UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

INVESTIGATING TECHNOLOGIES AND TECHNIQUES FOR FLOOD MONITORING AND DETECTING

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

By

ABDULLAH MOUSSELLI
Norman, Oklahoma
2022

INVESTIGATING TECHNOLOGIES AND TECHNIQUES FOR FLOOD MONITORING AND DETECTING

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Hazem H. Refai, Chair

Dr. Choon Yik Tang

Dr. Samuel Cheng

Dedication

To my beloved wife Seher
My mom and dad
Malak & Mahmoud
My beloved family
&
Friends.

ALL PRAISE IS TO THE ONE AND ONLY GOD.

Acknowledgements

Foremost, I would like to express my genuine gratitude and sincere thanks to my advisor Dr. Hazem Refai for his support and guidance. In addition, I would like to state my appreciation to my committee members—*Dr. Samuel Cheng* and *Dr. Choon Yik Tang*. Thanks to Michelle Farabough for helping me edit this thesis. I extend my gratitude to the OU-TULSA operations team for their help completing this work. Many thanks to my colleagues who participated in this research, especially Mohamat Irban bin Ali Kaja Najumudeen, Mohamed Ali Habib, and Mohamed E Afify. Finally, I want to recognize the support and great love of my family and friends.

Table of Contents

List of Tables

List of Figures

Abstract

The need for a flood alerting and forecasting system is becoming ever more critical, especially given the effects of increasing severe weather over the last ten years. Flash floods, particularly, are deadly because of the short time period in which they develop, as well as the difficulty in predicting their occurrence. Consequently, floods have become recognized as the most fatal natural disaster in the United States. Recent developments in artificial intelligence (AI) and machine Learning (ML) algorithms are promising for overcoming related issues by providing data driven solutions for more reliable and efficient forecasting.

The study reported in this thesis first explored market-available, industrial-based flood monitoring and alerting systems developed and deployed by several private companies and federal agencies. Weather conditions that accompany flash flood events were then analyzed as first steps in building an ML flood forecasting model. Weather attributes were analyzed both statistically and visually to uncover their importance for modeling. Subsequently, common ML classification techniques were compared with a variety of neural network architectures to determine the best model for forecasting flooding. Modeling results showed that a one-hour forecast XGBoost ensemble model outperformed others, with an 85% accuracy for forecasting a flood event and a 92% AUC score.

This thesis introduces a novel approach for evaluating non-contact flood detection technologies. An open-channel testbed was designed and constructed, and several non-contact water level and velocity technologies were evaluated under various conditions. Results showed that market available radar technology delivered the most accurate readings with an MAE score of approximately 2 cm and MAE of 0.1 m/s.

1 Introduction

Weather warning systems are designed to provide reliable, high quality data for alerting citizens of critical conditions in a timely manner. Such systems would enable sufficient reaction time for preventing or reducing loss of life and property. Amongst natural disasters, floods are most common. Several factors contribute to the occurrence of flash floods, with rainfall intensity and duration among two of the most critical key elements. Other factors include topography and soil condition.

Flood casualties could be affected by the availability or lack of information about an impending flood (e.g., type of flood, rain rate, drop size, cell size, weather conditions, and water level [or stage] and velocity. Providing such data is essential for building efficient, early-alerting systems. The goal is building robust and accurate neural network models for both short-term and long-term flood prediction.

Oklahoma's rainfall has proven variable from year to year, characterized by seasonal differences. Wintertime precipitation tends to be widespread and stratiform in nature [1]; summertime precipitation is almost convective in nature, resulting from thunderstorms. Autumn and spring offer both stratiform and convective precipitation. Oklahoma storms vary in severity, depending on rain duration and rate. One type is an Ordinary Cell [2], which forms from a single, small cell with high rain rate and short life. Typically, rainfall duration is short and rarely leads to floods. A second type is a Multi-Cell Cluster, which consists of many storm clusters. Cells have a varying amount of rainfall and can produce a significant amount of rainfall in small geographic areas. These systems form fast and travel slowly. Cells develop continuously, appear stationary, and lead to a significant amount of rainfall. A third type—Multicell Line—is the most common n Oklahoma. These systems consist of a squall line and can extend for hundreds of miles, producing huge amounts of rainfall and forming new cells. Flash floods are likely. A fourth type is recognized as a Supercell thunderstorm, characterized by a single cell that can persist for many hours Despite its small size, this storm system can produce extremely heavy precipitation with flash floods. On April 17, 2013, [3] Medicine Park, Meers, Chickasha, and Newcastle counties experience such a single supercell storm that caused between four and seven inches of rain in only two hours and resulted in flash flooding.

In this study, I assessed available technologies used for measure and collecting flood variables. Weather data from 15 Road Weather Information Systems (RWIS) deployed on road-bridge cross intersections along highway I-35 was utilized. Data was processed and analyzed during, before, and after historical flash flood events reported by the Oklahoma Department of Transportation (ODOT) and the National Weather Service (NWS). A short-term flood forecast model was enhanced by information about water level- and velocity-measuring technologies for enabling an extended forecast. Adding these flood variables resulted in a more efficient and generalized model for detecting all types of flash flood systems. An open channel testbed facilitated repeatable, experimental scenarios that improved the accuracy and suitability of technologies measuring water levels and velocities. Comparisons of Mean Absolute Error (MAE) scores were compared.

1.1 Thesis Objective

This thesis was developed with the objective of building an effective and efficient flash flood monitoring and alerting system. Several existing flood monitoring technologies and industrial flood alerting systems were initially reviewed. Weather attributes were then retrieved from RWIS systems deployed at different locations across Oklahoma’s I-35 highway system and analyzed for the prospect of building an ML flood forecasting model. Several non-contact technologies were considered to monitor water level and velocity. A novel evaluation approach for obtaining ground truth was used to determine the most reliable and accurate method for the proposed flash flood monitoring and altering system.

The main contributions of this thesis are summarized below:

[1] Provide a full description of existing flood monitoring technologies and alerting systems used by government agencies and leading companies positioned in the environmental monitoring field.

[2] Utilize weather data collected by RWIS stations to build short-term ML flood forecasting models.

[3] Discuss the methodology for designing and developing an open channel water testbed for evaluating non-contact water level and velocity measuring technologies.

[4] Introduce a novel approach for evaluating non-contact water surface technologies by emulating real-time water surface flow characteristics in a controlled lab environment.

As per the aforementioned stated objectives, this thesis is divided into multiple sections. Section 1 briefly introduces the objectives, work, and analyses undertaken for this thesis. Section 2 summarizes the background and a variety of related work investigating flood monitoring techniques and forecasting models. Section 3 details a number of industrial flood alerting and monitoring systems currently deployed by various companies and agencies throughout the United States. Section 4 showcases weather data analyses and their use for building an ML-based one-hour flash flood forecasting model. Section 5 discusses the process for designing and developing a testbed emulator to establish a ground truth for evaluating non-contact water surface technologies. Results are provided for several experiments conducted to assess evaluated technology accuracy. Finally, Section 6 summarizes the work completed for this thesis and suggests future work and improvements.

2 Background

This section features ongoing research and development in flood monitoring and alerting technologies, especially non-contact water level and water-surface velocity measuring techniques and flood forecasting methods. The thesis focuses on ML-based flood forecasting models and non-contact flood detecting technologies. A study in [4] built a risk assessment model using a Random Forest (RF) ML approach for determining each factor's contribution to a flood occurrence. Results showed that three-day precipitation, distance to the river, and stream power index had the highest risk scores based on the model, with about 71% accuracy. United Kingdom researchers in [5] utilized cumulative and rainfall intensity data to forecast the severity of flood occurrence. Data was classified into three categories, namely normal, abnormal, and severe. A RF classifier model outperformed others with an AUC score of approximately 85%. A literature review of ML applications designed for short-term and long-term flood predictions was conducted in [6]. Results in the journal article showcase the usage of methods like artificial neural networks (ANN), RF, support vector machine (SVM), and many other single and hybrid ML methods that have been used to build real-time, short-term (hourly), and long-term (monthly, seasonally) flood prediction models. Many models shown in [6] utilize factors like rain rate, drop size, river water level (stage), and streamflow (surface velocity or stream flow index). The first of these (e.g., rain rate and drop size) are important factors for predictions, as they affect the rising water level. In [7], an acoustic Disdrometer was used to measure drop size distribution. Sensor light rays determine rain size and speed. A series of signal conditioning and piezoelectric sensors were placed under the surface of a Zinc slab to prevent direct contact with the water and to increase surface area for data collection. Surface diameter was 13.7 cm with a surface area of 294,6733 cm^2. Notably, the sensor can read from 0.25 m to 6.5 mm. Rainfall was divided into two groups: 1) stratiform and 2) convective rain.

The stratiform travels less than 25 mm/h and lasts more than an hour, while the convective rain experiences speeds above 25 mm/h with a duration time of just a few minutes. Rain that falls to the ground is defined with intensity R, which is the number of points collected with diameter D within a one-hour interval. Total precipitation is dependent upon the speed and direction of raindrops on the surface of earth. The work shows that a rise in water level during rain varies greatly, depending on conditions. In one instance, there was a rise in river level of 0.01 meters per hour in stratiform rain and 0.4 meters per hour in convective rain.

Monitoring water level (stage) at creeks and rivers is crucial for early detection of floods; therefore, a continuous monitoring of surface level water is required. This can be accomplished using a variety of methods. Pressure sensors, ultrasonic sensors, radar sensors, optical sensors, and light detection and ranging (LIDAR) sensors are used most often. These methods provide accurate and reliable data for water level measurements. Using contact sensors (e.g., pressure sensors and pressure transducers) for water level measurements [8], [9] has major drawbacks, as these sensors are unable to independently measure static and dynamic pressures. Measurements would be affected by sensor orientation with respect to water flow. They can also be affected, by debris, rocks, and extreme weather conditions. It is important to know that sensors also require periodic maintenance and testing. Consequently, non-contact measuring technologies prove to be more efficient and reliable. Among these technologies, ultrasonic rangefinders, optical sensors, and LIDARs are most common. A study in [10] showcases the development of an early flood detection system using ultrasonic and infrared sensors. The system consists of six passive infrared (PIR) sensors and one ultrasonic sensor connected to a microcontroller platform. The system was installed on a pole at the edge of a bridge, looking down at both creek and road directions. All six infrared sensors were used to measure both ground and actual temperature. The ultrasonic sensor was used to measure distance to objects below. Both ground temperature and water level measurements were used to build an ANN model for predicting flood. Multiple neural network architectures were tested. Results demonstrated that air temperature variance affects the speed of sound, which, in turn, affects ultrasonic sensor measurements. Estimating measurement corrections using collected ground temperature led to accurate ANN models. Researchers concluded that ANN models proved to be more accurate than nonlinear regressions, fuzzy-based models, or decision-trees. Optical water level measurement is a relatively new method for flood detection and monitoring—one which has been investigated more widely in recent years. The study in [11] utilized enhanced water level detection based on the edge detection principle for developing a flood detection system. Day and night cameras, an infrared projector, white board, processor unit, data logger, and data transmission unit were utilized. The developed optical gauge was tested at several sites under different conditions (e.g., rain and heavy tropical rain events, snow, fog, daytime and nighttime). Results were compared with other conventional sensors, such as radar and pressure sensors. Accuracy was approximately 95%, 1 cm margin of error for water depth of 75 cm. A study in [12] utilized a scene-text recognition approach wherein the system consisted of four main components, namely water level riser, graduated flood gauge, camera, and microcontroller processing unit. The images captured by the camera were first preprocessed first by reducing the image size to 100 KB. Next, the scaled images were processed using computer vision (CV) and image processing techniques, including background extraction, template matching, and region of interest trimming. After determining the region of interest, dominant color detection and a digit recognition neural network were leveraged to measure water level change. The approach performed extremely well in daytime and showed only some limitations during nighttime, primarily due to lack of riser visibility at night. A similar approach was suggested for real time flood detection in [13] using video surveillance. After the frame was acquired, a background subtraction was applied. Due to the nature of flash floods, flooded water might contain debris and small objects (e.g., trees and rocks), which can be subtracted from the foreground using morphological closing techniques. During the next stage, an 8-directional, connected-component labeling algorithm was used to determine volume size of a body of water to estimate and categorize the flood. Results showed that flash floods rarely have a laminar flow. Given the development of time-of-flight distance sensors, non-water parts make distinguishing flash floods much easier. The use of LIDAR sensors for distance measurement has become more common due to its low-cost, high-energy efficiency, and small measurement footprint. LIDAR-based distance sensing is common for digital terrain mapping [14]. The principle of LIDAR ranging relies on the rugosity of the reflective surface for generating an ideal reflection of the incipient laser beam. As with radar, LIDAR ranging measures time-of-flight, albeit using higher frequency waves for greater pulse intensities. Near-infrared (NIR) light is most often used for this purpose. Wavelengths of 900–1,100 nm (270–330 THz) are common, due to the low cost of lasers operating in this wavelength range and the lower energy density than typical for visible spectrum [15]. LIDAR distance sensing accuracy is determined leveraging two characteristics: 1) pulse width and 2) method accuracy for measuring flight time. LIDAR has rarely been used to measure water levels. A primary difference between terrestrial LIDAR applications and water-level LIDAR applications is the lower reflectivity of water surfaces for infrared radiation compared to solid objects. A calm water surface may behave as a purely specular surface, which produces reflections instead of dispersing incipient energy. In practice, a mirror-like river surface will reflect ∼2–3% of an infrared radiation incident beam [16], while the remainder is transmitted through the water. It is important to note that the reflectivity of water changes with an increase in water rugosity.

Although researchers have rigorously pursued non-contact water level measuring techniques, attention on non-contact surface velocity measuring techniques is scarce and lacking. A concept suggested by [17] uses Doppler LIDAR sensors to measure water velocity. This method was tested in July 2020 [18]. Experiments were performed at two river sites. Obtained results were highly accurate and showed a residual velocity standard deviation of 0.07 m/s. The proposed system was used to monitor surface velocities during flood events, and the measurements provided by the doppler LIDAR were later used to estimate water discharge (i.e., flow rate).

3 Industrial Flood Monitoring and Alerting Systems

To achieve objectives for this thesis, a comparison was performed to identify differences between industrial flood monitoring and alerting systems currently deployed in the USA. Valarm [19], offers many industrial Internet of Things (IoT) monitoring systems, including water resources management, pollution and air quality monitoring, and fluid and water usage. Valarm's IoT water level sensor is composed of three sensors for monitoring (e.g., ultrasonic level sensors, pressure transducers, and Radar level sensors). Ultrasonic sensors send out sound waves from a bridge to determine river water levels. These non-contact sensors are critical, depending on monitoring scenarios deployed. Pressure sensors measure weight pressure of water above them. Radar sensors use electromagnetic waves for monitoring. This type of sensor is more expensive, as its output can penetrate objects that might interfere with level measurements. Valarm monitoring systems can be found on the East Coast, with ultrasonic and radar sensors most commonly deployed (See Figure 3-1).

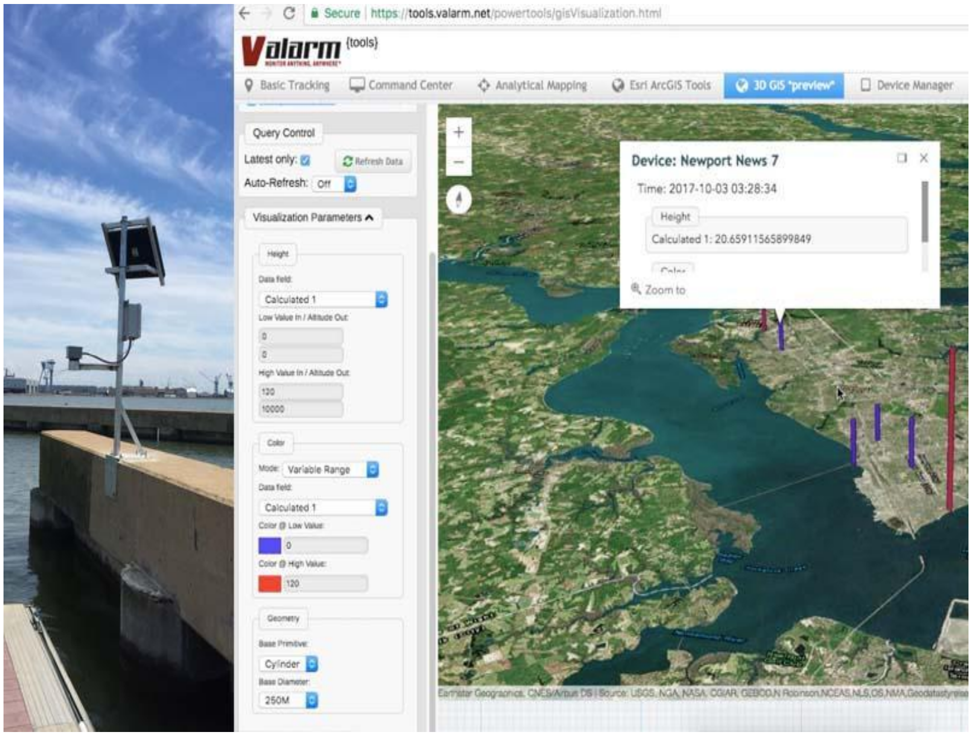


Figure 3-1. Valarm flood monitoring and alerting system.

The U.S. Geological Survey (USGS) and the National Weather Service (NWS) work together to maintain a flood warning system [20] across the country. The system issues alerts in four categories: 1) "Flood Watch," when a flood is anticipated within 12-48 hours; 2) "Flood Warning," when flooding is expected across a large region or is actively taking place; 3) "Flash Flood Watch and Warning," when there is a potential for flooding due to heavy rain or dam failure; and 4) "Flood Statement," when flooding is expected along major streams where there is no threat to people and property. The flood warning system is based on data from local rainfall, stream level, and streamflow; it requires three steps, including data collection via gaging, data processing, and dissemination of flood warning information. Alert gages perform two tasks: sensing and communicating. Sensors are employed to detect changes in precipitation and water levels that affect the parameters. Some alert gages provide site-specific information about the health of a system. After the data transmitter detects a specific event, the system will be activated. Days with no events will report "no rain" to ensure the system is active. Notably, system activity can vary widely from one location to another. Sensors and instruments used, as well as number of gage sites and their locations, is based on the nature of application and the size of the coverage area. A station is typically installed on riverbanks or a standing structure, such as a pier or bridge. Gages can also be built into standpipes, as shown in Figure 3-2. Each station has a tipping bucket rain gauge that collects the amount of precipitation and sends an electrical signal to the data transmitter. A data logging system houses the data logger, telemetry module, and power supply. Mounting hardware includes a pole attached to the bridge. The telemetry module provides access data in real time, transmitting 'ALERT' via radio frequencies. Data is are sent via email when parameters exceed predefined limits. The system provides an advanced and powerful real time monitoring system using advanced products selected for quality, reliability, and value. Each station is equipped with four devices: 1) RLS Radar Water Level Sensor from OTT, which uses radar pulse technology to measure area depth; 2) HSA TB3 tipping bucket rain gauge, which offers precipitation data with less than 3% margin of error for rainfall; 3) NexSens 3100-MAST wireless telemetry system, which contains the data logging system, cellular modem, and solar charging; and 4) WQData LIVE web datacenter, which supports remote access to collected data from any device.

