Additionally, exploring other machine learning and data processing techniques could further refine the predictive models. The implementation of newer, perhaps more experimental, algorithms could enhance model accuracy and efficiency. Lastly, there is an opportunity to expand the scope of these models to include other geographical areas or different types of extreme weather events. Such expansion could not only validate the models in varied contexts but also contribute to a broader understanding and preparedness for weather-related challenges globally. Subsection text here.

## 6 Acknowledgement

We would like to extend our sincere thanks to Professor Dr. Vishnu Pendayala for his consistent support and expert advice, which has been crucial in the successful completion of this project. His invaluable help has significantly contributed to our achievements, and we are grateful for his dedication to our academic pursuits.

## APPENDIX A

### Code Walkthrough

The presentation will go through all the code used in this project and clearly and concisely explain the core functionality of the code. The live demo provides a thorough understanding by leading the audience through the code's logic and structure, explaining the purpose of key functions and algorithms, and how they contribute to the overall project.

Description of the code files:

- DATA228_Project_EDA.ipynb: This file presents the exploratory data analysis process for the raw datasets. The data pattern and significant characteristics are generated by this notebook file.
- DATA228_Project_ML_Models.ipynb: This file consists of historical data preprocessing and machine learning models developing. The raw datasets are loaded, cleaned, and transformed into formats that can be used to train machine learning models. Three machine learning classification models such as Random Forest, Gradient Boosting, and XGBoost are generated using PySpark Machine Learning Library, trained, and evaluated by the preprocessed dataset.
- DATA228_Project_ML_Models (for local use).py: This Python file converts three machine learning classification models for local use.
- DATA228_Project_Producer.ipynb: This file represents the Kafka producer. Real-time data is obtained, cleaned, and preprocessed by this file. The Kafka producer sends cleaned data as a JSON string to a Kafka topic as a message queue.
- DATA228_Project_Consumer.ipynb: This file represents the Kafka consumer. It converts JSON messages into batch DataFrames, preparing data for the prediction models. Pre-trained models loaded from local storage and applied to structured real-time data.

### Presentation Skills (Includes Time Management)

The total length of the whole presentation will be strictly controlled within 30 minutes: 5 minutes of study, 5 minutes of significant paper research, 15 minutes of project presentation, and 5 minutes of Q&A and discussion. Everyone in the group will present some parts and be fully prepared. The presenter will maintain eye contact and use appropriate body language to engage the audience.

### Discussion / Q&A

After each section of the presentation, there will be a Q&A or discussion time, in total of 5 minutes. Everyone in the team will make sure to fully understand the content and prepare for potential questions. During the Q&A section, presenters will listen to questions attentively and provide direct and thoughtful answers.

### Demo

During the presentation, we will give a live demo. It shows how the system works step by step, with a full explanation of code and actions. It highlights the project's unique value proposition and shows the application in action, providing clear evidence of its functionality. The project pitch video is generated by a generative AI tool - simpleshow video maker. Link to simpleshow video maker:
https://videos.simpleshow.com/Hxry2unAPt
The link for the video in YouTube:
https://www.youtube.com/watch?v=cv7xCX-4OAY

### Report

The report uses the official IEEE template https://www.overleaf.com/latex/templates/ieee-demo-template-for-computer-science-journals/fxrxvtcsjqmm. By editing with LaTeX, the format of this paper strictly follows the official IEEE format. The paper is divided into five sections: introduction, the architecture of the system, methodology, experimental result, and conclusion, which provide sufficient and comprehensive information and explanation about the project. The paper is written in a research paper tone. Before submitting, we ran the detector of plagiarism and AI. The paper only contains the necessary figures to support the explanation of the system architecture and results.

### Version Control

Git is used as the version control tool. The team performed pull requests, and the code was reviewed before merging. The link can be accessed at:
https://github.com/jliang15/data228
Link to sample commit screenshots:
https://docs.google.com/document/d/1yIteOqnHk1MZ9X2aVOAqayhS7ujIwOuLQJXPaFb32o8/edit?usp=sharing

### Lessons Learned

By doing this big data stream analysis project, we have learned several valuable lessons. The details are included in both the report and presentation slides. The quality and the coverage of datasets play a crucial role in predictive modeling. The selection of buoy stations along the California coast highlighted the need for a careful balance between geographic representation and data reliability. The project also provides us with practical experience in handling and processing large real-time datasets using Apache Kafka and Apache Spark. We have good practice in collecting and processing large-scale data with PySpark dataframe and implementing machine learning models on real-time data to make predictions via PySpark Machine Learning Library (MLlib). Also, since the project is a team project, it is essential for every one of us to learn interdisciplinary collaboration.

