Hydrologic Modeling and System Optimization for IoT Flood Management

A Technical Report submitted to the Department of Systems Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Arnold Mai

Spring, 2022

Technical Project Team Members

Andrew Bowman

Nicholas Khattar

Lily Malinowski

Khwanjira Phumphid

Taja Washington

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Jonathan L. Goodall, Department of Civil Engineering

Hydrologic Modeling and System Optimization for IoT Flood Management

***Abstract*— The increasing frequency and severity of storms due to climate change is magnifying flooding impacts. The Internet of Things (IoT) revolution promises more ubiquitous sensing capabilities. When applied to water systems, IoT has the potential to increase insights into how hydrologic systems respond to extreme rainfall events, aiding in emergency management efforts before and during extreme weather events. In this paper, we provide a way to translate forecasted extreme rainfall events into flood impacts and optimize an IoT sensor network for real-time flood monitoring. First, we created a hydrologic model for a study area: the Dell Pond watershed in Charlottesville, Virginia. We used ArcGIS to obtain parameters for the model from geospatial datasets such as elevation, soils, land use, and land cover. The parameters obtained from ArcGIS, alongside the National Oceanic and Atmospheric Administration (NOAA) rainfall precipitation data, and readings from the IoT water sensors were combined to create a hydrologic model in HEC-HMS. To optimize the IoT sensor monitoring network and explore systems integration of the model and sensors, we first created models to determine the battery life of a sensor in the network, since the IoT sensors are battery powered with no additional power harvesting capability. We also deployed a new water level and a soil moisture sensor using the IoT network for the study watershed. The methods for estimating the battery life of the IoT sensor and the prototype deployment can be built on in future research to advance next-generation flood management systems that integrate computational models and IoT monitoring networks.**

Nicolas Khattar, Taja M Washington, Arnold Mai, Lili Malinowski, Andrew N. Bowman, Khwanjira Phumphid, Victor A Leal Sobral, Jonathan L. Goodall, University of Virginia

I. Introduction

Climate change is projected to cause temperature and rainfall changes in coming years. The frequency and severity of storms is already showing evidence of increasing trends, magnifying flooding risk. From 1995 to 2015, Charlottesville and Albemarle County, the study area for this research, experienced around one hundred floods that created more than a million dollars in damage [1]. Even minor flooding can lead to devasting results for the community, such as school and road closures. More significant flooding can create safety hazards, as floods can cause power outages and damage infrastructure and assets which halts economic activity. Floods can also be lethal to the lives of every person living in damp building conditions, due to the development of mold, diminishing indoor air quality, which could lead to respiratory tract irritation and infections, including pneumonia.

Climate change effects, including flooding impacts, are a worldwide concern. According to data from the World Resources Institute, by 2030, the number of people affected by floods in the world will double and triple by 2050. In the United States, climate change will cause flooding losses to jump more than 26 percent over the next three decades, with disadvantaged communities shouldering an outsize share of the economic burden [2]. Steps need to be taken to prevent disastrous losses from further escalating.

Without proper monitoring systems, flooding will cause economic loss, social disruptions, and damage to the urban environment. In response to flooding, cities and towns around the world are currently looking for a precautionary measure to minimize the adverse effects. In view of the frequency and severity of floods, many technological companies are using the Internet of Things (IoT) and flood forecasting models to propose flood monitoring and early detection systems that allow administrations to prepare for floods in advance. IoT is a crucial part in the development of smart cities because it promises improved environmental sensing capabilities. Among them flood monitoring systems with sensors are widely used.

Although IoT solutions and advanced modeling cannot prevent flooding, real-time monitoring and modeling can help minimize potential damage by building initiative-taking solutions for the community. When applied to water systems, IoT coupled with models has the potential to increase insights into how hydrologic systems respond to extreme rainfall events, aiding in emergency management efforts before and during extreme weather events.

II. Methods

A. System Architecture

The flood monitoring system uses an IoT network which includes water level, soil moisture, and weather sensors from Decentlab. They communicate via LoRaWAN, a Low Power, Wide Area networking protocol. LoRaWAN Cisco gateways are used to connect the sensors to a LoRaWAN network server, The Things Network (TTN). The data generated from the sensors is stored in Google BigQuery, a data warehouse and finally, it is visualized using Grafana, an open-source data visualization platform. The sensor data will be used along with a hydrologic model to forecast water levels for flood management decision-makers. The envisioned flood warning system is depicted in the following block diagram (Fig. 1). Green blocks represent the real-time IoT network, data storage and visualization. Blue blocks represent the hydrologic system model and forecast. Red block represents the alerting tools integration with stakeholders.