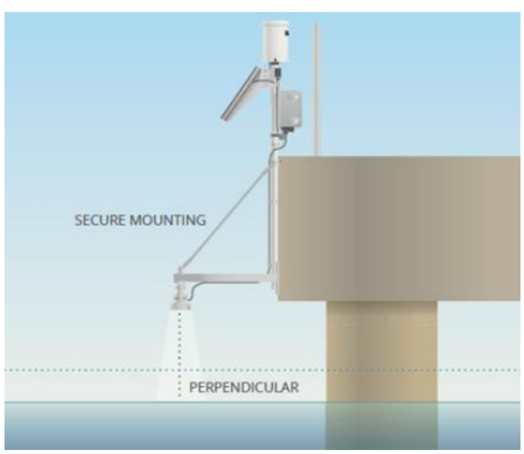


Figure 3-2. USGS and NWS flood monitoring and alerting system.

The “Water Detection Warning System” by ELTEC [21] detects wet, slippery, and/or flooded roads. The system utilizes a bronze-housed sensor positioned at the desired highwater detection point. When water touches the steel electrodes, a signal is sent and LEDs flash continuously. The system will send alerting texts, report flooding to the central office, and make phone calls announcing flood location. For wet road detection, an embedded sensor will warn drivers to slow down prior to reaching the affected area. It will also signal “slippery when wet” prior to drivers reaching an affected area.

Figure 3-3. ELTEC flood road monitoring system.

YSI technology developed another flood alerting system for consideration [23]. The system is currently used by major U.S. government environment monitoring agencies, including USGS and the United States Army Corps of Engineers (USACE). The YSI flood alert system includes a radar level and velocity sensor rain gauge, as well as a data acquisition and transmission cabin.

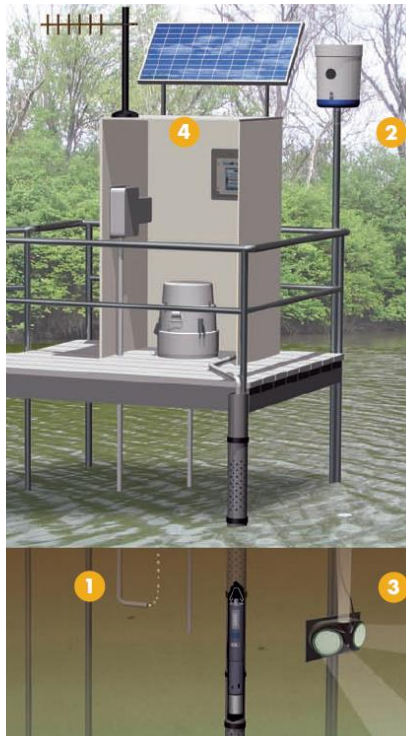


Figure 3-4. YSI flood monitoring and alerting system.

Figure 3-4 shows the four main components of the YSI system: 1) submersible vented pressure sensor (e.g., Dwyer SBLT2-10-40) for monitoring water level; 2) tipping bucket rain gauge for measuring rainfall (H-3401); the SDI-12 output data parameters represent: accumulated rainfall since last measurement, raw bucket count since last measurement, total accumulation, and total daily accumulation; 3) side looking radar water level and velocity sensor (e.g., SonTek-SL3000) was also used in the proposed system to determine water discharge (e.g., the sensor measures the water velocity change, while water level remains fixed; and 4) system cabin where data acquisition and transmission occur. The flood alert system by YSI provides multiple data acquisition methods, as well as accurate early detection for flood events. In spite of these advantages, the system lacks the ability to monitor road surfaces, as it is deployed within bodies of water; it also requires continuous maintenance. Deploying the system in water makes it vulnerable to extreme weather and environmental conditions, as well as confounding elements like rocks and debris, which are likely during flash flood events.

Figure 3-5 shows the system deployed by the USGS at three different locations, namely Georgia (GA), Texas (TX), and Indiana (IN).

Figure 3-5. YSI flood stations in GA (left), TX (middle), and IN (right).

4 Flash Flood Occurrence Forecasting using Weather Data

This section details the system's initial modeling approach. Weather attributes previously recorded by the RWIS system deployed by ODOT on I-35 highway were used to build an ML model using flood mitigating factors that had been proven influential in other studies discussed in Section 2 (e.g., precipitation intensity). Different short-term and long-term forecasting approaches were explored, and results for the best performing model are presented below.

4.1 Analyzing the Weather Changes in Severe Storms

Many studies have analyzed severe storm variables [23], [24], [25]. Previous Oklahoma storms have been accompanied by a significant temperature drop [1]. Storms are often accompanied by gusty winds immediately before rainfall commences. This factor causes flash floods.

The work detailed in this thesis analyzed daily historical weather graphs documented before, during, and after flash flood events reported by ODOT and NWS in Norman. Additionally, time series historical data retrieved from the ODOT Road Weather Information System (RWIS) were utilized. Data includes ambient temperature, air pressure, humidity, wind speed, gust wind speed, brightness, and precipitation intensity. These factors were used to build a flash flood forecast system. RWIS is an intelligent system designed and developed by Dr. Hazem Refai's research team at the OU-Tulsa campus for the purpose of monitoring weather and road conditions. Fifteen stations are currently deployed by ODOT along the Oklahoma's I-35 corridor at road-bridge cross sections. The distance between each is approximately 16 miles (See Figure 4-1).

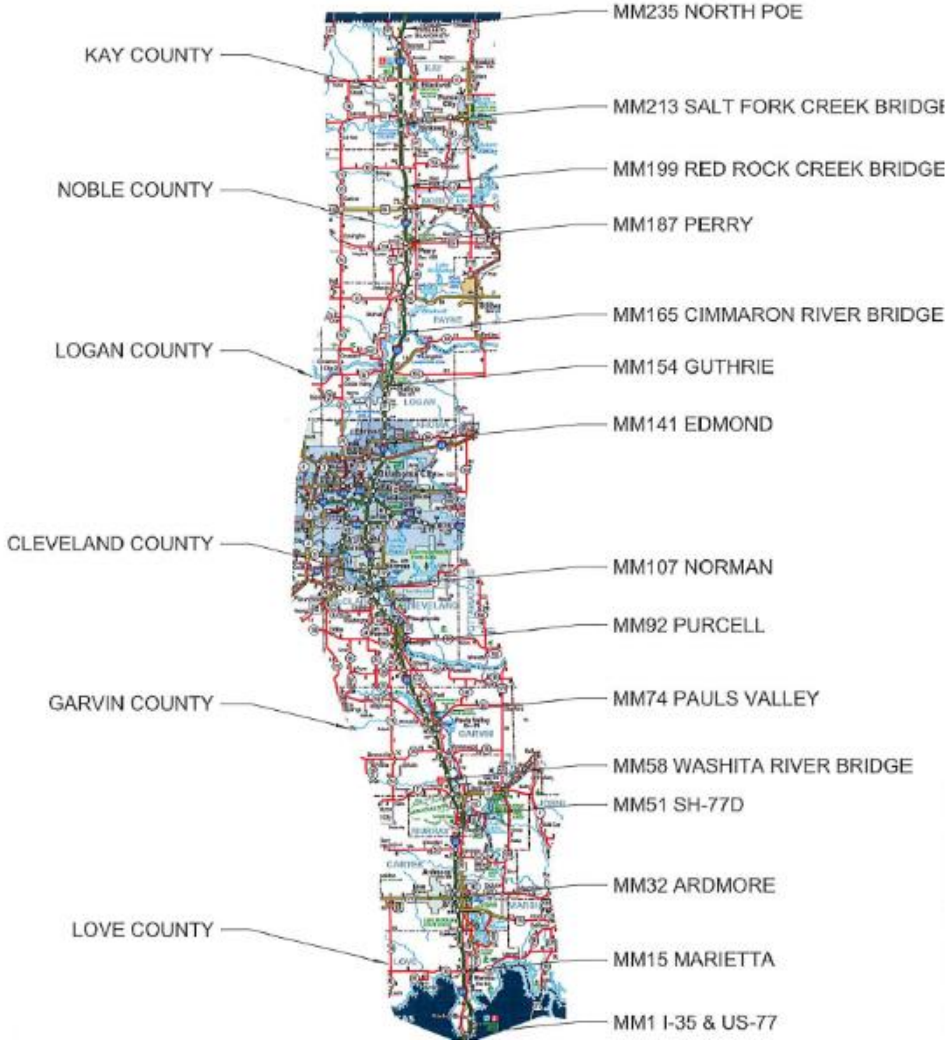


Figure 4-1. Oklahoma Hwy 1-35 RWIS station locations.

Each station consists of six components, as shown in Figure 4-2. The first is a THIES CLIMA [26] weather station. This sensor measures weather data, such as wind speed and direction, air temperature and pressure, humidity, brightness, and precipitation intensity and type. The second component is composed of two InfraRed surface sentinel sensors [27] that measure surface temperatures for both road and bridge. The third houses two underground temperature probes [28] for recording sub-surface temperature at 2-inch and 6-inch depths, and the fourth is an Axis camera for monitoring the road with live stream. The camera is programmed to capture and save a roadway image every five minutes. It is connected through ethernet communication to a modem housed in the system's fifth component—a cabinet, which houses the power controller and the dataloggers for communicating with each sensor. The cabinet's main component is the Roadside Embedded Extensible Computing Equipment (REECE) [29], which is a Linux-based intelligent embedded system for interacting with different system components and establishing connection with a cloud-based server through the Sierra Rv50 gateway [30]. The final system component is composed of two 100W solar panels that feed two 100 AH batteries through a power controller [31].

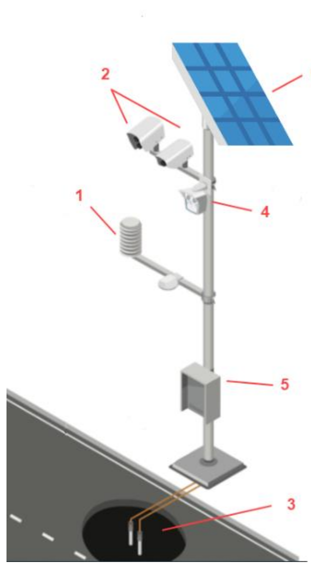


Figure 4-2. RWIS Site 3D illustration.

Oklahoma's flash floods are often the result of intense precipitation with extended duration and close repetition of precipitation and/or a combination of these factors. Over the past several years, such extreme weather events occur more frequently due to the rising global temperature and subsequent climate change [1], [24].

A May 16th, 2021, flood event reported by ODOT on SH-77 near lake Murray became the focus of intense analysis. A resulting flood occurred six miles from RWIS station 35ST015; graphs from the station were used for this study (See Figures 4-3 and 4-4).

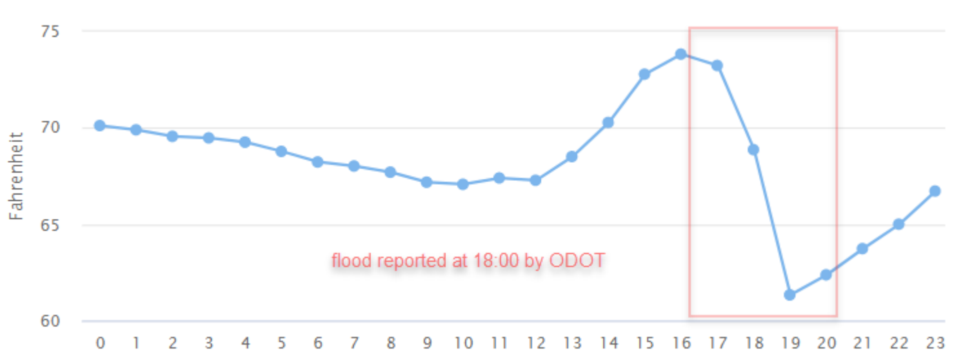


Figure 4-3. station 35ST015 temperature graph from May 16, 2021.

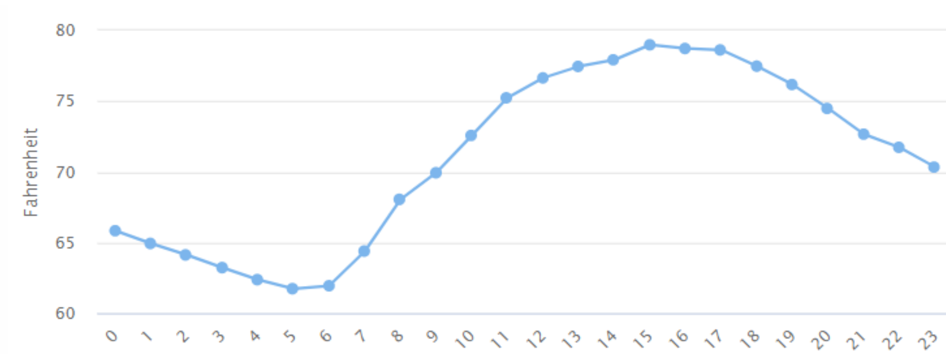


Figure 4-4. station 35ST015 temperature graph from May 15, 2021.

Humidity has been established as a parameter correlated with severe storms. Figure 4-5 shows an air pressure graph for May 16, 2021, on RWIS station 35ST015 where a flood event was reported at 18:00. Figure 4-6 shows the graph for May 15, 2021.

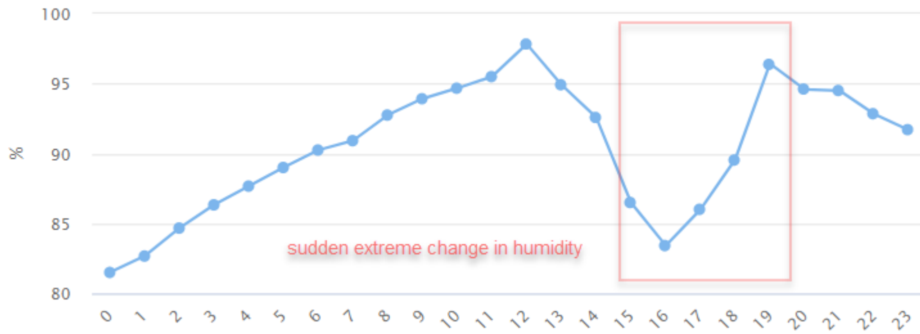


Figure 4-5. Station 35ST015 humidity recordings on May 16, 2021.

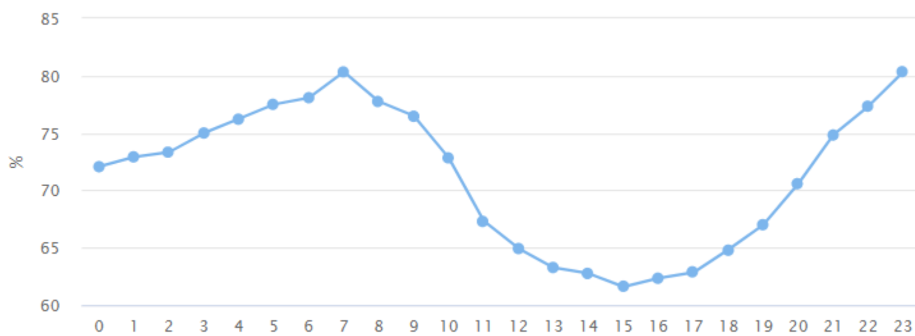


Figure 4-6. station 35ST015 humidity recordings on May 15, 2021.

Comparing these graphs and those based on other dates clearly shows that severe storms are accompanied by a sudden change in temperature and humidity. This information confirms results from previous studies. Other weather parameters (atmospheric pressure, wind speed, and gust wind speed) were captured before, during, and after severe storms and flood events.

Flash floods are caused by intense precipitation, extended duration, close repetition of precipitations, and/or a combination of these factors.

Figure 4-7 documents precipitation data gathered on May 16th, 2021, for station 35ST015. A flood event was reported at 18:00.

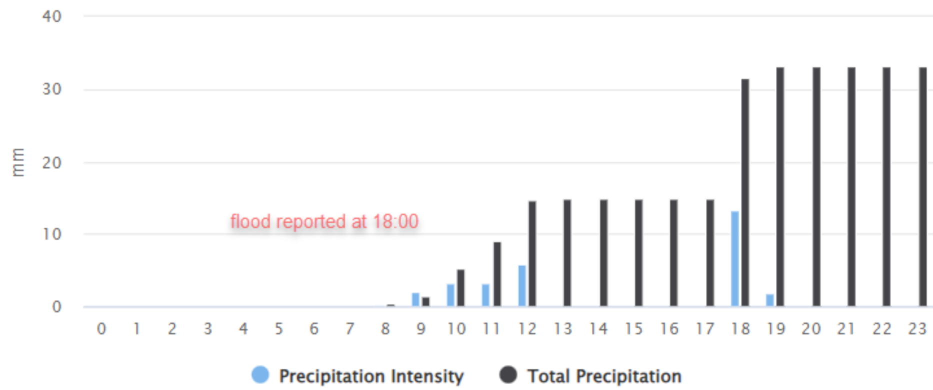


Figure 4-7. Station 35ST015 precipitation measurements for May 16, 2021.

Notably, the flood event depicted in Figure 4-7 was the result of a combination of an extended duration of precipitation, as well as an intense amount of precipitation within a short period of time. These types of flood events are typically easier to predict than other flash floods resulting from very high precipitation intensity occurring in small period of time (e.g., usually within an hour). This type of flood event is referred to as an "extreme flash flood" and is more dangerous and much harder to predict. Figure 4-8 shows the precipitation data on October 10th, 2021, from RWIS station 35ST187, where a flash flood was reported by the nearby NWS.

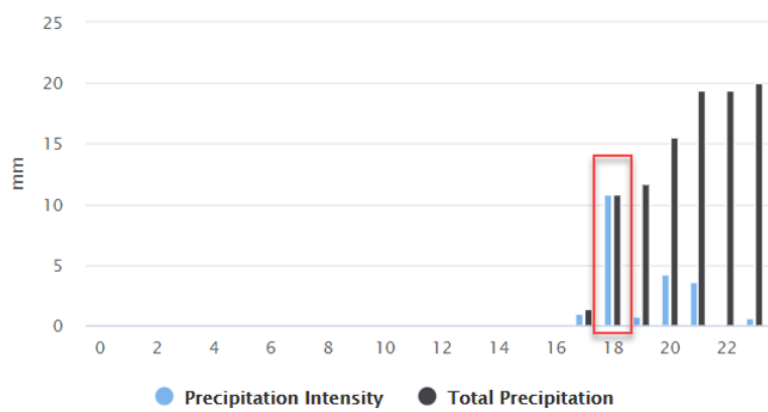


Figure 4-8. Station 35ST187 precipitation measurements on Oct. 10, 2021.

Figure 4-8 illustrates that the flood event was caused by high precipitation intensity falling in a short period of time (e.g., within an hour). It has also been observed that change in weather characteristics is proportional to storm severity. For example, on October 10, 2021, many counties in Oklahoma recorded level 10 on the tornado index and experienced severe weather storms, as shown in Figure 4-9.

