

ISSN(O): 2320-9801 ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.625

Volume 13, Issue 1, January 2025

⊕ www.ijircce.com 🖂 ijircce@gmail.com 🖄 +91-9940572462 🕓 +91 63819 07438

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Real Time Flood Monitoring and Alert System

Anusha C, Vaishnavi C, Vidya GK, Yashwanth Reddy PB, Harish Kumar HC

Department of Computer Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka, India

ABSTRACT: Floods are catastrophic natural disasters requires the better and efficient predictive systems, monitoring, comprehensive real-time flood monitoring and alert system leveraging Arduino software, IoT sensors, cloud computing, and machine learning to enhance flood preparedness. The system collects all the environmental data from sensors, including DHT11, ultrasonic, raindrop, vibration, soil moisture, sensors, processed by an ESP32 microcontroller. Data is dynamically transmitted to Google Sheets using App Script and stored in Google Cloud storage like buckets and Big Query for advanced analysis. The flood status is determined based on predefined thresholds and analysis using Looker Studio. Alerts are sent to people who are in flood areas.

KEYWORDS: Flood monitoring, IoT, ESP32, Google Cloud, Buckets, Big Query, Machine learning, Looker Studio, Appsheet, flood app.

I. INTRODUCTION

One of the most damaging natural disasters are floods, these have a terrible effect on both people, infrastructure and animals. This unique real-time flood monitoring and alert system which integrates sensor data to cloud computing, machine learning, and Internet of Things sensors in order to effectively analyze, anticipate, and measure flood scenarios. In alongside monitoring data in an app and generating timely alerts for proactive a flood management, the system provides an interface to collect, analyze, and visualize flood data.

With this system in communities, localities, place can be better prepared and respond more Fastly to potential and fast flooding events. Families and government can receive early warnings to evacuate safely, reducing the risk to lives ,local authorities can allocate resources more efficiently, ensuring that help reaches those in need as quickly as possible.



Figure 1.1 System architecture.

II. RELATED WORK

 Jong-uk Lee. Jae-Eon Kim, Daeyoung Kim, Poh Kit Chong, Jungsik Kim, Philjae Jang, developed a RFMS, a real-time flood monitoring system using wireless sensor networks for high accuracy and reliability in data transmission.
 Diogo de Souza, Iago Freitas Cardoso, Marcos Rodrigo Momo proposed a mobile application integrated with flood alert systems, focusing on improving accessibility and responsiveness.

[3] P. William, Oluwadare Joshua Oyebode ,Gandikota Ramu,Sorabh Lakhanpal, Keerat Kumar Gupta, Hassan M. Al-Jawahry, implemented an IoT-based monitoring system employing an LSTM network for flood prediction based on rainfall and water levels.

www.ijircce.com



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[4] Yung-Chung Tsao, Yin Te Tsai, Yaw-Wen Kuo, Chaokung Hwang, explored IoT-based weather monitoring systems to address environmental challenges, focusing on efficient data handling and analysis. These works highlights the increasing role of IoT, monitoring and analytics, which our project builds upon by integrating IoT sensors, cloud computing, and advanced visualization.

III. DESIGN AND IMPLEMENTATION

3.1 System components.

Sensors which include DHT11 to sense temperature,

Humidity, ultrasonic sensor which sense the water level distance, vibration sensor detects the vibration levels, soil sensor measures moisture, raindrop sensor for intensity of rainfall, these sensors sense the environment and uses the ESP32 microcontroller which is WIFI module, collects and transmits the data dynamically to storage.

Where the data collected is sent to cloud through google sheets via appscript for logging and monitoring, where this data is shown in appsheet app known as flood app which displays the data dynamically, data is imported to google cloud storage, like buckets and Big query, stores the tables for large scale analysis, and visualises trends and displays the real time analytics.

Custom app which linked to google sheets, which administrator to monitor live data and receive the feedbacks, and sent alert messages.