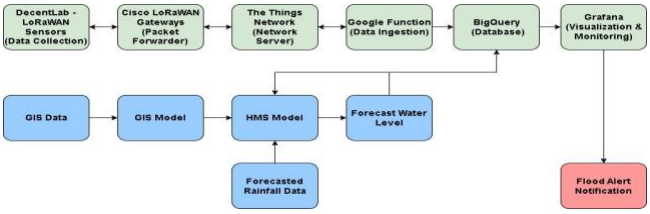

Fig. 1. The envisioned flood warning system block diagram.

B. Hydrologic Model Development

We used ArcGIS to replicate the Dell Pond watershed (referred to as the Dell watershed henceforth) and obtain parameters needed for the hydrological model in HEC-HMS. First, a digital elevation model was obtained from the United States Geological Survey [3] and used to create the stream network of the watershed, utilizing tools such as Flow Direction and Flow Accumulation, then translating the resultant raster into a network of polylines. The watershed, as seen in Fig. 2, was created through snapping a pour point onto the streamline near the outlet.

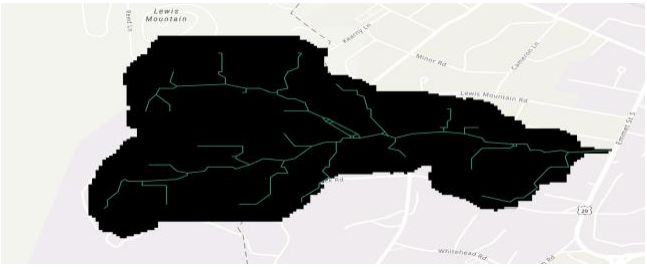

Fig. 2. Watershed of the Dell in ArcGIS

The parameters needed for the hydrological model in HEC-HMS included curve number, longest stream length, and average slope. Land cover data was obtained from the Chesapeake Bay Program Land Use/Land Cover Data Project [4]. Soil data was obtained from the Soil Survey Geographic Database [5]. To create the most accurate model, the watershed was split into five subwatersheds based on similar land cover, as seen in Fig. 3.

Fig. 3. Land Cover of the Dell Watershed

Curve numbers for each subwatershed were calculated by combining land cover and soil data through the Union tool, exporting the resultant table to Excel, sorting areas by land cover, and summing land cover areas based on hydrologic soil group. The HEC-HMS Technical Reference Manual [6] was used to assign curve numbers based on hydrologic soil group. The following equation was used to calculate the composite curve number of each subwatershed,

$$CN_{composite} = \frac{\sum A_i CN_i}{\sum A_i} \quad (1)$$

where A_i represents the individual areas and CN_i refers to the individual curve number associated with the area. The longest stream length for each subwatershed was found by using the measure tool in ArcGIS. The average slope for each subwatershed was calculated using the Slope tool. The Dell watershed was then modeled in HEC-HMS with the pre-existing parameters found in ArcGIS, as seen in Fig. 4. The five subbasins were inputted into the model and connected by a singular junction. To account for an overflow structure located in the stream network, a diversion was placed in the model, which diverts 10 cfs of water to the sink. This number can be changed based on the storm, but we did not change it for these simulations. For the model, the runoff from Subbasin 5 and Reach 5 are directed towards the diversion which then releases a small amount to the sink, so that it does not overflow. This was based on information given by the University's Facilities Management. The frequency storm simulated was a 10 year, 24-hour storm with a 15-minute intensity, and the data points were collected using local data provided by NOAA.

Fig. 4. HEC-HMS Model of the Dell Watershed

C. IoT Device Management

To explore IoT and systems integration, we determine the battery life of the IoT sensor devices by analyzing the battery voltage over time. The batteries studied are from the weather station (Model DL-ATM41) because it had a complete battery drawdown record. Two alkaline type C batteries are used in the sensor (Model LR14). An important variable to consider that affects the battery life is the spreading factor, which is the speed at which the signal frequency changes across the bandwidth of a channel. The estimation for the weather station battery life given from Dencentlab for a spreading factor of seven with a 10-minute interval between messages is 10.2 months. [7] Our weather station's spreading factor is automatically set by TTN using the Adaptive Data Rate (ADR), which is a mechanism that controls the data rate. Since our sensor is close to the gateway (~15 meters away), 95.4 % of the recorded messages had a spreading factor of 7, except for short periods where it was 8 (4.1%) and 9 (0.5%) due to atypical events such as intense rain. We used two methods to evaluate battery life using models with varying levels of complexity. Starting with the simpler approach, we fitted the data using linear and polynomial

regression models using Python, then we created a time series model using RStudio.