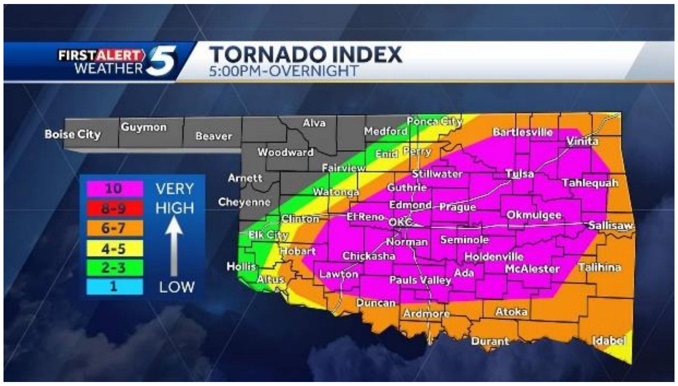


Figure 4-9. Oklahoma tornado index values on Oct. 10, 2021.

Figure 4-10 shows recorded temperature values for the day from station 35ST199, where a flash flood was reported nearby at 18:00.

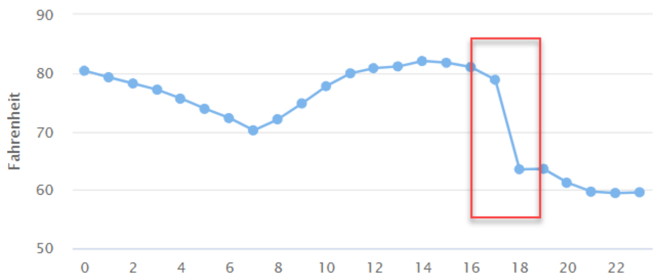


Figure 4-10. Station 35ST199 temperature recordings on Oct. 10, 2021.

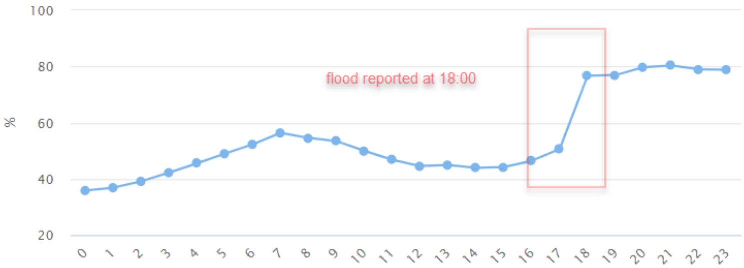


Figure 4-11. Station 35ST199 humidity recordings on Oct. 10, 2021.

When compared to flood events mentioned earlier in this thesis, one can see how the drop in the temperature is more extreme. This observation is in accordance with studies that highlighted positive correlation between storm severity and climate change severity that occurs immediately before or during the storm [32].

The National Oceanic and Atmospheric Administration Weather Prediction Center (NOAA/WPC) noted that 75% of all flash floods reported in the U.S. since 1986 occurred between late April and mid-September; 34% occurred between June 10 and Aug. 3.

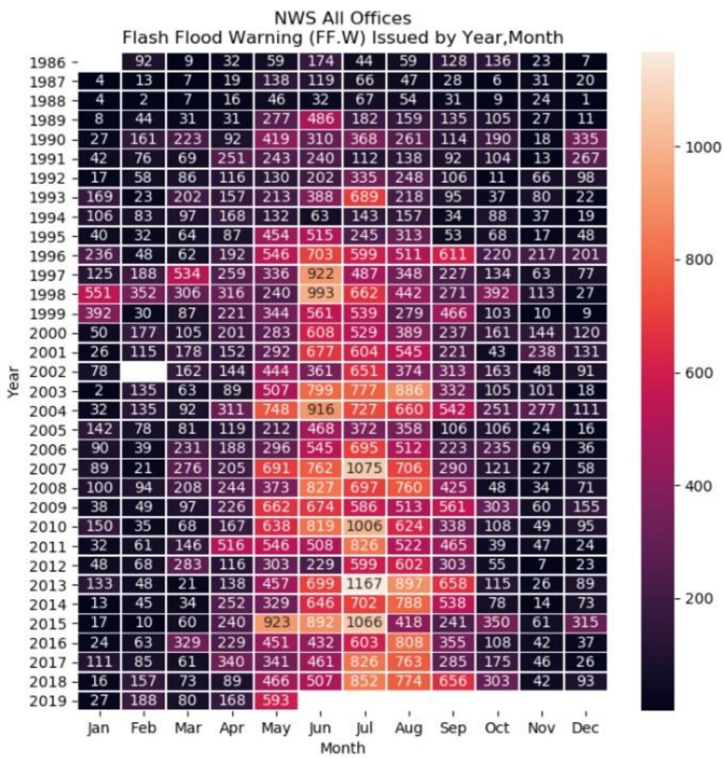


Figure 4-12. Number of flash flood warnings per year in the U.S.

Historical records of flash flood events used in this thesis were provided by ODOT. Additional historical flood data for 2021 were retrieved from the National Center for Environmental Information (NCEI). The following table shows records of flood events in Oklahoma for 2020 and 2021.

Table 4-1. Flash Flood Historical Data in Oklahoma for 2020-2021

County	Date	Start	End	Location	Nearby RWIS Station
Love	01/10/2020	16:00 CST	19:00 CST	[33.93 -97.12]	35ST001
Love	1/16/2020	12:00 CST	23:00 CST	[33.93,-97.12]	35ST001
Love	2/12/2020	02:00 CST	15:00 CST	[34.03 ,-97.07]	35ST001
Love	09/01/2020	10:00 CST	13:00 CST	[33.93, -97.12]	35ST001
Love	04/29/2021	01:00 CST	04:00 CST	[33.83, -96.41]	35ST015
Love	05/16/2021	18:00 CST	22:00 CST	[33.99, -97.12]	35ST015
Love	05/25/2021	07:00 CST	11:00 CST	[34.29, -96.49]	35ST015
Carter	03/19/2020	07:00 CST	11:00 CST	[34.38, -98.26]	35ST032
Carter	09/01/2020	07:00 CST	11:00 CST	[34.54, -97.92]	35ST032
Murray	05/15/2020	16:00 CST	20:00 CST	[34.53, -97.96]	35ST051
Garvin	03/19/2020	09:00 CST	12:00 CST	[34.70, -97.19]	35ST074
Garvin	05/15/2020	15:00 CST	19:00 CST	[34.73, -97.23]	35ST074
Garvin	04/27-28/2021	23:00 CST	03:00 CST	[34.73, -97.25]	35ST092
Garvin	04/28/2021	06:00 CST	09:00 CST	[34.73, -97.25]	35ST092
Cleveland	03/19/2020	07:00 CST	11:00 CST	[34.38, -98.26]	35ST092
Cleveland	08/31/2020	19:00 CST	20:00 CST	[35.20, -97.49]	35ST092
Cleveland	04/28/2021	07:00 CST	11:00 CST	[34.76, -96.66]	35ST107
Cleveland	05/27/2021	16:00 CST	20:00 CST	[35.84, -97.42]	35ST107
Cleveland	06/28/2021	15:00 CST	23:00 CST	[35.18, -97.51]	35ST092
Cleveland	07/01/2021	15:00 CST	18:00 CST	[35.21, -97.45]	35ST107
Oklahoma	03/19/2020	00:00 CST	03:00 CST	[35.49, -97.47]	35ST136
Oklahoma	09/01/2020	08:00 CST	11:00 CST	[35.58, -97.26]	35ST136
Oklahoma	05/27/2021	16:00 CST	19:00 CST	[35.68, -97.17]	35ST136
Oklahoma	06/27/2021	03:00 CST	06:00 CST	[35.46, -97.52]	35ST124
Oklahoma	07/01/2021	11:00 CST	14:00 CST	[35.67, -97.53]	35ST124
Oklahoma	07/10/2021	21:00 CST	23:00 CST	[35.40, -97.51]	35ST107
Oklahoma	08/13/2021	14:00 CST	17:00 CST	[35.50, -97.55]	35ST124
Oklahoma	10/10/2021	19:00 CST	22:00 CST	[35.56, -97.61]	35ST124

Flash flood historical data indicates that 88% of flooding events occurred between March and September. Hence, RWIS data between March 1 and Sept. 30 were used for modeling approaches in this thesis. RWIS CLIMA weather station data accuracy of different weather parameters was validated against data from NOAA weather stations.

Weather Underground [33] is a commercial weather service providing real-time access to the NWS weather stations and over 250,000 Personal Weather Stations (PWS) worldwide. The website is owned by the Weather Company, a subsidiary of IBM, and provides data in many languages. Like ODOT RWIS, Weather Underground displays weather data (e.g., temperature, precipitation intensity, humidity, and air pressure) sampled every five minutes, ensuring accuracy in validation.

Access to the Weather Underground's API network is limited for free users. We used web scraping techniques to extract data from the HTML web page.

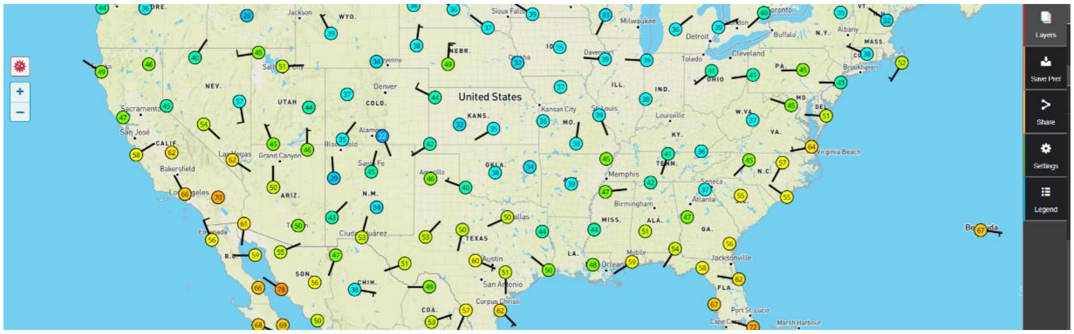


Figure 4-13. Weather Underground graphical user interface (GUI).

Each RWIS station is mapped to a nearby (i.e., within 20 miles) NWS station which adds to validation accuracy. Figure 4-14 shows the location of both RWIS station 35ST015 and NWS weather station KOARDMOR4. The stations are separated by approximately 12 miles and are situated 10 miles from a reported flooded area.

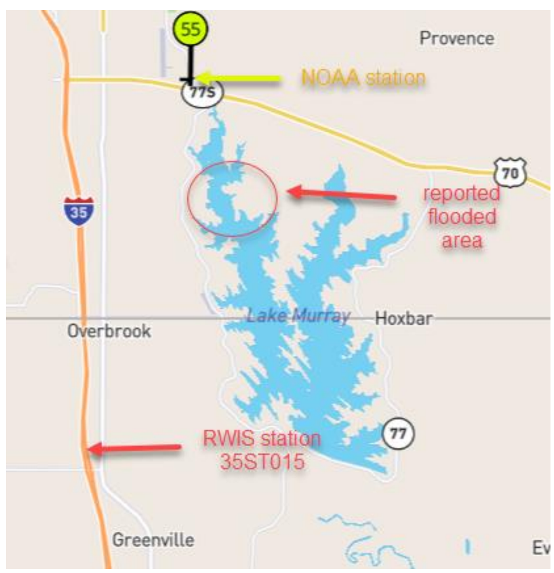


Figure 4-14. Relative location of NOAA and RWIS stations.

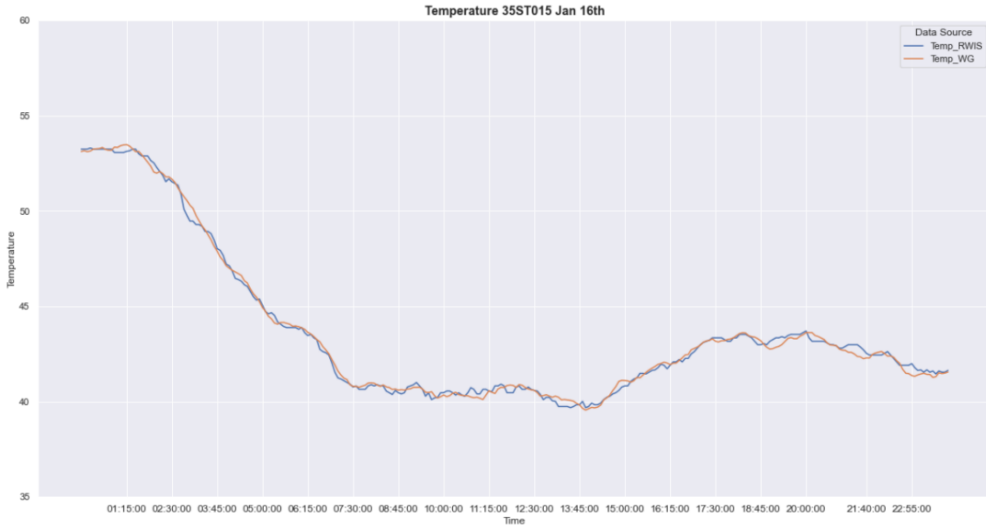


Figure 4-15. WU and RWIS temperatures comparison.

Similar validation processes were conducted for the other weather data recorded by RWIS (e.g., humidity, precipitation, wind speed, and the like).

Table 4-2 shows the Mean Absolute Percentage Error (MAPE) comparison for RWIS stations and NWS weather stations.

Table 4-2. MAPE NWS vs RWIS

Attribute	MAPE
Temperature	%1.13
Air pressure	%3.95
Humidity	%4.08
Precipitation	%3.62
Wind Speed	%2.21

After validating the accuracy of weather data measured by ODOT RWIS, data was used to build and train ML and recurrent neural network (RNN) models.

4.2 Data Preprocessing and Preparation

A Representational State Transfer Application Programming Interface (i.e., RESTful API) was built to interact with the RWIS cloud-based server. Before performing exploratory data analysis (EDA) on the data, missing values were identified and station selection for seasonal data was made (i.e., between March and Sept., 2021). Station down time analysis was conducted, as some stations were inoperable for an extended period of time due to maintenance or hardware malfunction.

Figure 4-16 shows downtime in days for all 15 RWIS stations between March 1 and Sept. 30, 2021.

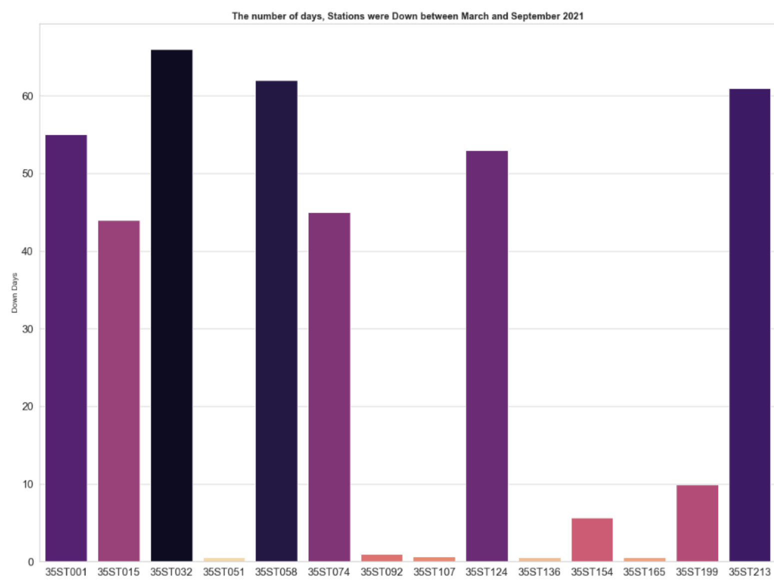


Figure 4-16. RWIS station downtime days between March 1 and Sept. 30, 2021.

Based on this analysis, seasonal data from RWIS station 35ST107 was used, as it experienced only one half day of data loss and was most proximate to historical flood data from Oklahoma County. Various weather features were examined and analyzed to better understand relevance of each and their distributions.

Figure 4-17 represents the boxplot that depicts distributions for each feature. Three y-axis scales were utilized to visually emphasize variability.

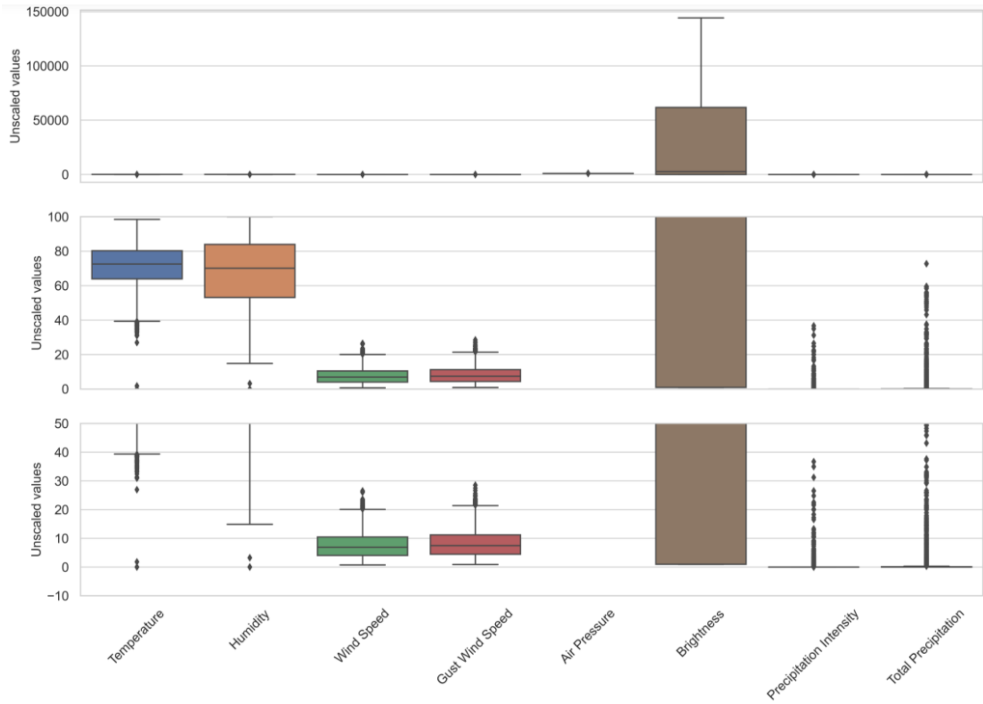


Figure 4-17. Comparison of weather features at different y-axis scales.

Figure 4-17 clearly demonstrates that brightness significantly dominates other features. Such dominance in magnitude might overestimate the feature's contribution to the model, especially when using ML algorithms. The result could produce a biased or inefficient model. To solve this, all weather features were fitted through Z-score normalization. Figure 4-18 shows the scaled distribution for all weather features.

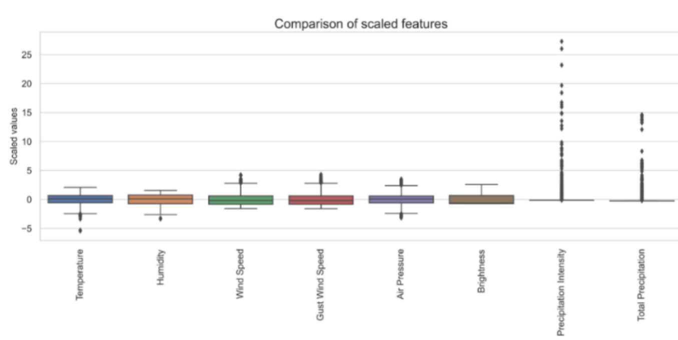


Figure 4-18. Comparison of scaled features.