Machine learning integration where model are developed to analyze data trends and predicts floods, google colab where model is trained and evaluate the accuracy with a confusion matrix.

3.2 Implementation

A. Data Collection and Transmission:

Iot sensor collects the data ,processed by esp32 microcontroller and connected wifi, and transmitted to google sheets dynamically through the Appscript which the extension in sheets and data parameters like water level, raindrop intensity, vibration, and soil moisture [1][2].where these thresholds are predetermined and determine the flood status in a new column.

B. Cloud Storage:

Data from google sheets are synchronized with query or imported to google cloud storage, where the project are created in google cloud, enabled the api services like bigquery api, googlesheets api, and enabled the credits for the project and data is imported to buckets for further process and notification.

C. Visualization and Alerts

The data which dynamically generated in sheets and sent to cloud storage, where the data is imported to looker studio where the analytical tool from google services, we can create charts . it detects give the analysed charts or can create the custom charts through selected parameters and metrics and these are analysed and created the report used for analysing the trends of future and present trends.

And the alerts are sent to users through sms or emails.

D. Machine learning

The data present in google cloud storage is used for training, where the vertexai provides the automl and custom ml model creation, we used custom where the data is trained and confusion matrix and graph is monitored[3][4].

E. Flood app

The data in the sheet is used for displaying the weather data to users , all andriod ,iphone and web browers can acces the data, which synchonize data regularly , this data is deployed in appsheet and designed the app.