The regression model is based on a first and third order polynomial fit. The data of the weather station was aggregated and downloaded from Grafana, then imported to Python as a CSV file. Since we did not have a continuous battery voltage record from full to depleted battery, we combined the readings of the end part of a previous discharge curve with more recent battery voltage readings in such a way that we could emulate a full discharge cycle. The date format was normalized and converted from DD/MM/YYYY HH:MM to hours using Excel by taking the difference between the current time and the first reading and multiplying by twenty-four. The following libraries were imported: Matplotlib was used to graph the data and predict the values, NumPy was used to convert the CSV data to arrays, and Scikit-learn was used to generate the regression models. Several polynomial models were created using the "LinearRegression" function and the curve fit was evaluated using the Mean Absolute Error (MAE). To get the equation for predicting the battery voltage at any given time, we printed the model coefficients of the best order polynomial fit and filled it manually in a cubic function of time.

To better model variation in the battery data over shorter time periods, time series analysis was used. The time series analysis used data collected starting when the battery was at full charge and contained 674 data points across 170 days. The time series data points were obtained by averaging battery voltage readings from the weather station, resulting in a rate of four data points per day. The battery voltage readings were aggregated, downloaded from Grafana, and then imported into RStudio. The data across each day was averaged to create a resampled time series object with consistent intervals. From the full data, 160 days' worth of data were used to build the model, with the remaining 10 days of data being used to compare to the model's forecasted values. Using the "forecast" package in RStudio, an ARIMA (0,2,1) model was created with the objective of minimizing the Akaike Information Criterion (AIC). This model uses the second difference of the non-stationary time series data, to create a stationary time series. This was necessary due to the downward trend inherent in the battery data. The model yielded by our time series analysis contains one moving average component. Then, the model was used to predict the next 10 days of voltage values, and then compared to the actual data for assessment, and the mean squared error (MSE) was calculated.

D. IoT Device Deployment

To prototype an IoT sensor deployment, we deployed a pressure sensor (model DL-PR26) and soil moisture sensor (model DL-SMTP). We also installed a LoRa gateway within range of the Dell Pond in Charlottesville and connected the sensors to TTN and Grafana. The unit of the pressure sensor is Bar. The unit of the soil moisture data is Scaled Frequency Units (SFu). Its subsurface probe measures soil moisture and temperature at eight distinct levels of depth.

III. RESULTS

A. Hydrologic Model Simulations

As mentioned, the HEC-HMS model has five subbasins and five reaches with a junction point and a diversion to represent the overflow structure. The parameters for each subbasin are area, curve number, imperviousness, and lag time (Table I).

TABLE I. SUBBASIN PARAMETER INPUTS

Subbasin	Area (mi^2)	Curve number	% impervious	Lag time (min)
1	0.0514	53	6.49	22.3
2	0.0595	52	11.1	29.4
3	0.0283	72	14.7	37.8
4	0.0438	70	47.8	24.6
5	0.0870	78	42.0	21.6

The five reaches represent the longest time traveled within the Dell watershed, and the lag time for each are listed in Table II below.

TABLE II. REACH LAG TIMES

Reaches	Lag time (min)
1	27.1
2	29.4
3	53.1
4	30.7
5	31.9

A simulation of the model was produced with a five year 24-hour storm with 15-minute intensity, and overall, there were 69.2 cubic feet per second (cfs) diverted from the watershed into the outlet structure leaving 10 cfs to flow into Dell Pond, the sink. Runoff from the subbasins 1-3 were all similar being 9.1, 12.0, and 13.3 cfs, respectively. The peak discharge from subbasins 4 and 5 were 29.1 and 63.1 cfs respectively (Table III). The increase of runoff in subbasins 3 and 5 may be due to several factors including the increased curve number and % impervious. The volume of the runoff seems to correspond with the peak discharge except for subbasin 3. This may be due to the basin's curve number and lag time.