Boxplots in Figure 4-18 show that most features are continuous. Precipitation intensity and total precipitation features are discrete with some outlier values. These values can be expected, as extreme precipitation events are rare in nature. Next, class distribution for the 6158 instances was plotted in our seasonal data, which was composed of data measured March 1 to Sept. 30, 2021, from station 35ST107 and all other historical flood data reported in 2020 and 2021 from other station locations. Data was classified as either “flood,” when a flood event was reported in that period, or “safe,” when no flood event was reported.

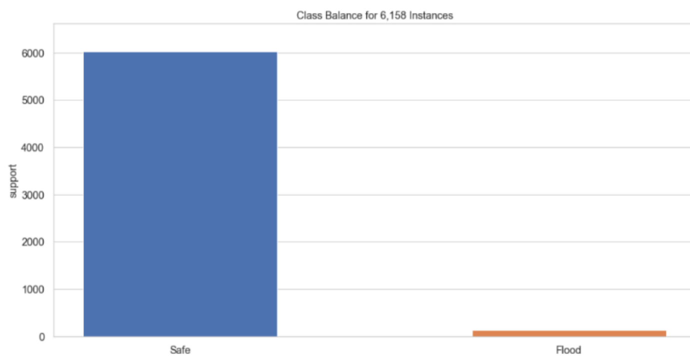


Figure 4-19. Class balance for 6,158 instances.

Figure 4-19 shows only 138 flood instances were identified out of 6,158 weather instances. While many methods can mitigate imbalanced class distribution in datasets, the study reported herein did not use under sampling or over sampling techniques. The objectives was for the dataset to represent a real-world environment, wherein it is natural to have very few flood events.

After preprocessing the data and establishing class balance distribution, an analysis was performed to understand the relationship between the data attributes and storm class. Pearson correlation was initially conducted.

Pearson correlation coefficient (PCC) is a measure of linear correlation between two data sets. Simply put, it is the ratio between the covariance of two variables and the product of their standard deviations. Thus, it is essentially a normalized measurement of the covariance, such that the result always has a value between −1 and 1. As with covariance, the measure reflects a linear correlation of variables.

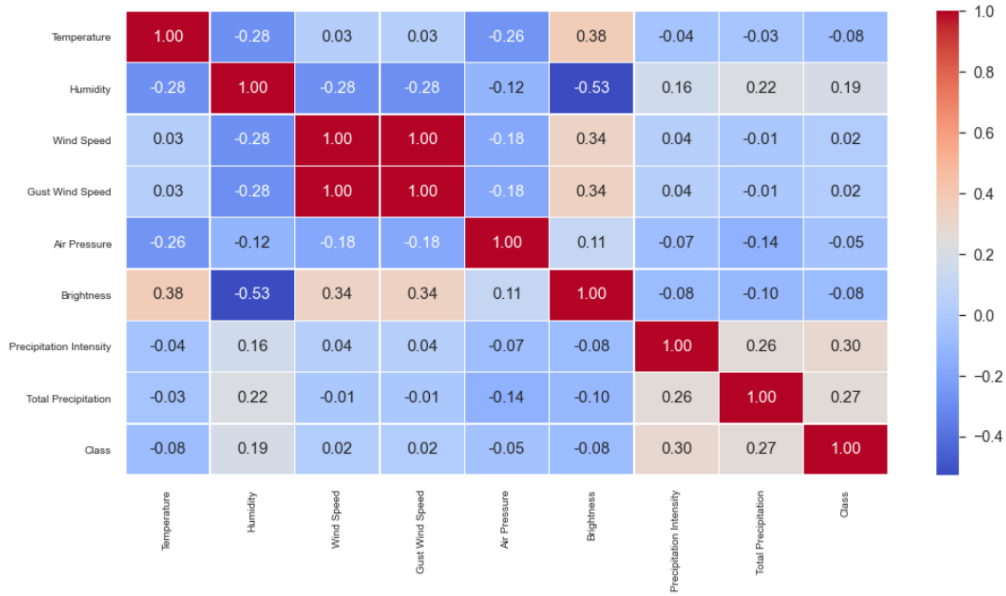


Figure 4-20. Pearson correlation map of data attributes versus storm class.

The correlation matrix in Figure 4-20 shows positive correlation between the class index and precipitation data attributes; it also demonstrates a correlation between class index and the attribute of humidity data. A negative correlation was observed between temperature and air pressure. Wind speed and gust wind speed were highly correlated.

Next, a one-dimension and two-dimension data analysis was conducted. A one-dimensional ranking of features utilizes a ranking algorithm that considers only one feature at a time (e.g., histogram analysis). The Shapiro-Wilk algorithm was utilized to assess the normality of the distribution of instances with respect to the feature.

Figure 4-21 shows that most attributes had a relatively normal distribution. However, this is not true for precipitation intensity and total precipitation, which can be expected, as these attributes are discrete and include many zero values.

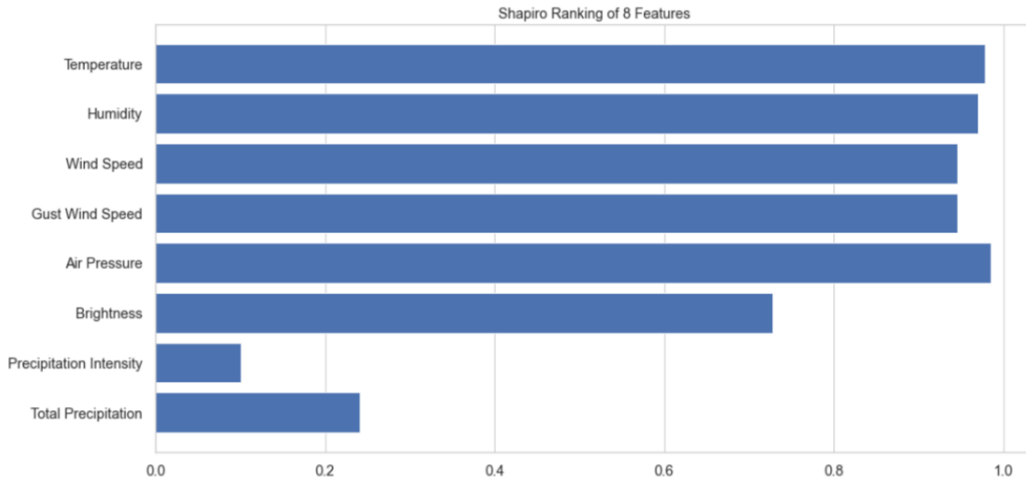


Figure 4-21. Shapiro-Wilk ranking of data attributes.

A two-dimensional ranking of features utilizes a ranking algorithm that considers pairs of features at a time (e.g., joint plot analysis). Features pairs are then ranked by score and visualized using the lower left triangle of a feature co-occurrence matrix. We utilized the covariance ranking algorithm, which attempts to compute the mean value of the product of deviations of variates from their respective means. Covariance loosely attempts to detect a colinear relationship between features. The analysis of covariance (ANCOVA) is a standard statistical methodology that combines the features of analysis of variance (ANOVA) and linear regression to determine if there is a difference in some response variable between two or more groups.

Figure 4-22 shows high inverse correlation between temperature and air pressure and temperature and humidity which supports our initial observations in storms.

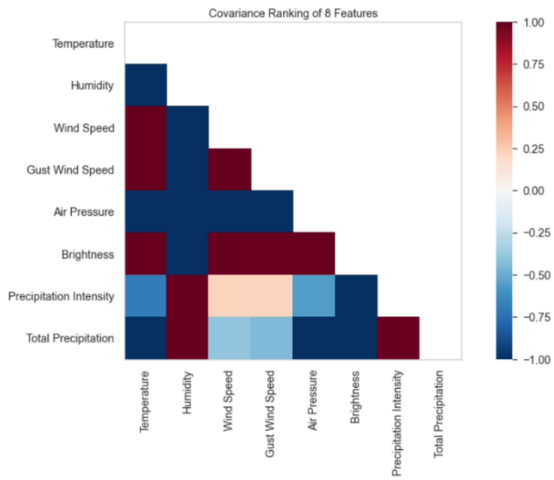


Figure 4-22. Pairwise covariance correlation between all data attributes.

Next, we analyzed the separability of the data points based on the classes. We utilized multiple visualization algorithms and techniques. First, we applied a multivariate data visualization algorithm called Radviz [34]. Radviz plots each feature dimension uniformly around the circumference of a circle then plots points on the interior of the circle such that the point normalizes its values on the axes from the center to each arc. This mechanism allows as many dimensions as will easily fit on a circle, greatly expanding the dimensionality of the visualization. This method is used to detect separability between classes.

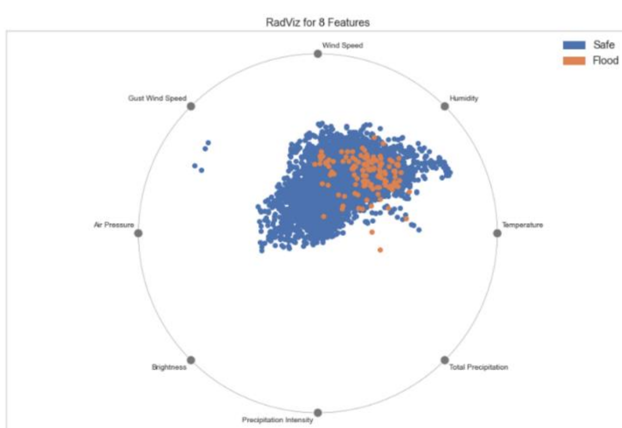


Figure 4-23. RadViz visualization for data attributes versus Class.

Figure 4-23 shows that all data attributes for both classes lean towards wind speed, humidity and temperature and further away from attributes such as brightness, precipitation intensity and total precipitation. Next a PCA visualization analysis was conducted. The PCA Decomposition visualizer utilizes principal component analysis to decompose high dimensional data into two or three dimensions so that each instance can be plotted in a scatter plot. The use of PCA means that the projected dataset can be analyzed along axes of principal variation and can be interpreted to determine if spherical distance metrics can be utilized.

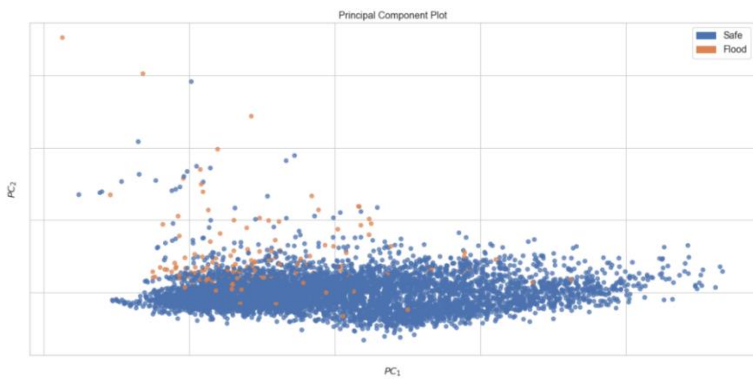


Figure 4-24. 2D PCA visualization.

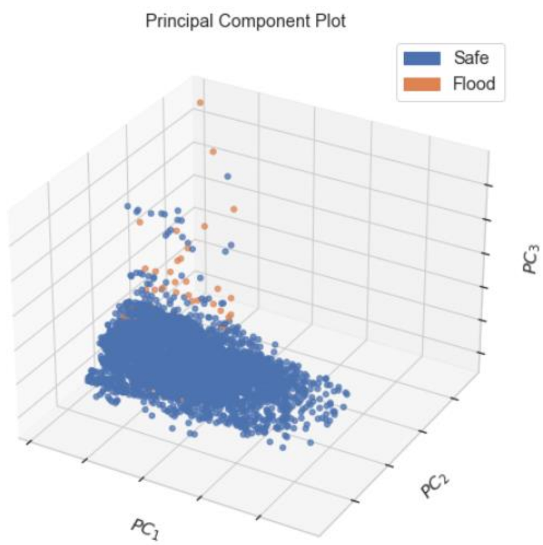


Figure 4-25. 3D PCA visualization.

The PCA analysis presented above utilizes linear dimensionality reduction using Singular Value Decomposition (SVD) of the data to project it to a lower dimensional space. The input data is centered but not scaled for each feature before applying the SVD. It uses the LAPACK implementation of the full SVD or a randomized truncated SVD [35].

From the 2D PCA visualization and the 3D PCA visualization shown in figure 4-24 and figure 4-25 we can notice a little separation for some of the data points belonging to Class flood. To further analyze the separability of the data points a manifold visualization technique was tested. The Manifold visualization method provides high dimensional visualization using manifold learning to embed instances described by many dimensions into 2, thus allowing the creation of a scatter plot that shows latent structures in data. Unlike decomposition methods such as PCA and SVD, manifolds generally use nearest-neighbors approaches to embedding, allowing them to capture non-linear structures that would be otherwise lost. The projections that are produced can then be analyzed for noise or separability to determine if it is possible to create a decision space in the data. Figure 4-26 shows an Isomap implementation which seeks a lower dimensional embedding that maintains geometric distances between each instance.

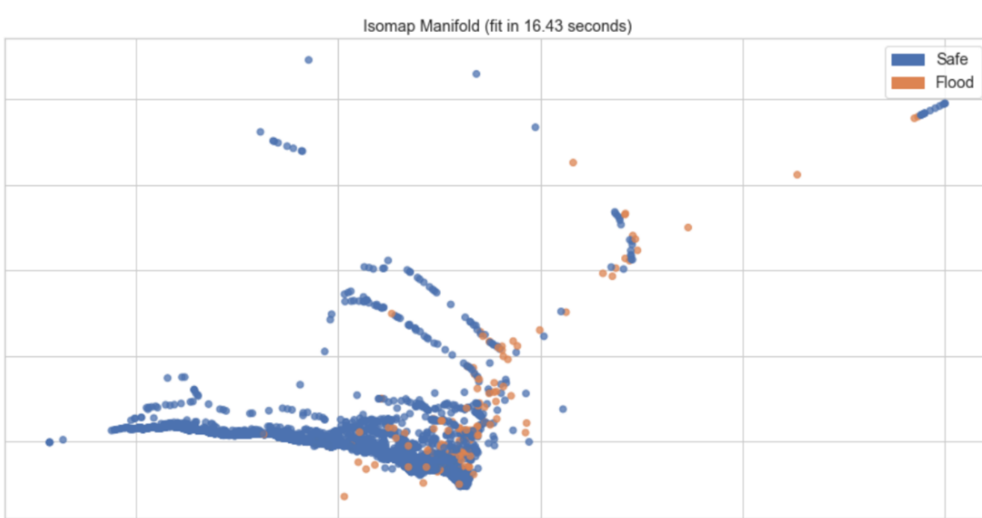


Figure 4-26. Isomap 10-neighbor visualizer.

From figure 4-26 we can observe no separation based on nearest neighbor dimension reduction approach. Finally, a Parallel Coordinates visualization approach was adapted. Parallel coordinates is a multi-dimensional feature visualization technique where the vertical axis is duplicated horizontally for each feature. Instances are displayed as a single line segment drawn from each vertical axes to the location representing their value for that feature. This allows many dimensions to be visualized at once. This method is used to detect clusters of instances that have similar classes.

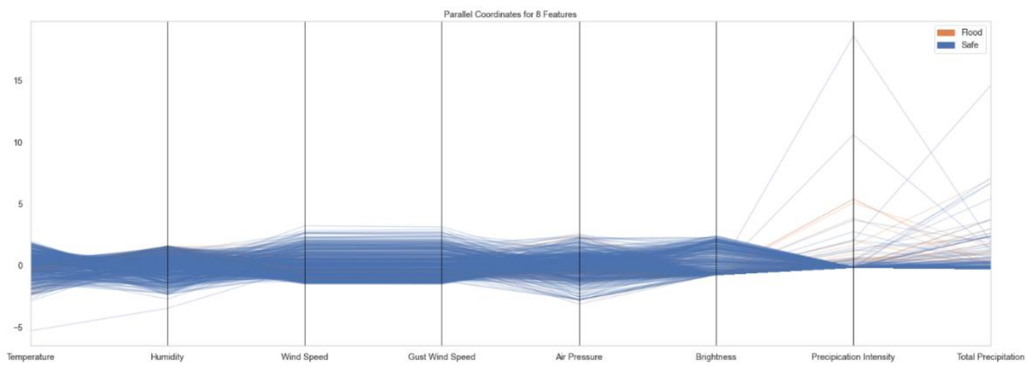


Figure 4-27. Parallel Coordinates visualizer for all data records.

From figure 4-27 we can see that most of the flood class data points are separable based on the precipitation intensity value of that data record.

Finally, we performed a distribution analysis for all features based on classes distribution to see how the distribution for each feature looks like for a given class.

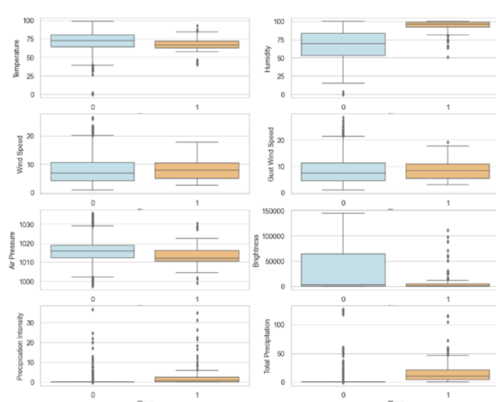


Figure 4-28. Boxplots for features versus class

Figure 4-28 shows how the values of numerical features vary across target groups. For example, we can tell that Humidity have distinct difference when target is 0 (safe) vs. target is 1 (flood), suggesting that it is an important predictor. Similarly, we can observe a distinct difference for precipitation intensity and total precipitation features, we can also notice a small difference for air pressure. However, wind speed and gust wind speed appear to be less outstanding, as the box plot distribution is similar between target groups.

4.3 Applying Machine Learning Algorithms on Time-Series Datasets

While it is most common to use Recurrent Neural Networks (RNN); which are deep learning models such as Long-Short Term Memory (LSTM); to model problems with sequential data such as time series data [15] [16]. Several studies have shown better performance using classical and ensembled Machine Learning algorithms [17], [18]. This was specially observed in studies with smaller datasets. A time series is a sequence of numbers that are ordered by a time index. This can be thought of as a list or column of ordered values. A supervised learning problem is comprised of input patterns (X) and output patterns (y), such that an algorithm can learn how to predict the output patterns from the input patterns.

Machine Learning algorithms do not have memory capabilities and treat each row (record) of data separately. To overcome this, feature engineering was performed on the dataset. Features that would embed the time concept of the time series data were added to the dataset. Four extra features for each weather attribute were added. T-1, T-2, T-3 and T-4 representing the difference between the current measurement at time T and the measurement at time T-n, where n=1,2,3,4 representing a window of 4-hours. To do this a function with four arguments was introduced.

```
def series_to_supervised(data, n_in=1, n_out=1, dropnan=True):
    """
        data: Sequence of observations as a list or NumPy array.
        n_in: Number of lag observations as input (X).
        n_out: Number of observations as output (y).
        dropnan: Boolean whether or not to drop rows with NaN values.
    Returns:
        Pandas DataFrame of series framed for supervised learning.
    """
```

Figure 4-29. Series to supervised data converter.