www.ijircce.com



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. RESULTS

diga 1 Function data(a) (detacgarges // Open the settime spreaddreat buildedge const strike spreaddreat SMSge // Open the settime spreaddreat Densite // Aprileve the parameters from the GIT request Bubledge // Aprileve the parameters from the GIT request Bubledge // Aprileve the parameters from the GIT request Bubledge // Aprileve the parameters from the GIT request Bubledge // Open the settime spreaddread Bubledge // Open the settime spreaddread Bubledge // Open the settime spreaddread Services + Bubledge 10 Bubledge 10 Bubledge 10 Bubledge 10 Shoets 11 10 10 11 11 12 11 13 11 14 11 15 12 16 14 17 11 18 14 19 11 19 11 11 11 12 11 13 11 14 11 15 12 16 <th>doGet 👻 Execution log</th> <th></th>	doGet 👻 Execution log	
deskoggerge 2 Function dafation (boxetge 1 // Open the active samedatheat boxetge acmst whet = Spreadsheatge.getActiveSpreadsheat(), getActiveSheet(); SMSge 2 // Aprime the parameters from the GFT request boxetge 7 boxetge 2 const intrive the parameters from the GFT request boxetge 3 const intrive the parameters from the GFT request boxetge 3 const intrive the parameter.html () [* 0'); boxetge 3 const intrive aperameter.html () [* 0'); boxetge 3 const intrive aperameter.html () [* 0'); boxetge 3 const into a parametor () parameter.html () [* 0'); boxetge 3 const into aperameter.html () [* 0'); boxetge 1 const into a parameter.html () [* 0'); boxetge 1 const into a parameter.html () [* 0'); boxetge 1 const into a parameter.html () [* 0'); boxetge 1 const into aperameter.html () [* 0'); boxetge 1 const into aperameter.html () [* 0'); boxetge 1 const into aperameter.html () [* 0'); boxetge 1 distance < 3 bit into aperameter.html () [* 0');		
bulsatge // Open the active isprediated: SMSge torns inter < Spreadbacking, petchickedprediative@preadback(), getXcliveShee()):		
Jobsongp Outsister (// Services the parameters from the Off request Difference // Services the parameters from the Off request Difference + 0 000000000000000000000000000000000000	ibeet	
SM5 ge 6 // Services the parameters from the OTF request Lbrades 4 0 const increations (parameters transportune]) "0"); Services 4 0 const increations (parameter, dustance]) "0"); Services 4 0 const increations (parameter, dustance]) "0"); Services 4 0 const increations (parameter, dustance]) "0"); Services 1 0 const increations (parameter, dustance]) "0"); Services 1 0 const increations (parameter, dustance]) "0"); Services 1 1 const increations (parameter, dustance]) "0"); Services 1 1 const increations (parameter, dustance]) "0"); Services 1 1 const increations (parameter, dustance]) ["0"); Services 1 1 1 1 Services 1 1 1 1 1 Services 1 1 1 1 1 Services 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 <td< td=""><td><pre>dp.gecheciveopreadoneec().getheciveoneec().</pre></td><td></td></td<>	<pre>dp.gecheciveopreadoneec().getheciveoneec().</pre>	
Ubravies 7 nonixi treprature - paraFloat(e, parameter - invaristy 10°); Ubravies 4 9 const trapparture - paraFloat(e, parameter - invaristy 10°); Services 4 9 const trapparture - paraFloat(e, parameter - invaristy 10°); Services 4 9 const trapparture - paramefloat(e, parameter - invaristy 10°); Services 4 10 const trapparture - parameter - invaristy 10°); Services 11 const trapparture - parameter - invaristy 10°); Services 12 const trapparture - parameter - invaristy 10°); Services 13 const trapparture - parameter - invaristy 10°); Services 14 14 const trapparture - parameter - invaristy 10°); Services 15 const trapparture - parameter - invaristy 10°); Services 16 16 const trapparture - parameter - invaristy 10°); Services 16 16 16 16 16 16 16 16 16 17 16 10 10 17 18 16 10 10 16 16 16 16 16 16 17 16 16 10 16 18 16	a from the GET request	
Libraries + 0 const functions percention(: percenter function) 0 Services + 0 const statumer percention(: percenter function) 0 Services + 0 const statumer percention(: percenter statumer) 0 Services + 0 const statumer percention(: percenter statumer) 0 Services - 1 percention(: percenter statumer) 0 Services - // Determine flood statum 0 Services - // Determine flood statumer) 0 Service - 1 // Determine flood statumer) 0 Service - 1 // Determine flood statumer) 0 Service - 1 // Append s new row to fine sheet stat the reserved data - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - </td <td>loat(e.parameter.tamperature "0");</td> <td></td>	loat(e.parameter.tamperature "0");	