TABLE III. FIVE YEARS STORM SUBBASIN RESULTS

Subbasins	Peak Discharge (CFS)	Volume (in)
1	9.1	0.98
2	12.0	1.11
3	13.3	2.36
4	29.1	3.19
5	63.1	3.39
Diversion	69.2	1.23

A simulation of the model was produced with a 10-year 24-hour storm with 15-minute intensity, and overall, there were 78 cfs diverted from the watershed into the outlet structure leaving 10 cfs to flow into Dell Pond, the sink. Runoff from the subbasins 1-3 were all similar being 14.3, 17.9, and 17.2 cfs, respectively. The peak discharge from subbasins 4 and 5 were 35.7 and 77.0 cfs respectively (Table IV).

TABLE IV. TEN YEARS STORM SUBBASIN RESULTS

Subbasins	Peak Discharge (CFS)	Volume (in)
1	14.3	1.41
2	17.9	1.55
3	17.2	3.04
4	35.7	3.92
5	77.0	4.16
Diversion	88.1	1.77

A simulation of the model was produced with a 25-yearr 24-hour storm with 15-minute intensity, and overall, there were 114.3 cfs diverted from the watershed into the outlet structure leaving 10 cfs to flow into Dell Pond, the sink. Runoff from the subbasins 1-3 were all similar being 22.6, 27.3, and 22.6 cfs, respectively. The peak discharge from subbasins 4 and 5 were 44.7 and 95.2 cfs respectively (Table V).

TABLE V. 25 YEAR STORM SUBBASIN RESULTS

Subbasins	Peak Discharge (CFS)	Volume (in)
1	22.6	2.11
2	27.3	2.26
3	22.6	4.07
4	44.7	5.01
5	95.2	5.30
Diversion	114.3	2.65

Looking across all three storms, the peak discharge increased 27.3% from the 5-year to the 10-year storm, 29.7% from the 10-year to the 25-year storm, and 65.2% from the 5-year to the 25-year storm. While all the diversions only allowed 10 cfs to go to the sink, they all occurred at different times. In Fig. 5, the peak diversion occurred at the 11th hour of the storm. In Fig. 6, the peak diversion occurred at the 10th hour of the storm, and in Fig. 7, it is shown that the peak diversion occurred at the ninth hour of the storm. Because of this outlet structure, the Dell Pond should not overflow, and this is what these simulations illustrated.