### Prospects of Winning Competition / Publication

This project designed and developed an innovative application that has the ability to collect and process real-time data from the National Data Buoy Center, which is available every five minutes. With real-time data, the application can make accurate predictions about whether an extreme weather event will happen in a timely manner, making it a vital application with the potential to save lives.

### Innovation

The innovation of this project is that we integrate real-time data processing with advanced machine learning for extreme weather event predictions. The project uniquely combines streaming technologies like Kafka and Spark Streaming with sophisticated algorithms such as Random

Forest, Gradient Boosting, and XGBoosting to stream and process data from the buoy station every five minutes. This approach harnesses a comprehensive range of data from the National Data Buoy Center. It represents a significant advancement in the prediction of extreme weather events, characterized by improved accuracy and timeliness.

### Teamwork

Weekly meetings were held to update progress and set future goals. The team used Colab and Git for collaboration in development. We used a Gantt chart to split the project work tasks and assigned them to every member of our team with due dates. Sample Colab collaboration and the Gantt chart show as follows. Link to teamwork screenshots:
https://docs.google.com/document/d/1yIteOqnHk1MZ9X2aVOAqayhS7ujIwOuLQJXPaFb32o8/edit?usp=sharing

### Technical Difficulty

Throughout this project, we faced several technical challenges, including managing the complexity and volume of data from the National Data Buoy Center. To address this challenge, bid data streaming technologies like Apache Kafka and Spark were applied. Additionally, to achieve a balance between the accuracy of our predictive models and providing sufficient lead time for practical use, we used stratified sampling and under-sampling to obtain balanced training and

### Practiced pair programming

Pair programming technique is used in building, testing, and evaluating the prediction models. Jiayi Liang and Ying Liu worked together on one notebook via a collaborative real-time editor Google Colab, and also used GitHub to perform version control. Initially, Ying Liu played a role as a driver in writing the initial codes for training and test data preparation and building prediction models using scikit-learn library. Jiayi Liang as an observer reviewed Ying's codes, provided the improvement for the codes, and came up with some ideas for adjusting the prediction models for streaming data. Then, they switched roles, with Jiayi as the driver and Ying as the observer. Jiayi switched the prediction models using PySpark MLlib. Ying reviewed the codes, converted the data format to spark data frame, and came up with some ideas for evaluating the models' performance. Jiayi and Ying switched roles of driver and observer several times for pair programming to complete building, testing, and evaluating the prediction models. Link to Screenshot for pair programming via Google Colab and GitHub:
https://docs.google.com/document/d/1yIteOqnHk1MZ9X2aVOAqayhS7ujIwOuLQJXPaFb32o8/edit?usp=sharing
Link to Google Colab:
https://colab.research.google.com/drive/1XpIFsvSgU5R2Nb-rWFCIS79iyVGv_Gx1#scrollTo=d899ef67
Link to GitHub:
https://github.com/jliang15/data228/blob/main/data228project_ml_models.ipynb

### GitHub Copilot Experience

GitHub Copilot is an AI-powered code completion tool, which is built on the GPT architecture. We installed GitHub Copilot in the IDE Visual Studio Code and felt it is very helpful for generating code snippets for models, providing coding suggestions, and improving efficiency for coding. The first experience is that users can chat with GitHub Copilot by providing some outlines. GitHub Copilot will give some coding suggestions. For example, we asked GitHub Copilot how to adjust the Random Forest classification model for data in PySpark Dataframe, and GitHub Copilot gave the sample codes for our requirement automatically.

The second experience is that GitHub Copilot gives users suggestions when they are writing code. It helps improve users' efficiency and accuracy in writing code. For example, when we are writing the code for the XGBoost classification model, GitHub Copilot provides some suggestions for our coding immediately, which helps us write code much faster than before.

The third experience is that GitHub Copilot can convert the comments to code. Users write comments in the code file, and then GitHub Copilot writes code to complete users' comments. For example, if we write a comment "write performance metrics for xgboost", GitHub Copilot automatically completes our explanation. Link to Screenshot for GitHub Copilot Experience:
https://docs.google.com/document/d/1yIteOqnHk1MZ9X2aVOAqayhS7ujIwOuLQJXPaFb32o8/edit?usp=sharing

### Practiced agile / scrum

This project used Trello as the project management tool to monitor the progress of the project. We held a weekly meeting every Friday to report the progress of the project, pose questions, and solve problems together. We also listed "To dot", "Doing", and "Done" task lists to make our project more efficient and clear to understand the progress. We used Trello to save our weekly meeting minutes. The tasks in Trello are defined by our Gantt chart (in Appendix A Teamwork). Link to the Trello workspaces:
https://trello.com/b/wJIqEqaZ/data228-extreme-weather-forecasts-project
Link to Screenshots for meeting minutes:
https://docs.google.com/document/d/1yIteOqnHk1MZ9X2aVOAqayhS7ujIwOuLQJXPaFb32o8/edit?usp=sharing