Services + 10 cover rate - paraEnt(e parameter (rain (19%)) cover vide - paraEnt(e parameter (rain (19%)) cover vide rates - parameter (rain (19%)) Sector cover soll - parameter (rain (19%)) Sector 10 (/ Determine flood status 11 (distance < 0 H4 vibration - rate flood 12 (floodStatus - 0 (// Default two flood 13 if (distance < 0 H4 vibration - rate flood status 14 (distance < 0 H4 vibration - rate flood status 15) 16 (distance < 0 H4 vibration - rate flood status 16) 17 (Append a new rew to the theet with the restived data 18 obst. copendice() 19) 10 temperature, // Fenerature ('C) 10 hatticty, // Actar Level Bistance (cn) 21 cistance, // Actar Level Bistance (cn) 22 cistance, // Actar Level Bistance (cn) 23 cistance, // Actar Level Bistance (cn)	it(e.parameter.humidity 107);	
Ministry T Inconstruction account of the account of t	arameter, rain (("0");	
<pre>hgCary 12 const soil = parseins(c.parameter.soil 'B'); figCary 13 // Determine flood status 15 ler floodStatus = 0 // Defailt: No flood 15 if (Sittance < 5 M4 vibration</pre>	<pre>r(e.parameter.vibration *8");</pre>	
<pre>Shorts 10 // Detentive fload status Shorts 15 let fload status = 0 // Default: No fload 16 if (distance < 5 M& vubration 1 A& soil < 2008 A& rain = 2000) { 17 [floadStatus = 0; // Fload cossible 18 // Append a new row to the short with the restived data 19 // Append a new row to the short with the restived data 19 choor.appendixe() 10 new cost(), // Timestamp (*) 10 temperature, // Temperature (*C) 10 temperature, // Temperature (*C) 11 estatus, // Aking Distance (on) 13 estatus, // Aking Distance (on) 14 rais. // Aking Distance value)</pre>	<pre>srameter.soil ('0');</pre>	
Shoots 15 let floodStatus = 0; // Defuilt No flood 16 if (distance > 3 E& virtuation == 1 & Soil = 2000 && raim = 2000) { 17 floodStatus = 1; // Flood possible 19 19 19 10 // Append a new row to the sheet sith the received data 21 eheot.appendRew(] 22 newsDate(), // limestamp 23 temperature, // Temperature ('C) 24 handSty, // Mediaty (Distance (cn) 25 raime, // Mediaty Distance (cn) 26 raime, // Mediaty Distance (cn) 27 raime, // Mediaty (Distance value)		
<pre>16 if (distance < 5 M& vibration 1 M& soil < 2009 4& rain < 2009) { 17 f(distance < 5 M& vibration 1 M& soil < 2009 4& rain < 2000) { 19 } 10 } 10 } 10 } 10 // Append a new rew to the sheet with the reserved data 10 otherspeendieve() 11 otherspeendieve() 12 temperature, // Temperature 'C) 14 temperature, // Temperature 'C) 15 temperature, // Temperature 'C) 16 temperature, // Astantion Internativ (raw senser value) 17 otherspeendieve() 18 of states, // Astantion Internativ (raw senser value) 19 of states, // Temperature (value) 10 of states, // Temperature (value) 10 of states, // Astantion Internativ (raw senser value) 10 of states, // Temperature (value) 10 of states (v</pre>	efault: Ne flood	
<pre>17 floedStatus = 1; // Floed possible 10 19 19 20 // Append a new new to the sheet with the restived data 21 sheet.appendixe([22 new Date(), // Tainstamp 23 temperature, // Temperaure ('C) 24 nanticity, // Matter Level Distance (cn) 25 rain, // Matter Level Distance (cn) 26 rain, // Matter Level Distance (cn) 26 rain, // Matter Level Distance (cn) 27 rain, // Matter Level Distance (cn) 28 rain, // Matter Level Distance (cn) 29 rain, // Matter Level Distance (cn) 20 rain, // Matter Level Distance (cn)</pre>	tion 1 && soil < 2099 && rain < 2009) {	
1) 7 3) // Appond a new row to the sheet with the received data 3) sheet.appendixe([20 new Det(), // limestamp 2) temperature, // Temperature 'C) 24 temperature, // Temperature 'C) 25 statane, // Asiation Internativ (raw senser value) 26 rain, // Asiation Internativ (raw senser value)	jd possible	
3 // Append a new rew to the sheet with the reserved data abort.opendiew() rew.Dost(), // Insettamp temperature, // Temestamp temperature, // Headingtone (*G) headingty, // Headingtone (*G) distance, // Headingtone (ce) real, // Headingtone (ce) real, // Headingtone (ce)		
<pre>21 short.appendixx() 22 new Det(), // Innettamp 23 temperature, // Fenoreture (*C) 24 hutticity, // Henoreture (*C) 24 hutticity, // Metar Luvel Bistance (en) 25 cistance, // Metar Luvel Bistance (en) 26 rain, // Metar Luvel Bistance (rain senser value) 26 rain, // Metar Luvel Bistance (rain senser value)</pre>	s sheet with the received data	
22 new Dote(). // lanestamp 23 temperature. // Temperature (*C) 24 huntdity. // Hundity (%) 25 distance. // Kotar Lavel Distance (on) 28 rish. // Handros Intensity (raw senser value) 29 valuestamp // Handros Intensity (raw senser value)		
20 temperature, // fenomerature (°C) 84 manifesty, // Menar Lawel Sistemes (cn) 25 cistance, // Mear Lawel Sistemes (cn) 25 rain, // Macar Lawel Sistemestry (raw Senser value)	linestamp	
<pre>24 initially (* maining (*) 25 initial (*) 26 initial (*) 28 initial (*) 28 initial (*) 29 initial (*) 29 initial (*) 20 initial (*) 20</pre>	Temperature (°C)	
25 rain, // Reindron Interstry (respective value)	Water (euc) Distance (cm)	
22 subration (1) Schedulin Datasted (1)(0)	Haindrop Intensity (raw sensar value)	
an Fabre Castly of Fabre Castle Castl	/ibration Detected (1/0)	
28 soil, // Soil Moisture (raw sensor value)	Soil Moisture (raw sensor value)	