In this study we are using n_in = 4 representing 4 hours of lag observations. We will use n_out = 1 to represent the forecast period of 1 hour. Table 3 shows the datasets after adding the extra features.

Table 4-3. Data shape after feature engineering

Index	Var1(t-4)	…	Var8(t-4)	Var1(t-3)	…	Var2(t)	…	Var8(t)	Var9(t+1)
1	…	…	…	…	…	…	…	…	0
2	…	…	…	…	…	…	…	…	0
…	…	…	…	…	…	…	…	…	0
120	…	…	…	…	…	…	…	…	1
…	…	…	…	…	…	…	…	…	…

Multiple lag values were tested, and our initial pick of 4 hours sequence input produced the most accurate models as our initial analysis showed that a 4-hour window of data is the most suitable one to capture the weather changes that accompanies flash flood events as shown in figure 4-30.

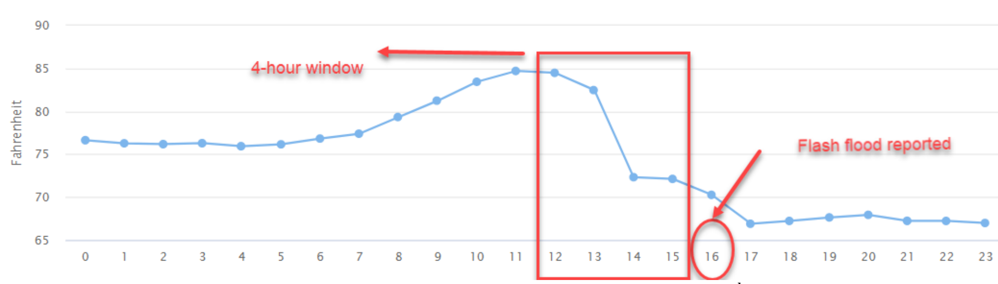


Figure 4-30. Temperature graph on May 27th, 2021.

4.4 Supervised Machine Learning Classification

The modeling we adopted in this study is a binary classification approach where two classes are present, 0 representing a safe class where no flood event occurred and 1 representing a flood event. Each row of data represents a sequence of 4-hour historical data for each feature(X-1, X-2, X-3, X-4) where X-1 is the difference between the attribute x at times t and t-1, the current hour of data for each feature (X) and the class 1 hour in the future (y+1). X is the list of input features : temperature, humidity, wind speed, gust wind speed, brightness, air pressure, precipitation intensity and total precipitation. y represents the class for the next hour and is a binary attribute where y=0 means no flood event(safe) and y=1 means flood event.

Before fitting the data to the models, data were split into %75 for training and %25 for testing. As we are dealing with imbalanced data it is important to split the target classes accordingly to have the same percentage representation for class distribution in both training and testing datasets. A stratify split approach was used to ensure that both the train and test sets have the proportion of examples in each class that is present in the provided train-test split percentage.

Several classical classification algorithms were tested and evaluated. A K-fold cross validation approach was utilized in the training process with k=5. Cross-validation is a resampling method that uses different portions of the data to test and train a model on different iterations. Cross Validation is a very useful technique for assessing the effectiveness of the trained model, particularly in cases to investigate overfitting occurrence.

The performance for each method was then summarized using score metrics defined in terms of true-false positive and true-false negative.

ROC_AUC score which is an evaluation metric for binary classification problems. It utilizes the Receiver Operator Characteristic (ROC) curve that plots the True Positive Rate (TPR) against

False Positive Rate (FPR) at various threshold values. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the classes.

Precision is a metric that quantifies the number of correct positive predictions made.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

Recall is a metric that quantifies the number of correct positive predictions made from all positive predictions

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

The F1 score is a weighted harmonic mean of precision and recall. Generally, F1 scores are lower than accuracy measures as they embed precision and recall into their computation. Recall, precision and F1 scores will be obtained on a per-class basis using the Classification Report method from the Sci-kit learn python library. Confusion Matrix and an ROC_AUC plot is also provided to evaluate each model.

The first Machine Learning classification algorithm we tested was the Decision Tree Classifier. Decision tree is consisted of tree branches where each branch is considered as a conditional statement. The hierarchy is then formed based on a feature importance score. Final classification happens at the leaves of the decision tree.

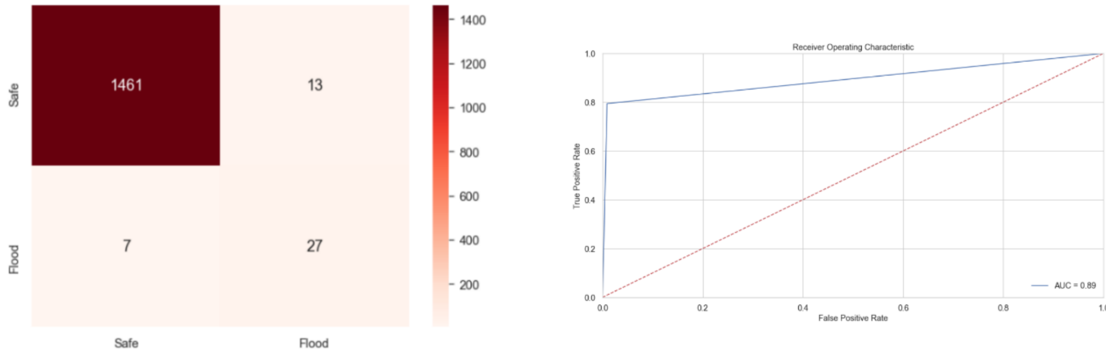


Figure 4-31. Confusion Matrix and ROC plot for Decision Tree Classifier.

Table 4-4. Classification report for Decision Tree Classifier

Class	Precision	Recall	F-1 Score	Support
Safe	1	0.99	0.99	1474
Flood	0.68	0.79	0.73	34

The decision tree classifier showed promising results with an AUC score of %89 and a recall value of %79 for classifying flood class. Decision trees algorithms are immune to multicollinearity by nature and work well with small size datasets hence the good performance compared to other classifiers.

Next, a Support Vector Machine Classifier was used. Support vector machine tries to find the best split that fits data points belonging to different classes. The decision for the split is based on the relative position in relation to a supporting border. This border is known a as the hyperplane which maximize the distance between data points from different classes.

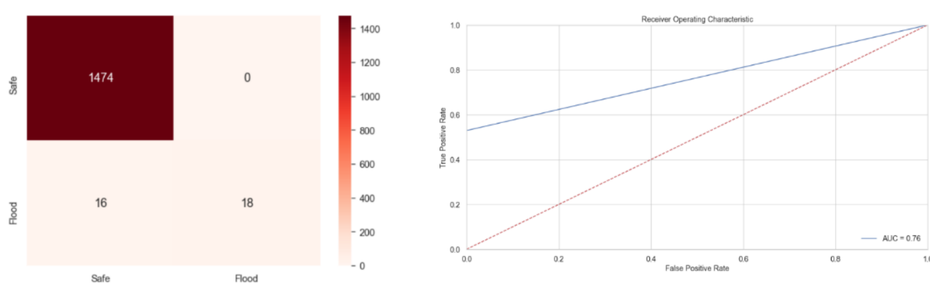


Figure 4-32. Confusion Matrix and ROC plot for SVM Classifier.

Table 4-5. Classification report for SVM Classifier

Class	Precision	Recall	F-1 Score	Support
Safe	0.99	1	0.99	1474
Flood	1	0.53	0.69	34

The SVM classifier showed poor results classifying the flood class with a recall value of %53 and an AUC score of %76. A survey study in [36] showed that with SVM classifiers they tend to preform poor when the number of records (rows) are much smaller than 2^K where k is the number of features (columns).

Most of the classification algorithms do not work well with imbalanced datasets as they try to minimize the error rate rather than focus on the minority class. Decision Tree (DT) based Ensemble machine learning algorithms tend to preform very well on unbalanced datasets as they have the ability to incorporate both class weights; hence it penalizes misclassifying the minority class and sampling techniques as each tree is built on portion of the dataset; hence growing the decision trees on a balanced dataset.

Next two Machine Learning Ensemble methods were tested. Random Forest which is a bagging technique utilizes a number of decision trees on subsets of the given datasets. The Random Forest algorithm takes the prediction from each tree and based on majority votes it predicts the final class.

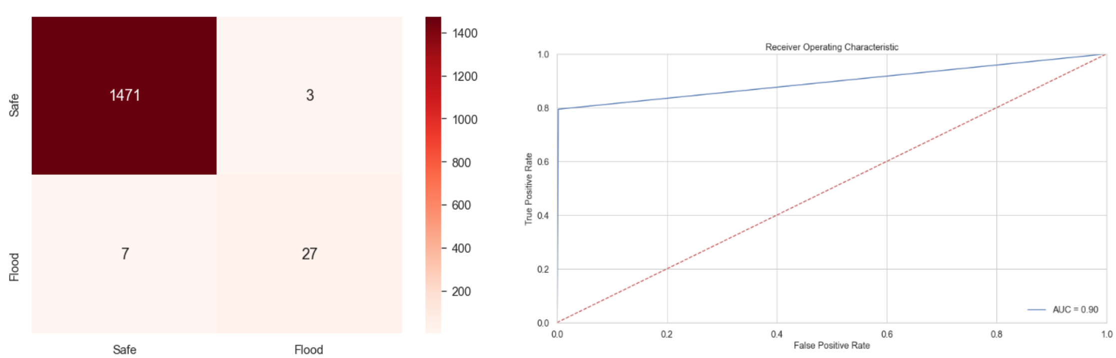


Figure 4-33. Confusion Matrix and ROC plot for RF Classifier.

Table 4-6. Classification report for RF Classifier

Class	Precision	Recall	F-1 Score	Support
Safe	1	0.99	1	1474
Flood	0.90	0.79	0.84	34

XGboost is an ensemble of Decision Trees like the Random Forest algorithm. However, with XGboost each new tree depends on the evaluation of the previous tree. This sequential way of adding Decision Tree classifiers is called boosting. Unlike traditional boosting the XGBoost runs tree optimization in parallel which makes it faster compared to other boosting and ensemble methods. A grid search method was tested to tune the optimal set of hyperparameters to use for a XGBoost classifier.

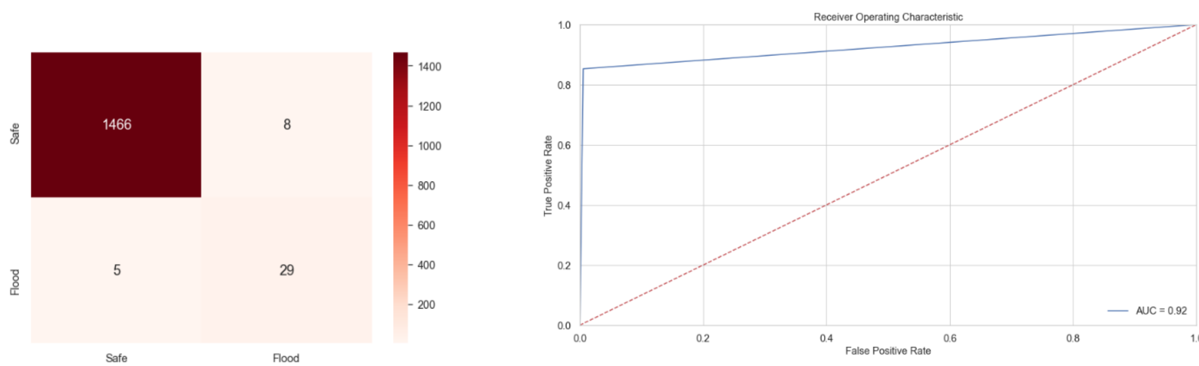


Figure 4-34. Confusion Matrix and ROC plot for XGBoost Classifier.

Table 4-7. Classification report for XGboost Classifier

Class	Precision	Recall	F-1 Score	Support
Safe	1	0.99	1	1474
Flood	0.78	0.85	0.82	34

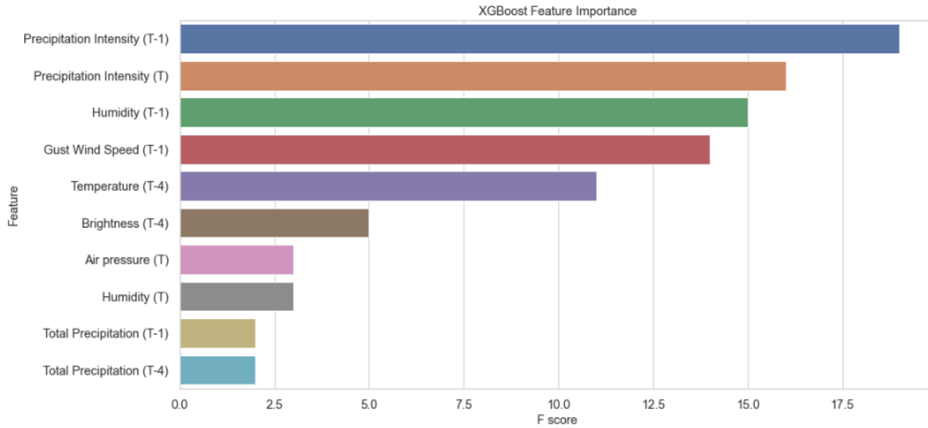


Figure 4-35. Feature Important based on the XGBoost Classifier.

Figure 4-35 shows the feature importance score. The feature importance score simply represents the number of split decisions each feature was used. We can see that the one-hour difference in precipitation had the highest importance with 18 split decisions based on its value. The one-hour difference of humidity and gust wind speed also played an important role in building the model highlighting the importance of the features added to represent the change over a window of time.

4.5 Recurrent Neural Network (RNN) Classification

Recurrent Neural Networks (RNN) was introduced as a concept in 1986 [37]. RNN targets sequential modeling where the current output depends not only on the current input but also on previous inputs. Recurrent Neural Networks suffer from loss of information which is often referred to as the vanishing gradient problem [38]. The vanishing gradient problem is caused by the repeated use of the recurrent weight matrix. A recurrent neural network architecture was proposed in 1991 to solve the vanishing gradient problem [39] and was called Long Short-Term Memory (LSTM). In LSTM the recurrent weight matrix is replaced by an identity function and is controlled by a series of gates. This method gained popularity for modeling sequential data especially time series data and became the most cited architecture with over 16,000 citations in a single year in 2021 [40].

Table 4-8. LSTM Classification model Architecture

Layer Type	Output Shape	Param	Activation Function
LSTM	(None,4, 4)	208	Hard Sigmoid
Batch Normalization	(None, 4,4)	16	-
LSTM 2	(None, 2)	56	Hard Sigmoid
Batch Normalization	(None ,2)	8	-
Dense 3	(None, 1)	3	Sigmoid

The LSTM model built in this study is a simple two layered with batch normalization to accommodate for the number of dimensions. Several architectures were tested through trial and error.

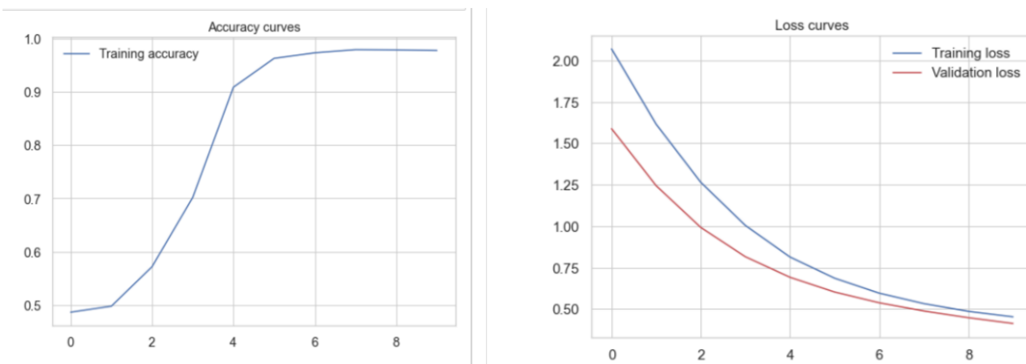


Figure 4-36. Accuracy and loss curve for the LSTM classification model.

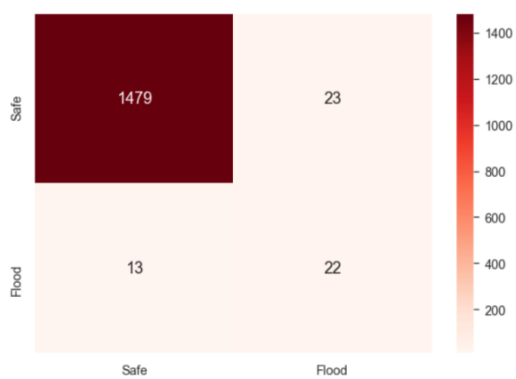


Figure 4-37. Confusion Matrix for the LSTM model.

In figure 4-36, x-axis represents the number of epochs which we set to be 10. We can see that the model fit the data and reached the accuracy within the first 6 epochs as the number of samples in the data is too small to fit a deep learning neural network. this highlights the importance of the feature engineering we preformed to fit the time series dataset to be used with machine learning classical and ensemble methods that preformed much better than RNN LSTM models on the small dataset.

In this study the best performing classification model was a XGboost ensemble classifier with %85 accuracy detecting the flood class and %92 AUC score. This study emphasizes the importance of understanding the characteristics of weather conditions in flood events in order to choose suitable classification algorithms and forecast time windows. However, the modeling that produced acceptable results was only a 1-hour forecast model as these weather changes happen extreme precipitation events.

5 Non-Contact Flood Detection Technologies` Evaluation

Climate changes are not the only factor in determining the occurrence of flood events, terrain of the water body and the hydrological-topographical characteristics of the area plays an important role in determining a flood probability for a given area as well [41].

5.1 Design and Development of an Open-Channel Water Flow Emulator

Obtaining environmental field measurements is very important. However, experiments conducted under such conditions are often difficult to replicate due to several uncontrollable factors such as background flows and surface topography. Having the capability to perform and repeat hydrodynamic experiments in a controlled setup is essential to be able to accurately evaluate the water level and water surface velocity measuring technologies under study. A proper research-purposed and commercial water tanks are very expensive, for example the Hills Research Corporation’s Model 1520 water tunnel [42] shown in figure 5-1 costs around hundred thousand

USD, the tunnel can generate flow up to 0.3 m/s (11.87 inch/s) and can hold approximately 3785 liters (1000 gallons).