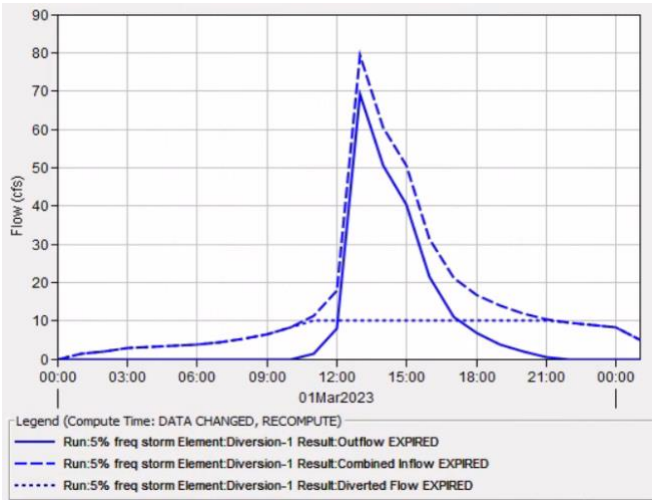

Fig. 5. Five Years Storm Diversion Hydrograph

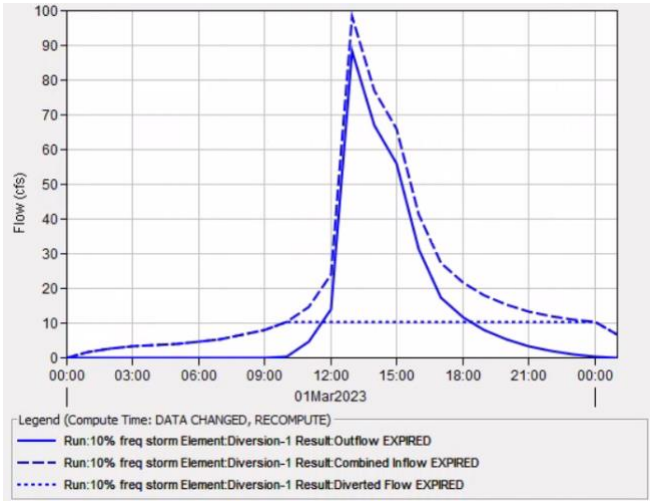

Fig. 6. Ten Years Storm Diversion Hydrograph

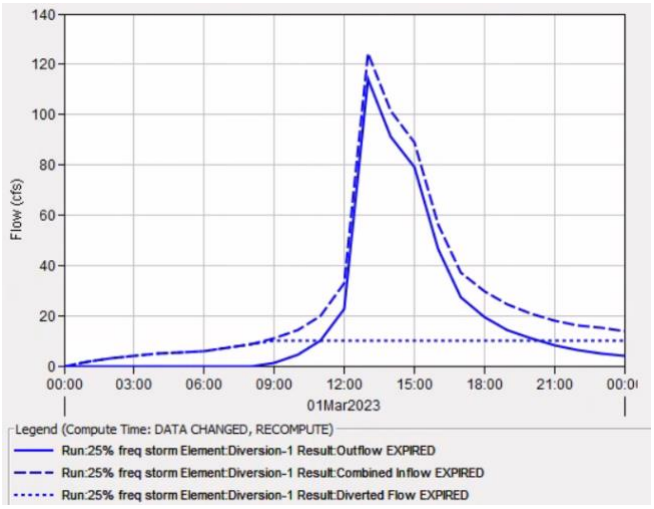

Fig. 7. Twenty-Five Years Storm Diversion Hydrograph

B. IoT Energy Model

After acquiring voltage data from the weather sensor via Grafana, two models were created to predict the voltage as a function of time to better estimate when a given sensor's battery would need to be replaced. This would allow better understanding of how the battery discharges and the behavior of the battery when it discharges. Although, the recommended time to replace the battery is 2.0V, the last message sent from the weather station was at 2.1V, thus we assume the battery should be replaced at the latest at 2.1V.

In Python, regression models of first and third orders (Fig. 8) were applied to the voltage data to foresee when the voltage reaches 2.1V and the best time to replace the battery. It was more appropriate to use the third order because it has the lowest MAE value of 0.0126 compared to the MAE of the first order which has a value of 0.0313. We did not choose to keep increasing the order of regression because after the third order, the MAE does not substantially decrease, for example, the MAE of the fourth order is 0.0115. Thus, we adopt a third order polynomial model as a reasonable tradeoff between complexity and precision.

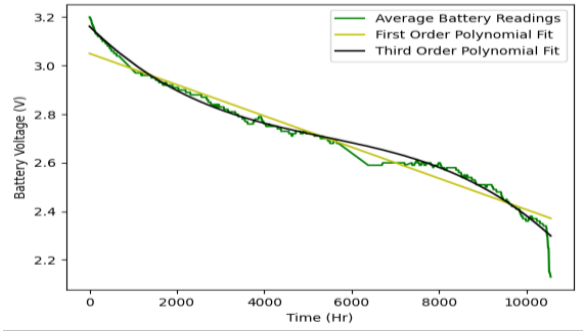

Fig. 8. Battery Voltage as a Function of Time with a 1st and 3rd. Order Regression

$$y = -6.43 \times 10^{(-5)}x + 3.05 \quad (2)$$

$$y = -1.58 \times 10^{(-4)}x^{(3)} + 2.57 \times 10^{(-8)}x^{(2)} - 1.77 \times 10^{(-4)}x + 3.16 \quad (3)$$

It was determined that the third order equation was the best fit of the voltage. Based off this model, we calculate the amount of time taken for the battery to deplete using (3). Expected Battery Depletion Time = Y(T) = 2.1V is subtracted from Current Time of the New Battey = Y(T) = 2.9 V. The estimate of the time it takes to reach 2.1V from a new battery is 15.9 months.

The time series analysis resulted in the following ARIMA (p=0, d= 2, q= 1) model (4), which can also be described as a MA (1) model on the second differences. As mentioned in the methods section, this model uses the second difference of the time series data to transform it into a stationary time series. It contains one moving average component. $\nabla^2 x_t$ refers to the second difference, w_t refers to an independent, identically distributed normal random variable with parameters N (0, 4.26×10^{-5}).

$$\nabla^2 x_t = 2.89 + w_t - 0.9712 w_{t-1} \quad (4)$$

Assessing the time series model, we find its AIC value to be –1136.08. Fig. 9 shows that the QQ plot of the fitted values against the residuals is Gaussian for our model, making it valid to use for forecasting.