Figure 4.1 Appscript code .

	File Edit View	Insert Format	Data Too	ols Extensi	Defaul T	- [10]		. ,		A	۵.	-		-	1 -	14. 1	
		10070		+		1				-		m.		-	÷	1.	10
AT	→ j _R 22/12/2	024 10:38:45															
	٨	D	c		D	5			10		0			2882			13
1	22/12/2024 10 38 45	TEMPERATURE	HUMIDITY	DISTANCE		RAINDROFT	VALUE	VIBRA	NON		SOIL M	OIST	HLO	JD STA	IUS		
2	22/12/2024 10:38.57	27.8	09		235.4		4095			0		4095			U		
3	22/12/2024 10:30:11	27.0	00		235.4		4005			0		4090			0		
4	22/12/2024 10:39:24	27.6	60		65.87		4085			0		90.7			0		
÷.	22/12/2024 10:39:38	27 8	69		235.06		4095			0		1342			n		
6	22/12/2024 10 39 52	27.6	89		235.77		4095			U		1472			0		
7	22/12/2024 10:40:04	27.8	69		255.11		4095			U		1493			U		
8	22/12/2024 10:40:17	27.6	00		235.70		1437			0		1575			0		
ч.	22/12/2024 10:40:31	27.6	60		235.37		1451			0		1567			0		
10)	22/12/2024 10:40:45	27.6	69		235 43		1634			1		1597			0		
11	22/12/2024 10:40:58	27.8	вя		3.503		1807			0		1657			D		
12	22/12/2024 10.41.11	27.8	UU		3.93		1778			1		1000			1		
13	22/12/2024 10:41:29	27.6	60		3.93		1808			0		1649			0		
14	22/12/2024 10:41:42	27.6	60		4.25		1762			0		16.30			0		
19	22/12/2024 10:41:56	20	60		4 25		095			n		1648			0		
16	22/12/2024 10 42 09	28	88		4.25		1186			0		1661			0		
17	22/12/2024 10.42.23	28	68		4.25		14//			U		1681			U		
15	22/12/2024 10:42:37	28	68		4.25		1037			0		1091			0		
14	22/12/2024 10:42:51	28	60		4.25		1665			0		1636			0		
20)	22/12/2024 10:43:04	20	60		4.25		1679			n		1600			n		
21	22/12/2024 10:43:17	28	BB		4.25		1893			0		1648			Ū.		
22	22/12/2024 10.43.32	28	68		3.6		1680			U		16//			U		
23	22/12/2024 10:43:44	28	08		3.93		1080			0		1631			0		
74	22/12/2024 10:43:58	28	68		.3.6		1606			0		1600			0		
the.	22/12/2024 10:44:10	20	60		36		1705			0		1500			0		