Figure 5-1. Model 1520 water tank

Due to the high cost of the commercial water tanks in the market, we suggested developing a relatively low-cost water tank with acceptable quality to create a repeatable testbed environment to properly evaluate the non-contact water level and surface flow technologies. A study in [43] shows the development process of a water tank for the purpose of studying fluid dynamics. However, the proposed design did not consider the flow characteristics or adjusting the flow levels during an experiment. Before building the testbed, it is very important to understand the characteristics of the water surface profiles in open channel flows in order to simulate a natural body of water flow. A study in [44] classifies water surface profiles based on two criteria: Hydraulic slop and Hydraulic Curve, based on that the water surface profile is then classified into different categories. A study in [45] conducted by the United States Geological Survey (USGS) analyzed the Hydrologic and Geospatial characteristics of the lower part of the Arkansas river. The study shows that in high-energy depositional environments (stream and river channels, creeks) the riverbed tends to have a mild slope, which means that the surface water flow is not affected by the underwater flow. Based on the results of this study we can build the water tank to have equal depth (horizontal slope) and still be able to emulate a high-energy depositional environments water surface flow. Figure 5-2 shows a 3D illustration of the water tank, the dimensions are: 1.7 m (70 inches) in width, 1.6 m (62 inches) in depth and 1m (40 inches in width). These dimensions were picked to meet the specifications of the sensors under evaluation such as field of view, and minimum water depth.

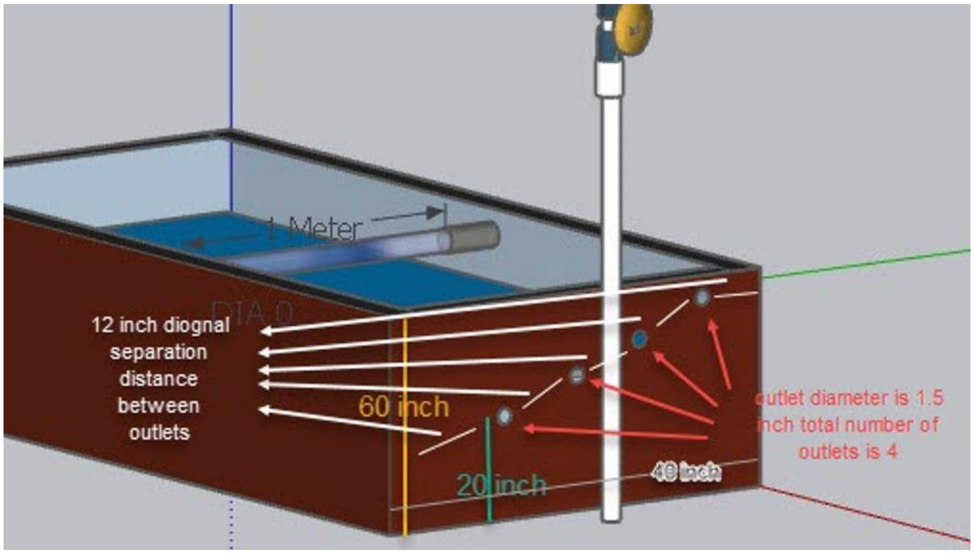


Figure 5-2. 3D illustration of the water tank using SolidWorks

As shown in figure 5-2, the tank will have one inlet and 4 outlets. Attached to the tank is a height adjustable arm where the sensors will be attached. The tank can contain around 2700 L (700 gallons) so proper bracing and support was required. The tank was built with the help of the OU operation team. ¾ inch plywood was used due to its durability and relatively less expensive than other available building materials (glass), the interior was then covered with a water-proof epoxy layer. The exterior of the tank was also painted to prevent degradation and erosion.


Figure 5-3. experimental water tank

As mentioned before, it is very important to analyze the characteristics of the surface flow of the intended area to accurately evaluate the water level and surface velocity sensing technologies. The sensors are planned to be deployed on the sides of a river or a stream channel in the state of Oklahoma, so in order to guarantee accurate evaluation it is important to create a similar environmental setup. Many propeller water pumps introduce modes that simulate currents found in high-energy reefs. The modes are produced by changing the frequency of the pump emulating crashing waves and tidal surges. An example is shown in figure 5-4.

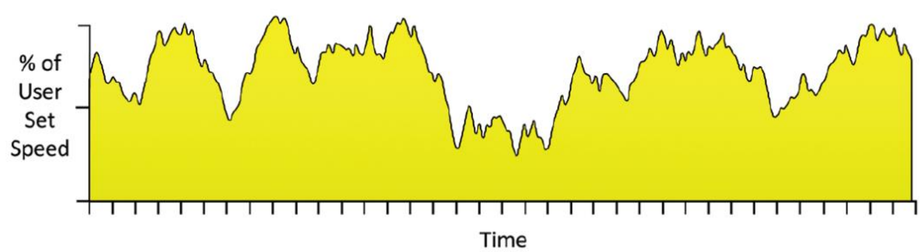
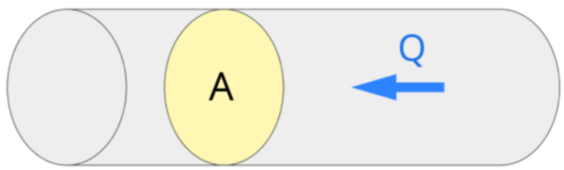


Figure 5-4. high-energy water surface flow emulation [46]

The Vortech MP60 [46] comes with a built-in mode that can achieve this. However, this unit is quite expensive compared to other propeller pumps in the market, the company was also backlogged to May 2022, as a result we went for the available less expensive Aquality LBS-50

[47]. This pump uses an adjustable sine wave which allows us to fine-tune the desired wave forms A set of 3 water pumps are used to generate enough energy and turbulence that would emulate a river channel or a creek. The sensors are combined using a wired master/slave communication protocol. It is important to differentiate between the water velocity V usually measured in m/s and the flow rate Q usually measured in L/min. The velocity V is equal to the flow rate Q divided by the cross-sectional area A, for ease of calculation it is usually assumed that the area of interest is circle shaped, this is why most non-contact water velocity sensing technologies tend to have their sensor lenses shaped as circle or a square.



$$V = \frac{Q}{A}$$

Figure 5-5. Water Velocity calculation

The average velocity of the Arkansas river is between 0.6 and 0.7 m/s this is based on the facts that the area of the river is around 170000 miles squared, and the average flow rate is 68843885 L/minute. In extreme condition the velocity can reach up to 9 m/s. similarly to generate a water velocity within that range and based on the area of view for the sensors under study and the overall area of the water tank we can achieve similar water velocity range (from 0.3 m/s up to 1.5 m/s) with a flow rate that ranges between 1 and 60 L/minute. This flow rate is achievable using a ¾ garden hose which is used as an inlet for the water tank.

5.2 Water Level and Surface Velocity Ground Truth Sensing Setup

After setting up the water surface flow to emulate a real-world environment. It is also important to establish an accurate ground truth sensing mechanism based on which we can use to compare the accuracies of the different non-contact water level and surface velocity technologies under evaluation. Using contact sensors such as pressure sensors and pressure transducers for water level and surface velocity measurements has proven to be most accurate under lab experimental setup [8], [9]. Although these methods suffer from major drawbacks under real-world environment measurements as they are very sensitive to their orientation with respect to the water flow, they are also affected by debris, rocks and extreme weather conditions, they also require periodic maintenance and calibration. However, for a lab experimental setup like the one we developed where we can constantly monitor and maintain them, they provide the most accurate results compared to other traditional contact and non-contact measuring techniques and technologies. The Blue Robotics Bar02 ultra high-resolution depth pressure sensor [48] is used to continuously log water level data, the sensor was calibrated using a ruler. The sensor communicates over I2C with 3.3 V logic level. We used REECE [29] to communicate with the sensor over I2C, the REECE provides a 3.3 V logic input, so no extra level converter was required.

Figure 5-6. Bar02 depth pressure sensor

The Digiten G ¾ Hall effect sensor [49] was used to continuously log water flow rate data. The water flows in through the inlet, out through the outlet and three flow meters to measure the average water velocity in the point of interest. The water current drives the wheel to turn and the magnet on the wheel turns with it. The magnetic field rotation triggers the Hall effect which outputs high- and low-level square waves.

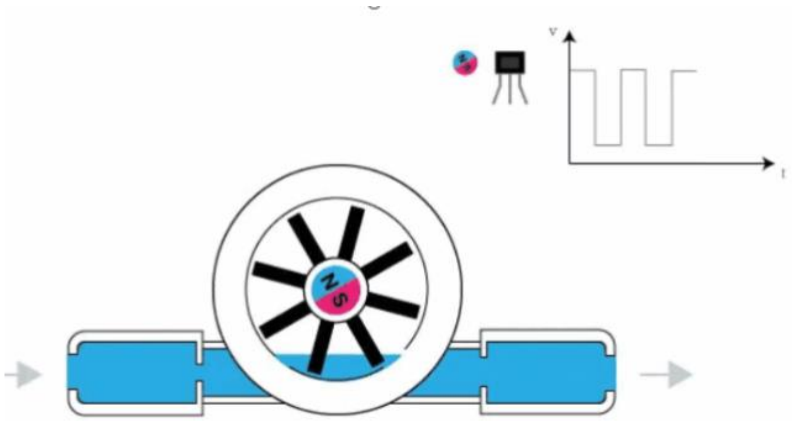


Figure 5-7. Hall sensor working principle

All the flow meters were calibrated through the bucket and stopwatch method. As we are interested in logging flow data from more than one point at the same time (inlet flow rate, outlet flow rate and area of interest flow rate). It is important to synchronize these different sensors to measure at the exact moment as the flow rate is constantly changing due to the change in the waveforms generating the high-energy flow form discussed above. Synchronizing these sensors is a difficult matter as they do not have an internal clock that can be forced synced through an external datalogger like the method we used for the water pumps. The method we approached to achieve this synchronization is to utilize GPIO pins of the REECE which can be used as interrupt inputs (up to 4 interrupts). The code forces an external interrupt on the GPIO pins connected to the sensors when a pulse is detected triggering a pulse counter, after the duration of the interrupt is finished (measuring period), the number of pulses is then utilized to calculate the flow rate.

5.3 Water Level and Surface Velocity Technologies Evaluation

Two water-level sensors and one water surface velocity sensor were evaluated. In this section we discussed the specifications for each technology we evaluated. Experimental setups that we conducted and based our results on is further detailed. The different setups are created by changing parameters such as: distance between sensors and water surface, tilt angle of the sensor, the rate of water level change, changing the water surface and flow characteristics.

5.3.1 Water Level Measuring Technologies

The first technology we will detail is the NovaLyx Ultrasonic water depth sensor [50]. This sensor communicates over RS232 8N1 ascii protocol. The measurement range is between 0.5 m up to 10 m. The sensor is set to only transmit data so there is no Rx data line between the firmware and the user end. We can only start a measuring cycle by setting the enable cable to high, this process requires and additional voltage switch or by directly connecting the enable cable to the power source we can achieve continuous measuring cycles. Each measuring cycle takes between 3 to 5 seconds depending on the number of attempts it made to deem the measurement successful. To get accurate measurements it is recommended to use a programmable switch to setup a measuring cycle no less than 30 seconds.

The sensor outputs 4 values, first value is the air temperature in kelvin, as the accuracy of an ultrasonic sensor is affected by the air temperature. The sensor uses this temperature to accommodate for distance error measurements based on an offset factor calculated internally.

The second value is the measurement cycle duration in milliseconds. The third value is the distance measurement to the target (water surface). This value is calculated using the following equation:

$$Q = S + (F * (K - 273))$$

Q: The speed of sound after factoring out the temperature effect.

S: The speed of sound in dry air at 20 c.

F: Constant (0.607).

K: Temperature in Kelvin.

Then the distance to the target is calculated using the following equation:

$$D = T * Q$$

Where T is the time of travel.

The fourth value the sensor outputs is the number of measurements it took to produce a successful measurement.

Figure 5-8 shows how the sensor is mounted on the testbed we built for the purpose of evaluating these technologies.

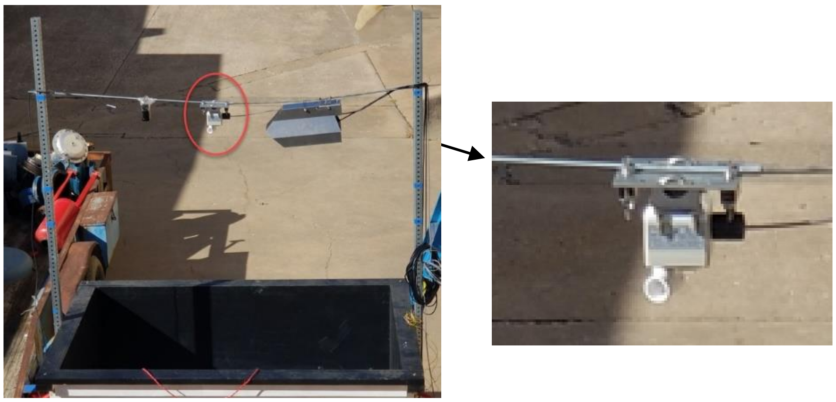


Figure 5-8. NovaLyx Ultrasonic water level sensor mounting position

The sensor was mounted using loop brackets that were purchased separately. The sensor was mounted vertical to the water surface as highlighted in the sensor manual.

The second water level measuring technology we evaluated is the OTT RLS [51]. The sensor communicates over sdi12 ascii protocol, it can also interface through RS485 8N ascii protocol. The measurement range is between 0.4 m and 35 m. The sensor can transmit and receive data over the sdi12 data line. The sensor outputs different parameters depending on the command sent through the sdi12 interface. The senor can be set to two different measuring modes, the first mode can output level measurement (water level related to a level zero), the second mode; which is the mode that we used; measures the distance between the sensor and the water surface.

The sensor outputs the following parameters: time in second the sensor has determined the measured result acceptable, level/distance value (outputs +99999999 in case of invalid measurement), status of the measurement (0 measured value acceptable, 2 no target detected, 4 internal error, 8 variance of individual measurements is too large, 16 sdi12 interface interruption) and an SNR value in dB (value>15 means good signal quality which indicates well-chosen mounting location and parallel alignment.

Figure 5.9 shows how the sensor was mounted to the testbed:

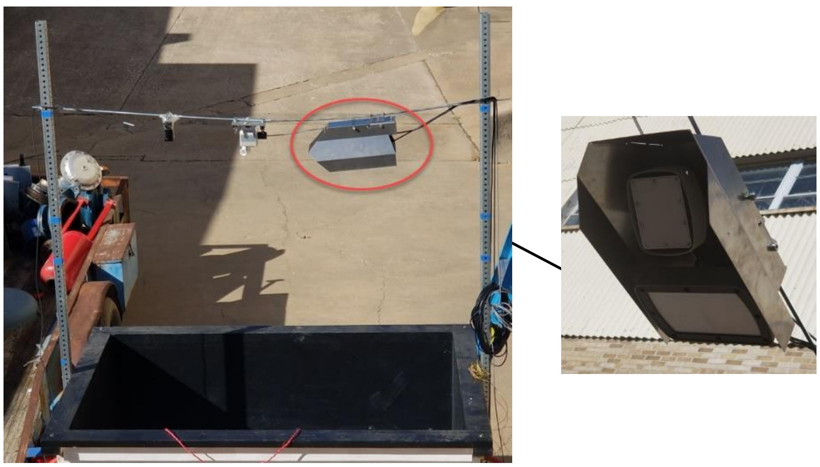


Figure 5-9. OTT RLS Radar water level sensor mounting position

The sensor was mounted with the OTT surface velocity radar using a special housing that was purchased separately. The sensor was parallel aligned with the water surface as instructed in the manual.

5.3.2 Water Surface Velocity Measuring Technologies

For water surface velocity technology evaluation, we chose the OTT SVR [52]. This sensor interfaces with sdi12 ascii protocol, it can also interface through RS485 and RS232 ascii protocols. The measurement footprint ranges between 0.5 m up to 35 m. The sensor needs to be mounted at a nominal angle of 30 degrees to the horizontal. The alignment must be parallel and ideally against the main flow direction. The functional principle of the sensor is based on the physical Doppler effect. A transmission aerial emits radar pulses with a frequency of 24.4 GHz, if the surface water is moving and has a minimum roughness the radar pulses are reflected with a slight frequency shift and then received by an aerial antenna from the frequency shift the sensor uses an algorithm to calculate the average flow velocity. The sensor has Tx and Rx sdi12 data lines, this fact allows us to interact with the firmware through the sdi12 ascii protocol interface. The parameters that we can set through the sdi12 interface are: Internal filter type, this parameter can be set to either IIR (Infinite Impulse Response filter) or floating mean in both cases the sensor determines approximately 20 individual measured values, depending on factors such as roughness of the water surface, wind influence, precipitation or turbulence the measured values can be scattered in this case the filter allows the measurements to be smoothed

The following equation is used to calculate the filtered speed:

$$v(f) = v(t) * Q + v(t-1) * (1-Q)$$

Where V(f): filtered velocity, v(t): current velocity, Q: constant (0.3).

Another parameter is the sensor sensitivity, a value that ranges between 10 and 46 a higher sensitivity can help getting accurate measurements when the water surface is smooth. We can also set the flow direction filter, it can be set towards the sensor, away from the sensor or deactivated. The sensor outputs the following parameters: average flow velocity, current flow velocity, sensor tilt angle to horizontal in degrees, signal quality index which is a value between 0 and 3 where 0 represents an excellent signal and 3 represents a very poor signal, vibration index which a value between 0 and three where 0 indicates no device vibration and 3 indicates very significant device vibration. The sensor also outputs an SNR value that is defined as the logarithmic ratio of the average power of the signal to the average noise power of the interfering signal.

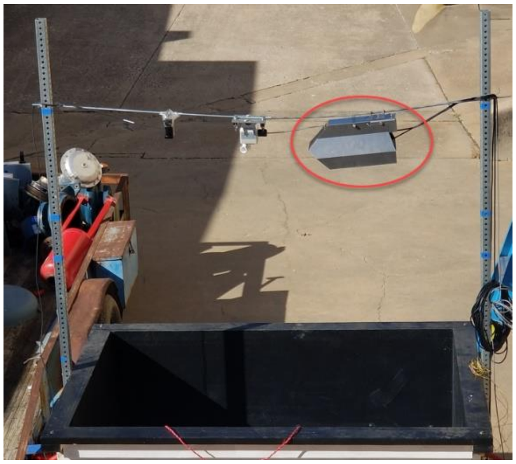
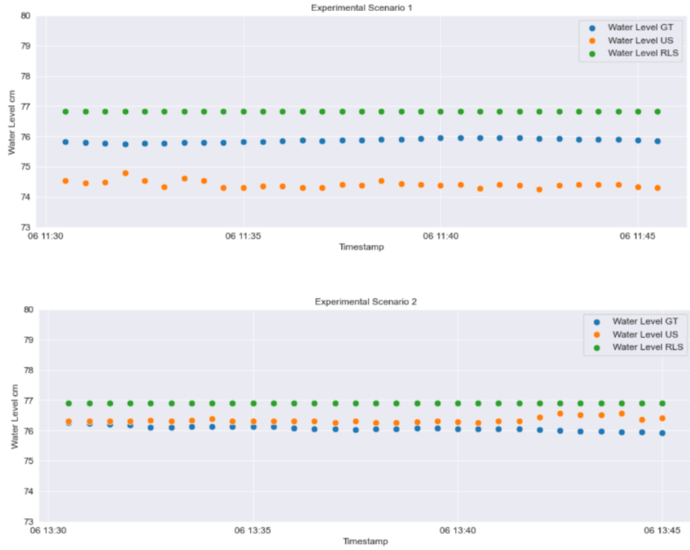



Figure 5-10. OTT SVR water surface velocity sensor mounting position

5.3.3 Experimental Scenarios and Evaluation Results

In the process of evaluating the accuracy and efficiency of these sensors, many experiments were conducted. By changing parameters such as: distance between sensors and water surface, the rate of water level change, changing the water surface and flow characteristic, we created several experimental scenarios where we compared the accuracy of the several non-contact technologies using a Mean Absolute Error (MAE) score against the ground truth measurements. Each experiment was repeated three times to ensure the repeatability of the established scenario. The duration for each experiment is 15 minutes.