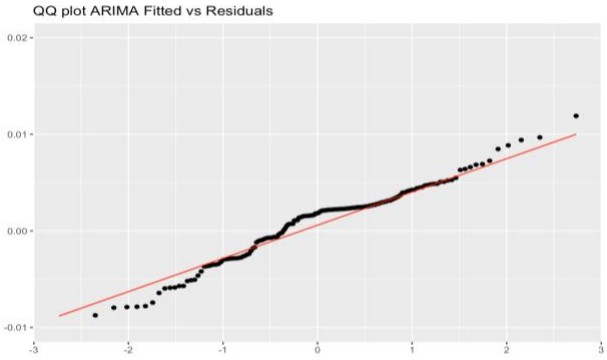

Fig. 9. Diagnostic plot of the "Voltage.auto" ARIMA model

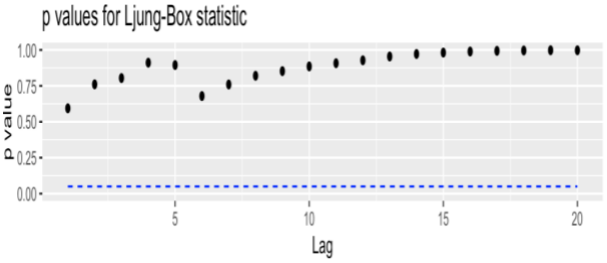

Fig. 10. Plot of Ljung-box Test

The plot of the Ljung-box test in Fig. 10 shows that the model is adequate for more than twenty lags, which also increased our confidence in the model.

Using our time series model, we forecasted the values for the next 10 days, and compared them to actual data.

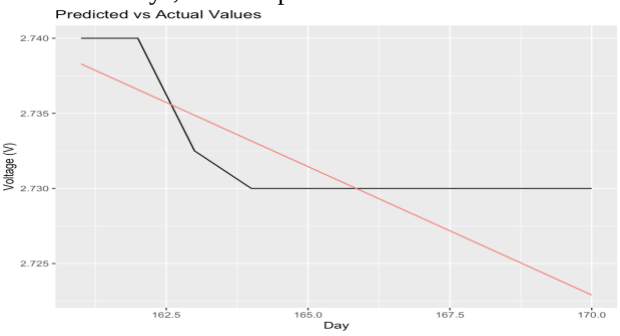

Fig. 11. Predicted vs Actual Voltage

The predicted voltage trend shown in red in Fig. 11 closely follows the actual values shown in black. The MSE of the set of predicted and actual values was extremely low, at 1.29×10^{-5}. This, along with the diagnostics performed indicate that the model we created can give reliable predictions of future battery data.

C. Deployment of IoT Sensors

Our team deployed near the Dell Pond in Charlottesville one pressure sensor (model DL-PR26) and one soil moisture sensor (model DL-SMTP) on March 21, 2023. The levels of soils moistures represented in Fig.12 are: 100, 200, 300, 400, 500 mm which respectively correspond to levels 0,1,2,3,4,5.

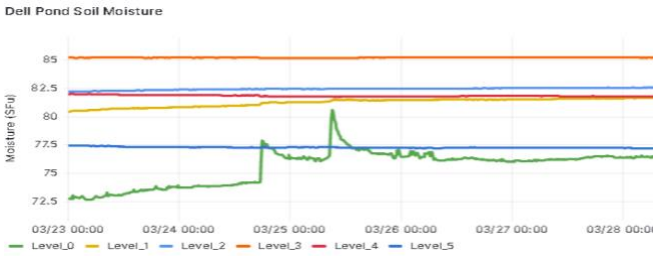

Fig. 12. Moisture Levels from Five Depths at the Dell Pond

As seen in Fig. 13, we have an average of 55mBar. The spike shows an increase in water pressure, evidence of a storm. It infers how pressure is related to how deep the water is. As the pressure increases, the water depth also increases; this can be used for notifications for flood alerts.

Fig. 13. Water Pressure Levels at the Dell Pond

IV. Discussion

A. Limitations of the Hydrologic Model

Our model is now only considering subbasins around the Dell watershed. There are some flood control structures along the pond that have not yet been added to the model. There are also some limitations to the HEC-HMS software. The simplified HEC-HMS model cannot model loop stream networks or backwater. The model can be improved by adding more parameters such as soil moisture. Since the sensor is deployed, we can add the soil moisture and precipitation data derived from sensor. However, the time required to run the model is also worth considering. If there is an excessive number of parameters presence in the model, the software will take a longer time to simulate.