### Used Grammarly / other tools for language

Quilbot is used as a language tool. It is a Google Chrome Extension that allows highlighting text and making grammatical suggestions. The screenshots below show examples of how it helps the team make corrections in grammar. Link to Screenshots for Grammarly:
https://docs.google.com/document/d/1yIteOqnHk1MZ9X2aVOAqayhS7ujIwOuLQJXPaFb32o8/edit?usp=sharing

### Slides

Our PowerPoint presentation has been designed to comprehensively cover all aspects of our project. Each slide

is crafted to convey key points effectively, ranging from the initial concept and data collection methods to the intricacies of our predictive models and the technical challenges we faced. The presentation is structured to guide viewers through the journey of our project. It highlights our innovative approach, detailing the methodologies and technologies we utilized. The results we achieved are clearly presented, showcasing the effectiveness of our models. Additionally, the presentation covers the lessons learned along the way, reflecting on both the successes and challenges of the project.

### Used LaTeX

LaTex is used for formatting the report. Overleaf template is used to implement LaTex.The link can be found here:
https://www.overleaf.com/latex/templates/ieee-demo-template-for-computer-science-journals/fxrxvtcsjqmm
And DATA228_Project_Report(latex).tex is the file we used to generate the project report.

### Used creative presentation techniques

Prezi is used to develop the presentation slides. The team used "Ask AI" feature in Prezi to help make the bullet points. The figures below are an example of how the team used "Ask AI" to turn a paragraph into bullet points. Link to Screenshots for AI used Prezi:
https://docs.google.com/document/d/1yIteOqnHk1MZ9X2aVOAqayhS7ujIwOuLQJXPaFb32o8/edit?usp=sharing

### Literature Survey

This project is built upon our significant research paper and other papers. The significant paper talks about the application of the system using Kafka and Spark Streaming. Other paper includes the usage of the NDBC dataset and machine learning models for this project. We have listed the details of the literature survey in part 1.4.

## Appendix B: Author Contributions

### Xin Ling:

Xin Ling is tasked with identifying obstacles, choosing streaming data, and deciding on project features. Additionally, Xin Ling is crucial in preparing data by gathering buoy and event information and reformatting image and video data for modeling.

### Lumie Yang:

Lumie Yang's responsibilities include identifying potential advantages, reviewing and summarizing pertinent documents, and creating a project outline. During the data preparation stage, Lumie helped choose historical and real-time weather information while also assisting in eliminating ambiguous data.

### Ying Liu:

Ying Liu is responsible for identifying potential advantages, investigating applicable regulations, and contributing to the assessment phase by evaluating model validity and establishing economic viability. In terms of data preparation, Ying played a vital role in gathering extreme weather event data and integrating buoy information with extreme weather event datasets.

### Jiayi Liang:

Jiayi Liang's responsibilities include evaluating the current technology landscape, creating project plans, and adjusting hyperparameters to enhance models. During data preparation, Jiayi selected historical and real-time weather data while also eliminating noise, errors, and outliers from the dataset.

### Yamini Muthyala:

Yamini Muthyala is involved in identifying potential advantages, grasping the constraints of current methods, and establishing project timelines. During the modeling stage, her responsibilities include creating test designs for models, adjusting hyperparameters, and assessing system detection and recovery.

## References


[1] Chattopadhyay, A., Nabizadeh, E., & Hassanzadeh, P. (2020). Analog Forecasting of Extreme-Causing Weather Patterns using Deep Learning. *Journal of Advances in Modeling Earth Systems*, 12(2). *https://doi.org/10.1029/2019ms001958*

[2] Liu, Y. (2016, May 4). Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv.org*. *https://arxiv.org/abs/1605.01156*

[3] Lou, R., Wang, W., Li, X., Zheng, Y., & Lv, Z. (2022). Prediction of ocean wave height suitable for ship autopilot. *IEEE Transactions on Intelligent Transportation Systems*, 23(12), 25557–25566. *https://doi.org/10.1109/tits.2021.3067040*

[4] Nitu, C., Dobrescu, A. S., Krapivin, V. F., & Soldatov, V. (2019). Algorithm for Decision Making and Big Data Processing. *2019 22nd International Conference on Control Systems and Computer Science (CSCS). https://doi.org/10.1109/cscs.2019.00087*

[5] Racah, E. (2017). ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. *https://proceedings.neurips.cc/paperfiles/paper/2017/hash/519c84155964659375821f7ca576f095Abstract.html*

[6] Zhou, B., Li, J., Wang, X., Gu, Y., Xu, L., Hu, Y., & Zhu, L. (2018). Online Internet traffic monitoring system using spark streaming. *Big Data Mining and Analytics*, 1(1), 47–56. *https://doi.org/10.26599/bdma.2018.9020005*