In the first set of experiments, we changed the distance between the water surface and the mounted sensors. The water surface was smooth as no inlet, outlet or propellers were turned on. These experiments aim to evaluate the performance of the technologies as the distance to the surface water changes and all other parameters remain constant.



(a) **(b)**

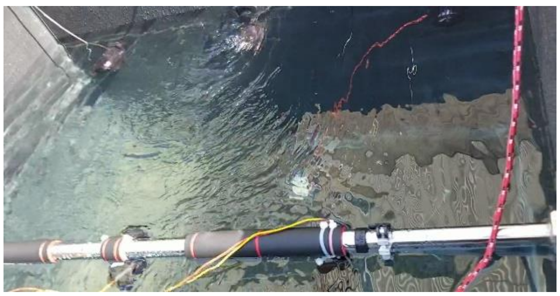
Figure 5-11. first set of experiments` water surface (a) and water level measurements (distance 171 cm) (b)

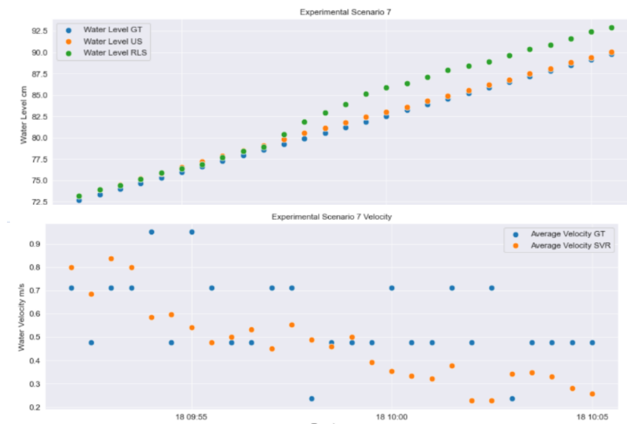
Table 5-1 shows the Mean Absolute Error for each water level measuring technology at different distances between the mounted sensors and the water surface:

Table 5-1. Results for first set of experiments (water level)

Distance between sensor and water surface (cm)	MAE Ultrasonic	MAE Radar
210 cm	2.15 cm	1.07 cm
190 cm	1.92 cm	1.05 cm
171 cm	1.45 cm	0.97 cm
149 cm	0.27 cm	0.81 cm
123 cm	0.27 cm	0.81 cm

For the second set of experiments, the level of the water was changed through inlet (ascending) and outlet (descending). Water level and water velocity measurements were recorded, experiments were conducted with and without the existence of the several water surface profiles (high energy, tidal (low energy) and pulsing mode). Figure 5-12 shows the water surface and plots for water level measurements and water velocity measurements during a water level change (ascending) experiment (distance between sensor and water level was 123 cm.





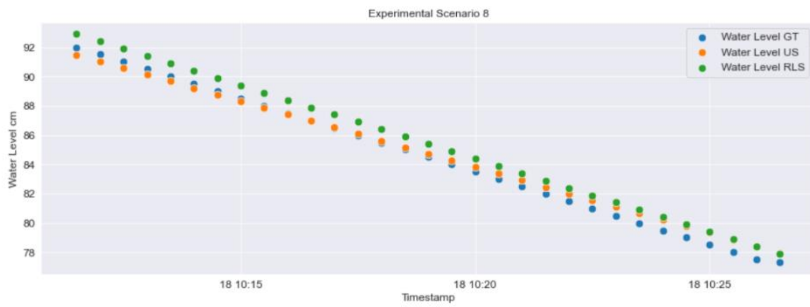
(a) **(b)**

Figure 5-12. water level change experiments water surface (a) and water level and velocity measurements (72 to 92 cm) (b)

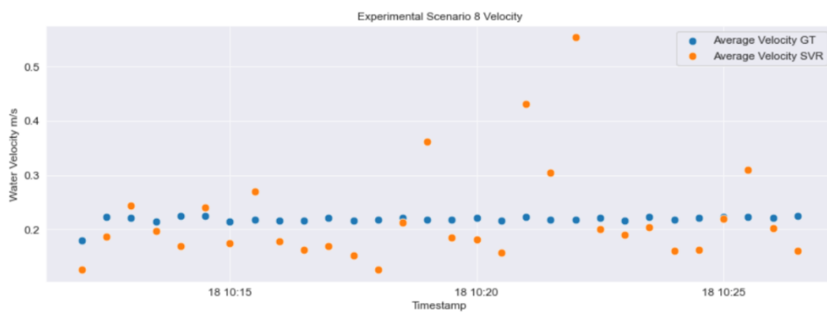
Table 5-2. Results for level change (ascending) experiments (water level and water velocity)

Distance Between Sensor and Water Surface	Water level from-to	MAE Ultrasonic Level	MAE Radar Level	MAE Radar Velocity
210 cm	72 cm to 92 cm	3.12 cm	2.13 cm	0.19 m/s
171 cm	50 cm to 65 cm	2.51 cm	2.00 cm	0.18 m/s
123 cm	65 cm to 90 cm	0.46 cm	2.00 cm	0.18 m/s

Similarly, experiments were conducted to evaluate the non-contact technologies when the water level decreased (descending). Figure 5-13 shows the plots for water level and velocity measurements.



(a)



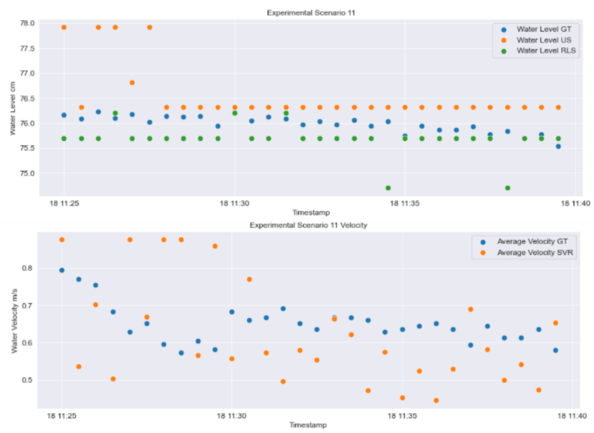
(b)

Figure 5-13. water level change experiment (descending) water level (a) and water velocity measurements (92 to 78 cm) (b)

Table 5-3. Results for level change (descending) experiments (water level and water velocity)

Distance Between Sensor and Water Surface	Water level from-to	MAE Ultrasonic Level	MAE Radar Level	MAE Radar Velocity
180 cm	92 cm to 78 cm	2.75 cm	1.02 cm	0.08 m/s
150 cm	50 cm to 65 cm	1.31 cm	1.01 cm	0.08 m/s
104 cm	92 cm to 78 cm	0.40 cm	0.89 cm	0.06 m/s

The next set of experiments were conducted to evaluate the performance of the non-contact technologies under different water surface profiles. The process for generating these different profiles were discussed in earlier reports. Figure 5-14 shows the high energy water surface profile. A high energy surface profile is an emulation of a creek surface flow under rough weather conditions.



(a) **(b)**

Figure 5-14. high energy water surface (a) and water level and velocity measurements (distance 169 cm) (b)

Table 5-4. Results for high energy water surface experiments

Distance between sensor and water surface (cm)	MAE Ultrasonic Level	MAE Radar Level	MAE Radar Velocity
190 cm	2.03 cm	0.81 cm	0.14 m/s
169 cm	1.56 cm	0.64 cm	0.12 m/s
123 cm	0.35 cm	0.56 cm	0.12 m/s

Next, experiments were conducted using a low energy water surface profile. A low energy water surface profile is an emulation for a creek surface flow under normal weather conditions. Figure 5-15 shows the water surface profile and the water surface and velocity measurements.

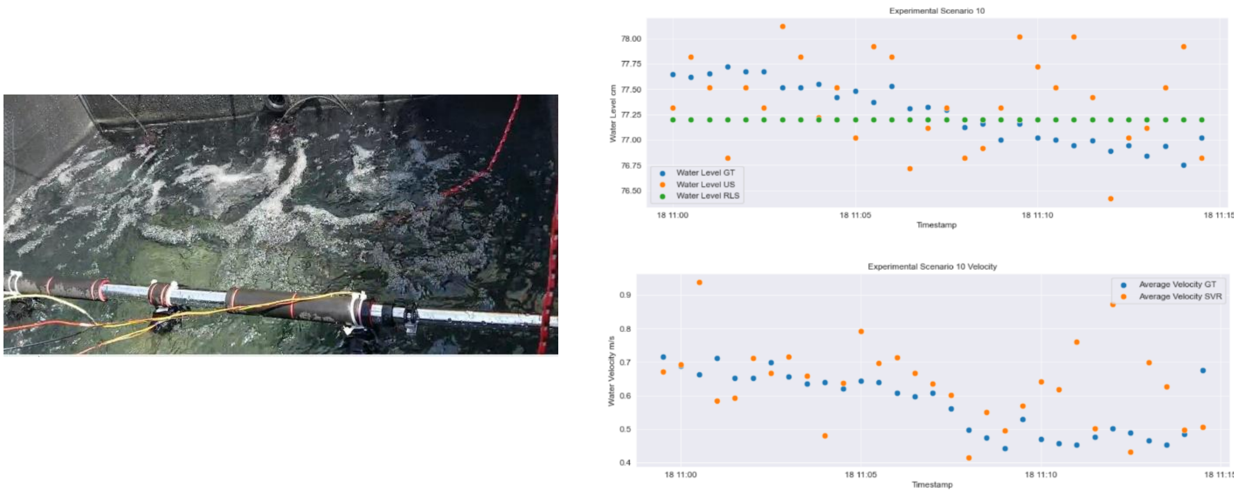
(a) **(b)**

Figure 5-15. low energy water surface (a) and water level and velocity measurements (distance 169 cm) (b)

Table 5-5. Results for low energy water surface experiments

Distance between sensor and water surface (cm)	MAE Ultrasonic Level	MAE Radar Level	MAE Radar Velocity
190 cm	1.66 cm	0.42 cm	0.08 m/s
169 cm	1.07 cm	0.41 cm	0.07 m/s
123 cm	0.20 cm	0.41 cm	0.07 m/s

Tidal (pulse) is another water surface flow profile that was used in the process of evaluating the non-contact surface and velocity technologies. A tidal flow emulates the water surface flow of a river. Figure 5-16 shows the water surface profile and the water surface and velocity measurements.



(a) **(b)**

Figure 5-16. Tidal water flow surface (a) and water level and velocity measurements (b)

Table 5-6. Results for tidal water surface experiments

Distance between sensor and water surface (cm)	MAE Ultrasonic Level	MAE Radar Level	MAE Radar Velocity
190 cm	0.98 cm	0.27 cm	0.10 m/s
169 cm	0.42 cm	0.27 cm	0.10 m/s
123 cm	0.21 cm	0.25 cm	0.09 m/s

Similarly, several experiments were conducted using different tilt angles of the surface velocity radar sensor. Depending on the installation height and the limitation of the water body area a range between 40 and 70 degrees was tested and all angles had similar MAE values. Based on the several sets of experiments detailed above we conclude the following for the water level measuring technologies: Both radar and ultrasonic technologies had an acceptable error rate (less than %5) in all cases. Ultrasonic technology proved to be more accurate when the distance between the mounted sensor and the water surface was relatively close (less than 1.5 m). Radar technology displayed higher accuracy and more stability when increasing the distance between the sensors and the water surface. Disturbing the water surface flow threw off the accuracy of the ultrasonic technology and did not seem to have a significant effect on the accuracy of the radar level sensing technology. In terms of power consumption, the radar technology is more efficient as it only consumes 50 mA only when a measuring cycle is activated compared to 80 mA for the ultrasonic water level sensing technology. The radio level sensor outputs an SNR index value which is a good indicator to tell if the sensor is installed correctly of if something is blocking its measuring area.

For the water velocity measuring technique: The Surface Velocity Radar (SVR) showed consistent and good accuracy under all different experimental scenarios. As its accuracy is affected by the tilt angle and the footprint of its measuring surface, having an embedded tilt and vibration sensor alongside a signal quality index and an SNR index is very handful to determine the correct installation of the sensor and is important to consider when developing an alternative solution.

6 Conclusion and Future Work

The work presented in this thesis detailed a variety of analyses aimed at investigating flood monitoring and detection techniques and technologies for the purpose of developing and deploying several stations across the state of Oklahoma at locations where historical flood events occurred. First, the study explored current flood detection and alerting systems in the market. We detailed the advantages and disadvantages of each system. Next, a Machine Learning based 1-hour flood forecasting system was developed using historical weather data from RWIS and historical flood events from the ODOT and the NWS. The most accurate model was a RF ensemble model with %85 accuracy forecasting flood event and %92 AUC score. The study then introduced a novel approach for evaluating non-contact flood detection technologies. An open channel testbed was designed and developed, and several non-contact water level and velocity technologies were evaluated under different conditions. The testbed can help in developing new water surface sensing technologies. Results showed that radar technology had the most accurate readings with a MAE score of around 2 cm and MAE of 0.1 m/s. We believe that collecting rain rate and water level combined will give a high accuracy prediction for floods. However, this information must be obtained during a precipitation event. Therefore, the system cannot issue warning before the rain occurs. To enhance the prediction duration, Doppler Radars can track storms and issue an earlier warning of upcoming storms in the direction of the location site. The Next Generation Weather Radar (NEXRAD) has 160 sites throughout the United States. Computer processing generates numerous analyses as level-III data. These data can provide us with important data such as the number of storm cells in a specific geometrical region, the storms total dBZ which indicates the reflectivity of factor Z of the radar signal, direction of the storm cell from the radar, speed of the storm and the traits as hail and rotating detection. These information with the ground sensors implemented will allow building longer duration forecasting models.

References

[1] "Climate of Oklahoma." *Oklahoma Climatological Survey*, Oklahoma Climatological Survey, climate.ok.gov/index.php/site/page/climate_of_oklahoma.

[2] US Department of Commerce, NOAA. "Types of Thunderstorms." *NWS JetStream*, NOAA's National Weather Service, 12 Aug. 2019, www.weather.gov/jetstream/tstrmtypes.

[3] US Department of Commerce, NOAA. *Daily Historical Weather Information for the NWS Norman Forecast Area*, NOAA's National Weather Service, 14 Apr. 2020, www.weather.gov/oun/wxhistory.

[4] Wang, Z.; Lai, C.; Chen, X.; Yang, B.; Zhao, S.; Bai, X. Flood hazard risk assessment model based on random forest. J. Hydrol. 2015, 527, 1130–1141.

[5] M. Khalaf et al., "A Data Science Methodology Based on Machine Learning Algorithms for Flood Severity Prediction," 2018 IEEE Congress on Evolutionary Computation (CEC), 2018, pp. 1-8, doi: 10.1109/CEC.2018.8477904.

[6] Mosavi, Amir, Pinar Ozturk, and Kwok-wing Chau. "Flood Prediction Using Machine Learning Models: Literature Review." Water (Basel) 10.11 (2018): 1536. Web.

[7] M. S. Yadnya and I. W. Sudiartha, "Measurement of drop size distribution rain in mataram utilize disdrometer acoustic for floods prediction," 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA), Lombok, 2016, pp. 107-110, doi: 10.1109/ISITIA.2016.7828642.

[8] C. Lai, Y. Lo, J. Yur and C. Chuang, "Application of Fiber Bragg Grating Level Sensor and Fabry-Pérot Pressure Sensor to Simultaneous Measurement of Liquid Level and Specific Gravity," in IEEE Sensors Journal, vol. 12, no. 4, pp. 827-831, April 2012, doi: 10.1109/JSEN.2011.2161075.

[9] Basha, Elizabeth & Ravela, Sai & Rus, Daniela. (2008). Model-based monitoring for early warning flood detection. SenSys'08 - Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems. 295-308. 10.1145/1460412.1460442.

[10] M. Mousa, X. Zhang and C. Claudel, "Flash Flood Detection in Urban Cities Using Ultrasonic and Infrared Sensors," in IEEE Sensors Journal, vol. 16, no. 19, pp. 7204-7216, Oct.1, 2016, doi: 10.1109/JSEN.2016.2592359.

[11] Lin, Yan-Ting & Lin, Yi-Chun & Han, Jen-Yu. (2018). Automatic Water-Level Detection Using Single-Camera Images with Varied Poses. Measurement. 127. 10.1016/j.measurement.2018.05.100.

[12] B. Basnyat, N. Roy and A. Gangopadhyay, "A Flash Flood Categorization System Using Scene-Text Recognition," 2018 IEEE International Conference on Smart Computing (SMARTCOMP), Taormina, 2018, pp. 147-154, doi: 10.1109/SMARTCOMP.2018.00085.

[13] A. Filonenko, Wahyono, D. C. Hernández, D. Seo and K. Jo, "Real-time flood detection for video surveillance," IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society, Yokohama, 2015, pp. 004082-004085, doi: 10.1109/IECON.2015.7392736.

[14] A. H. Özcan and C. Ünsalan, "LiDAR Data Filtering and DTM Generation Using Empirical Mode Decomposition," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 1, pp. 360-371, Jan. 2017, doi: 10.1109/JSTARS.2016.2543464.

[15] Fernandez-Diaz, Juan & Carter, William & Shrestha, R. & Glennie, Craig. (2014). Now You See It… Now You Don't: Understanding Airborne Mapping LiDAR Collection and Data Product Generation for Archaeological Research in Mesoamerica. Remote Sensing. 6. 9951-10001. 10.3390/rs6109951.

[16] Paul, J. D., Buytaert, W., & Sah, N. (2020). A technical evaluation of lidar-based measurement of river water levels. Water Resources Research, 56, e2019WR026810. https://doi.org/10.1029/2019WR026810.

[17] Tamari, Serge & Guerrero-Meza, Vicente. (2016). Flash Flood Monitoring with an Inclined Lidar Installed at a River Bank: Proof of Concept. Remote Sensing. 8. 834. 10.3390/rs8100834.

[18] F. Alimenti et al., "Noncontact Measurement of River Surface Velocity and Discharge Estimation With a Low-Cost Doppler Radar Sensor," in IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 7, pp. 5195-5207, July 2020, doi: 10.1109/TGRS.2020.2974185.