B. Limitations of the IoT energy model

The analysis of the battery level was based on only one sensor, the weather station sensor, which is the only one for which we have a full battery discharge cycle. For future studies, more data used from the other sensors as well as more iterations of the same sensor going through multiple discharge cycles would more accurately depict the lifespan of a given sensor. Although preliminary results of individual models were acceptable, more investigation is needed on how to achieve an integrated approach to battery voltage modeling using principles from both methods demonstrated in this paper to capitalize on the different strengths that each can offer and provide a real-time display of remaining battery time. Additionally, to improve the battery prediction models, regression models should include other additional variables such as spreading factor, which can provide insights on their significance to battery life.

V. Conclusion

Charlottesville, like many cities and communities worldwide, is expected to experience a greater frequency and intensity of storms due to changing weather patterns resulting from climate change. This paper holistically incorporates components important to IoT functionality to create a basic understanding of one of many watersheds in the Charlottesville area. It provides a step towards accurately predicting and warning the local community about floods in real-time.

This work contributes a simple model of applying IoT in flood management. It delivers general guidance on how to create flood modeling using geographic data in ArcGIS and precipitation data added in HEC-HMS. Then, simulation hydrographs for 5-, 10- and 25-years storm can be generated.

Our preliminary investigation of battery life contributes to the ongoing discussion of IoT system management, as it can support managers in efficiently allocating resources for IoT battery replacement tasks, especially when dealing with many IoT devices and hard-to-reach deployment locations. Furthermore, the sensors deployed near the Dell Pond will help to create a small-scale testbed which can be scaled up for creeks, rivers, and ponds in Charlottesville going forward. With more IoT deployments, more battery discharge data will be collected, allowing for more complex battery life estimation models.

To further develop this flood management system, the next steps would involve creating code to automate the process of incorporating precipitation data into the HMS model and the predicted water levels into the database. The precipitation data would come from the sensors. Ideally, a complete dashboard would consist of an interface which displays the current water levels as well as simulated water levels and alerts when the actual water levels are reaching a threshold point. The interface would allow flood response teams to compare the actual water levels to the potential flooding water levels and enact their flood warning response sooner and more accurately. It would also include the ability to predict the battery level to enable proactive maintenance of the IoT system.

References

[1] "Local effects of climate change," Jun-2020. [Online]. Available: https://static1.squarespace.com/static/5a0c67f5f09ca475c85d7686/t/5efe15db11d5fa0d7fffd167/1593710045709/Local+Effects+of+Climate+Change.pdf. [Accessed: 01-Apr-2023].

[2] "Release: New data shows Millions of people, trillions in property at risk from flooding - but infrastructure investments now can significantly lower flood risk," World Resources Institute, 23-Apr-

2020. [Online]. Available: https://www.wri.org/news/release-new-data-shows-millions-people-trillions-property-risk-flooding-infrastructure#:~:text=The%20number%20of%20people%20exposed,and%20to%20triple%20by%202050. [Accessed: 01-Apr-2023].

[3] "GIS Data Download," The National Map. [Online]. Available: https://www.usgs.gov/the-national-map-data-delivery/gis-data-download. [Accessed: 03-Apr-2023].

[4] "CBD Land Use/Land Cover Data Project," Chesapeake Conservancy, 2022. [Online]. Available: https://www.chesapeakeconservancy.org/conservation-innovation-center/high-resolution-data/lulc-data-project-2022/. [Accessed: 03-Apr-2023]

[5] "Download ready-to-use project packages with over 170 attributes derived from the SSURGO dataset," ArcGIS, 14-Feb-2022. [Online]. Available: https://www.arcgis.com/home/item.html?id=cdc49bd63ea54dd2977f3f2853e07fff. [Accessed: 03-Apr-2023].

[6] "HEC-HMS Technical Reference Manual – CN Tables," HEC-HMS Documentation. [Online]. Available: https://www.hec.usace.army.mil/confluence/hmsdocs/hmstrm/cn-tables. [Accessed: 03-Apr-2023].

[7] DecentLab, "DL-ATM41 Datasheet". Available: https://cdn.decentlab.com/download/datasheets/Decentlab-DL-ATM41-datasheet.pdf. [Accessed: 03-Apr-2023].