[19] Flood Warning Systems, Water Level Monitoring, And Alerting. Last accessed on May 28, 2022, from https://www.valarm.net/flood-warning-systems/

[20] Fondriest Environmental, Inc. "Flood Warning Systems." Fundamentals of Environmental Measurements. 17 Sep. 2015. Web. < https://www.fondriest.com/environmental-measurements/parameters/water-quality/water-temperature

[21] Traffic Flood Warning Systems: Flood Detection Systems. (n.d.). Last accessed on May 28, 2022, from https://elteccorp.com/products/warning-systems/flood-warning-systems/

[22] YSI Flood Warning System. Last accessed on May 29, 2022, from https://www.ysi.com/File%20Library/Documents/Brochures%20and%20Catalogs/E99-Flood-Alert-Monitoring-brochure.pdf

[23] Climate de Oliveira, M.M.F., de Oliveira, J.L.F., Fernandes, P.J.F. et al. Extreme climatic characteristics near the coastline of the southeast region of Brazil in the last 40 years. Theor Appl Climatol 146, 657–674 (2021). https://doiorg.ezproxy.lib.ou.edu/10.1007/s00704-021-03711-z

[24] FUHRMANN, Christopher M, and Charles E KONRAD. "A Trajectory Approach to Analyzing the Ingredients Associated with Heavy Winter Storms in Central North Carolina." Weather and Forecasting 28, no. 3 (2013): 647-67.

[25] Buzzia, R., and P. Chessa, 2002: Prediction of the U.S. storm of 24– 25 January 2000 with the ECMWF Ensemble Prediction System. Mon. Wea. Rev., 130, 1531–1551.

[26] Thies CLIMA radar weather station. Retrieved May 18, 2022, from https://www.thiesclima.com/en/Products/Miscellaneous-Devices-Weather-stations/?art=197

[27] Surface Sentinel Temperature Sensor – Model 5439-00. Retrieved May 18, 2022, from https://hsierra.com/product/surface-temperature-sensor-model-5439-00/

[28] Campbell Scientific temperature probe. Retrieved May 18, 2022, from https://www.campbellsci.com/109#:~:text=The%20109%20is%20a%20rugged,or%20submerge%20it%20in%20water

[29] Roadside Embedded Extensible Computing Equipment (REECE). Retrieved May 19, 2022, from http://itrafficsystems.com/index.html

[30] Sierra RV50X industrial LTE Router. Retrieved May 19, 2022, from https://www.sierrawireless.com/products-and-solutions/routers-gateways/rv50/

[31] Rover Li 40 Amp MPPT Solar Charge Controller. Retrieved May 19, 2022, from https://renogy.com/rover-li-40-amp-mppt-solar-charge-controller/

[32] Price, C. (2009). Thunderstorms, Lightning and Climate Change. In: Betz, H.D., Schumann, U., Laroche, P. (eds) Lightning: Principles, Instruments and Applications. Springer, Dordrecht. https://doi.org/10.1007/978-1-4020-9079-0_24

[33] Weather Underground, weather stations across the USA. Accessed May 19, 2022, from https://www.wunderground.com/wundermap

[34] L. Nováková and O. Štepanková, "RadViz and Identification of Clusters in Multidimensional Data," 2009 13th International Conference Information Visualisation, 2009, pp. 104-109, doi: 10.1109/IV.2009.103.

[35] Halko, et al. (2009). “Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions”

[36] Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, Asdrubal Lopez, “A comprehensive survey on support vector machine classification: Applications, challenges and trends”, 2020, Neurocomputing, Volume 408

[37] Williams, Ronald J.; Hinton, Geoffrey E.; Rumelhart, David E. (October 1986). "Learning representations by back-propagating errors". Nature. 323 (6088): 533–536. Bibcode:1986Natur.323..533R. doi:10.1038/323533a0. ISSN 1476-4687. S2CID 205001834.

[38] Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 6, 2 (April 1998), 107–116. https://doi.org/10.1142/S0218488598000094.

[39] Hochreiter, Sepp (1991). Untersuchungen zu dynamischen neuronalen Netzen (PDF) (diploma thesis). Technical University Munich, Institute of Computer Science, advisor: J. Schmidhuber.

[40] Schmidhuber, Jürgen (2021). "The 2010s: Our Decade of Deep Learning / Outlook on the 2020s". AI Blog. IDSIA, Switzerland. Retrieved May 19, 2022.

[41] Lee, Jae-Yeong and Kim, Ji-Sung, Detecting Areas Vulnerable to Flooding Using Hydrological-Topographic Factors and Logistic Regression, Applied Sciences, 2021, https://www.mdpi.com/2076-3417/11/12/5652

[42] Rolling Hills Research Corporation 2014 Research Water Tunnels Specification USA

[43] M Zahari and S S Dol 2015, Design and Development of Low-Cost Water Tunnel for Educational Purpose https://doi.org/10.1088/1757-899x/78/1/012040

[44] Namoto, Shinsuke & Ishikawa, Tadaharu & Kojima, Takashi. (2018). Characteristics of water surface profiles of open channel flow with levee overtopping. International Journal of Safety and Security Engineering. 8. 484-492. 10.2495/SAFE-V8-N4-484-492.

[45] Hydrogeologic Characteristics and Geospatial Analysis of the Water-Table Changes in the Lower Part of the Arkansas River. United States Geological Survey.

[46] VorTech MP60mQD - Mobius Ready QuietDrive Propeller Pump. Retrieved May 18, 2022, from https://www.bulkreefsupply.com/vortech-mp60qd-mobius-ready-quietdrive-propeller-pump-ecotech-marine.html

[47] Aqua Dream 24v Dc Sine Wave Maker Pump Marine Aquarium Coral Reef 4800 GPH [LBS-50]. Retrieved May 18, 2022, from https://aquadreamusa.com/collections/lbs-series/products/aquility-wave-maker-4800-gph-aq-lbs-50

[48] Bar02 Ultra High Resolution 10m Depth/Pressure Sensor. Retrieved May 18, 2022, from https://bluerobotics.com/store/sensors-sonars-cameras/sensors/bar02-sensor-r1-rp/

[49] DIGITEN G1" Water Flow Sensor, Food-Grade Hall Effect Sensor Flow Meter Flowmeter Counter 1-60L/min. Retrieved May 18, 2022, from https://www.digiten.shop/products/digiten-g3-4-water-flow-hall-sensor-switch-flow-meter-1-60l-min

[50] 260-700 Ultrasonic Water Depth Sensor. Retrieved May 18, 2022, from https://novalynx.com/store/pc/260-700-Ultrasonic-Snow-Depth-Sensor-p275.htm

[51] OTT RLS - Radar Level Sensor. Retrieved May 18, 2022, from https://www.ott.com/products/water-level-1/ott-rls-radar-level-sensor-861/

[52] OTT SVR – Surface Velocity Radar. Retrieved May 18, 2022, from https://www.ott.com/products/water-flow-3/ott-svr-100-2406/

Appendix A: Source Codes and Functions

1. Ground Truth Data logger

```
#include <Wire.h>
#include "MS5837.h"

MS5837 sensor;

byte statusLed    = 13;

byte leftSensorInterrupt = 0;  // 0 = digital pin 2
byte leftSensorPin       = 2;

byte middleSensorInterrupt = 1;  // 1 = digital pin 3
byte middleSensorPin       = 3;

byte rightSensorInterrupt = 4;  // 4 = digital pin 19
byte rightSensorPin       = 19;

float calibrationFactor = 0.5; // The hall-effect flow sensor outputs approximately 4.5 pulses per second per litre/minute of flow.

volatile byte left_pulseCount;  // define the pulse counters as volatile to avoid losing data as stated in the Reference
volatile byte middle_pulseCount;
volatile byte right_pulseCount;

float left_flowRate;
float middle_flowRate;
float right_flowRate;

float inlet_flowLitres;
float outlet_flowLitres;

float inlet_totalLitres;
float outlet_totalLitres;

unsigned long oldTime; // to control the one second reading interval

String dataLabel1 = "Depth";
String dataLabel2 = "Temperature";
String dataLabel3 = "Inlet_flow_rate";
String dataLabel4 = "Total_inlet_flow";
String dataLabel5 = "Outlet_flow_rate";
String dataLabel6 = "total_outlet_flow";
bool label = true;
void setup()
{

  // Initialize a serial connection for reporting values to the host
```

```
  Serial.begin(9600);
  Wire.begin();
  // set bar02 and initialize the sensor
  sensor.setModel(MS5837::MS5837_02BA);
  sensor.init();
  sensor.setFluidDensity(997); //fresh water
  // Set up the status LED line as an output
  pinMode(statusLed, OUTPUT);
  digitalWrite(statusLed, HIGH);  // to make sure the code is executed

  pinMode(inletSensorPin, INPUT);
  pinMode(outletSensorPin, INPUT);

  digitalWrite(inletSensorPin, HIGH);
  digitalWrite(outletSensorPin, HIGH);

// initialize parameters to calculate inlet flow
  inlet_pulseCount= 0;
  inlet_flowRate= 0.0;
  inlet_flowLitres= 0;
  inlet_totalLitres=0;

// initialize parameters to calculate outlet flow
  outlet_pulseCount= 0;
  outlet_flowRate= 0.0;
  outlet_flowLitres= 0;
  outlet_totalLitres=0;

  oldTime= 0;

  // The inlet sensor is connected to pin 2 which uses interrupt 0.
  attachInterrupt(inletSensorInterrupt, inlet_pulseCounter, FALLING);
  // The outlet sensor is connected to pin 3 which uses interrupt 1.
  attachInterrupt(outletSensorInterrupt, outlet_pulseCounter, FALLING);
}

void loop() // main loop
{

   if((millis() - oldTime) > 1000)    // Only process counters once per
second
  {
    // Disable the inlet interrupt while calculating flow rate and sending
the value to the host
    detachInterrupt(inletSensorInterrupt);
    detachInterrupt(outletSensorInterrupt);
    inlet_flowRate = ((1000.0 / (millis() - oldTime)) * inlet_pulseCount) /
calibrationFactor;
    outlet_flowRate = ((1000.0 / (millis() - oldTime)) * outlet_pulseCount)
/ calibrationFactor;
    oldTime = millis();
```

```
    // Divide the flow rate in litres/minute by 60 to determine how many
litres have passed through the sensor in this 1 second interval,
    inlet_flowLitres = (inlet_flowRate / 60);
    outlet_flowLitres = (outlet_flowRate / 60);

    // Add the litres passed in this second to the cumulative total
    inlet_totalLitres += inlet_flowLitres;
    outlet_totalLitres += outlet_flowLitres;
    sensor.read();

    // Print the flow rate for this second in litres / minute
    //print out column headers
    while(label){ //runs once
        Serial.print(dataLabel1);
        Serial.print(",");
        Serial.print(dataLabel2);
        Serial.print(",");
        Serial.print(dataLabel3);
        Serial.print(",");
        Serial.print(dataLabel4);
        Serial.print(",");
        Serial.print(dataLabel5);
        Serial.print(",");
        Serial.println(dataLabel6);
        label=false;
  }
    Serial.print(sensor.depth(), 4);
    Serial.print(",");
    Serial.print(sensor.temperature(), 4);
    Serial.print(",");
    Serial.print(inlet_flowRate, 4);
    Serial.print(",");
    Serial.print(inlet_totalLitres, 4);
    Serial.print(",");
    Serial.print(outlet_flowRate, 4);
    Serial.print(",");
    Serial.println(outlet_totalLitres, 4);

    // Reset the pulse counters
    inlet_pulseCount = 0;
    outlet_pulseCount = 0;

  // Enable the interrupt
  attachInterrupt(inletSensorInterrupt, inlet_pulseCounter, FALLING);
  attachInterrupt(outletSensorInterrupt, outlet_pulseCounter, FALLING);

  }
}


//Insterrupt Service Routines

void inlet_pulseCounter()
```

```
{
  inlet_pulseCount++; // Increment the inlet pulse counter
}
void outlet_pulseCounter()
{
  outlet_pulseCount++; // Increment the outlet pulse counter
}
```

2. RLS Data Logger

```
timeout_start = time.time()
while True:
    try:
        ser_RLS.write(b'run sdi recorder\r\n')
        y = str(ser_RLS.readline())
        if y[6:9] == "ACK":
            print("Converter Acknowledged")
        else:
            print("Converter Error")
        # read level sensor
        ser_RLS.write(b'0M!\r\n')
        if ser_RLS.inWaiting():
            x = str(ser_RLS.readline())
            print("Number of seconds for measurement: " + x[7:10] )
            print("Number of measured values: " + x[10:11] )
            ser_RLS.write(b'0D0!\r\n')
            getData=str(ser_RLS.readline())
            print(getData)
            file = open(fileName, "a")
            timestamp = str(time.time())
            timestamp = str(datetime.datetime.fromtimestamp(time.time()).strftime('%Y-%m-%d %H:%M:%S'))
            print(timestamp + ',' + getData)
            file.write(timestamp + ',' + getData + "\n")
            if time.time() > timeout_start + 900:
                file.close()
                ser_groundtruth.close()
                break
            ser_RLS.write(b'0M1!\r\n')
            z = str(ser_RLS.readline())
            print("time in seconds until the sensor makes the status available: "+ z[7:10])
            ser_RLS.write(b'0D0!\r\n')
            t = str(ser_RLS.readline())
            print("status: "+ t[8:9])
            SNR = t[10:13]
            print("SNR: "+ SNR)
    except:
        print("Keyboard Interrupt")
        file.close()
        ser_RLS.close()
        break
```

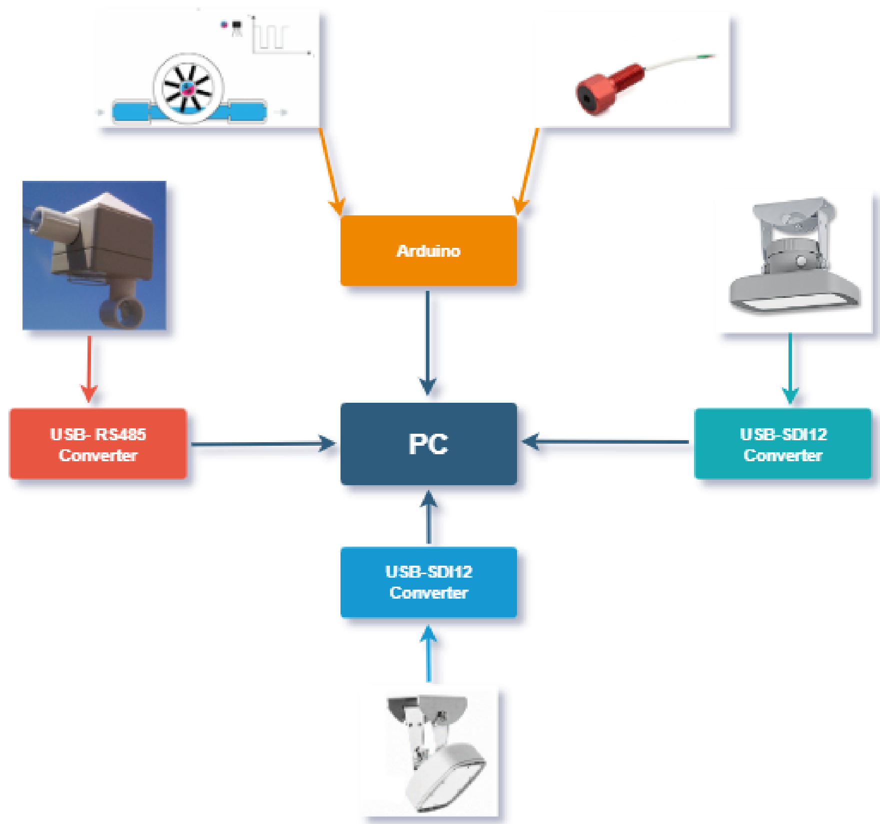
3. RWIS Data Parser

```
#station_id = "35ST058" D_from = "2020-10-15" D_to = "2020-10-16"
def API_to_DataFrame(station_id,D_from,D_to):
    link = "https://rwis.tulsa.ou.edu/rwis/api/weatherAll/data?"
    station = "station="
    add = "&"
    date_from = "date_from="
    date_to = "date_to="
    message = "".join([link, station,station_id,add,date_from,D_from,add,date_to,D_to])
    Json_data = requests.get(message).json()
    n= len(Json_data)
    station = []
    date_time = []
    wind_speed = []
    wind_direction = []
    gust_wind_speed = []
    gust_wind_direction = []
    temperature = []
    air_pressure = []
    precipitation_event = []
    precipitation_intensity = []
    total_precipitation = []
    precipitation_type = []
    brightness = []
    temp1_surface_temperature = []
    temp1_dew_point = []
    temp2_surface_temperature = []
    temp_probe_1 = []
    temp_probe_2 = []

    for x in range(n):
        station.append(Json_data[x]['station'])
        date_time.append(Json_data[x]['date_time'])
        wind_speed.append(Json_data[x]['wind_speed'])
        wind_direction.append(Json_data[x]['wind_direction'])
        gust_wind_speed.append(Json_data[x]['gust_wind_speed'])
        gust_wind_direction.append(Json_data[x]['gust_wind_direction'])
        temperature.append(Json_data[x]['temperature'])
        air_pressure.append(Json_data[x]['air_pressure'])
        precipitation_event.append(Json_data[x]['precipitation_event'])
        precipitation_intensity.append(Json_data[x]['precipitation_intensity'])
        total_precipitation.append(Json_data[x]['total_precipitation'])
        precipitation_type.append(Json_data[x]['precipitation_type'])
        brightness.append(Json_data[x]['brightness'])
        temp1_surface_temperature.append(Json_data[x]['temp1_surface_temperature'])
        temp1_dew_point.append(Json_data[x]['temp1_dew_point'])
        temp2_surface_temperature.append(Json_data[x]['temp2_surface_temperature'])
        temp_probe_1.append(Json_data[x]['temp_probe_1'])
        temp_probe_2.append(Json_data[x]['temp_probe_2'])
    RWIS_Dataframe = pd.DataFrame({'Station':station,'Date':date_time,'Wind Speed':wind_speed,
                               'Wind Direction':wind_direction,'Gust Wind Speed':gust_wind_
                          'Gust Wind Direction':gust_wind_direction,'Temperature':temperature,
                           'Air Pressure':air_pressure,
                          'Precipitation event':precipitation_event,
                          'Precipication Intensity':precipitation_intensity,
                          'Total Precipitation':total_precipitation,
                          'Precipitation type':precipitation_type,'Brghtness':brightness,
                          'Road Surface Temp':temp1_surface_temperature,
                           'Due Point Temp':temp1_dew_point,'Bridge Surface Temp':temp2_surface
                          '2 inch Underground Temp ':temp_probe_1,'6 inch Underground Temp':tem

    return RWIS_Dataframe
```

Appendix B: Emulator System Design and Setup



The ground truth sensing mechanism is set using a pressure sensor (bar02) for water depth recording and a Hall effect flow sensor for water velocity recording. The bar02 is controlled by an Arduino nano microprocessor, it is important to use a microprocessor that can handle 3.3V logic input to provide accurate measurements. The sensor interfaces through I2C communication protocol and the microprocessor is connected to the PC through USB serial. The hall effect water flow sensor communication is established through digital interrupts, so it is important to use a microcontroller that has a minimum of 5 interrupt pins; in this setup an Arduino Mega was used. For the Novalyx Ultrasonic water level sensor an RS-485 to USB converter was used to connect the sensor to the PC. Communication with the sensor is established using Modbus ASCII protocol.

Both the OTT SVR and the OTT RLS were connected to the PC using an SDI12 to USB converter.

The sensors are then interfaced through the PC using Modbus ASCII communication protocol.