Figure 4.2 google sheet

=	FLOOD APP			Q, su	an PLOOD XPP				0 - 0
1	FLOOD APP								+ Add = 🖂
0	45548.44357583241 \wp	TEMPERATURE	HUNIDITY	DISTANCE	RAINDROP VALUE	VERATION	SOL MOISTURE	FLOOD STATUS	i
	12/22/2824 11 12:27 58	2625	20	2.96	2.941	1	2,067	n	3
	12/23/2024 11 17 17 44	7675	27	2.94	2.911	3	2,008	n	,
164	12/28/2024 11:17:05 AM	26.70	67	2.96	2.882	ä	1,939	0	,
	12/28/2024 11:16:56 AM	26.70	67	2.96	2.879	4	2.008	0	,
	12/23/2024 1516-47 AM	26.70	e)	2.96	2.928	1	2.025	ш	,
	12/23/2024 1516:39 AM	26.79	6/	2.96	28/5	1	1942	ц	2
	12/23/2824 1338:29.64	7675	67	7.Rc.	2 0.39	1	2,041	n	\$
	12/23/2024 11 W:20 AM	2675	0	2.86	2.005	3	2,037	n	,
	12/28/2024 11:10:11.44	26.70	47	4.25	2.807	31	1,9.15	0	,
	12/28/2024 11:16:02 AM	26.70	67	4.25	2,777		1,919	0	,
	12/23/2814 1515:33 AM	26.79	6)	4.25	2304	1	3.011	u	,
	12/23/2824 1512:16 AM	26.20	67	2.31	1.443	1	2,080	ц	3
	12/25/2024 11:12:07 AM	26.20	17	6.39	1.119	4	Jane	п	\$
	12/23/2024 13 HSB AM	36.20	12	2.3%	1.852	3	2,014	п	,
	12/28/2024 11:11:49.44	26.20	67	2.8	1.126		2.021	0	

Figure 4.3 Flood app data.



Columnes Indow/					
COTOBILS! THOEXC	['TEMPERATU	RE'. 'HUM	UDITY'. 'D	STANCE', 'RAINDROP VALUE', 'VIB	RATION'.
SOTL MO	TSTURE', 'F	LOOD STAT	us'1.		
dtype='ob	iect')		100 A.C.		
Missing values:					
TEMPERATURE	0				
HUMIDITY	0				
DISTANCE	0				
RAINDROP VALUE	0				
VIBRATION	0				
SOIL MOISTURE	0				
FLOOD STATUS	0				
dtype: int64					
		and a second sec			
<ipython-input- df = df.fillna</ipython-input- 	1-8ed1d760e a(method-'f	8d3>:29: fill')	FutureWarn	ng: DataFrame.fillna with 'meth	od' is deprec
<pre><ipython-input-: 1.0<="" accuracy:="" df="df.fillna" pre=""></ipython-input-:></pre>	1-8ed1d760e a(method-'f	8d3>:29: fill')	FutureWarn:	ng: DataFrame.fillna with 'meth	od' is deprec
<pre><ipython-input-: 1.0="" 1<="" accuracy:="" classification="" df="df.fillna" pre=""></ipython-input-:></pre>	1-8ed1d760e a(method-'f Report:	8d3>:29: fill')	FutureWarn:	ng: DataFrame.fillna with 'meth	od' is deprec
<ipython-input- df = df.filln Accuracy: 1.0 Classification I</ipython-input- 	1-8ed1d760e a(method-'f Report: precision	8d3>:29: fill') recall	FutureWarn: f1-score	ng: DataFrame.fillna with 'meth support	od' is depred
<pre>cipython input- df = df.filln Accuracy: 1.0 Classification I I</pre>	1-8ed1d760e a(method-'f Report: precision 1.00	8d3>:29: fill') recall 1.00	f1-score	g: DataFrame.fillna with 'meth support 268	od' is depred
<pre><ipython-input- df - df.fillm. Accuracy: 1.0 Classification 1 0 1 0 1</ipython-input- </pre>	1-8ed1d760e a(method-'f Report: precision 1.00 1.00	8d3>:29: fill') recall 1.00 1.00	f1-score 1.00 1.00	g: DataFrame.fillna with 'meth support 268 37	od' is deprec
<pre>cipython-input- df - df.filln Accuracy: 1.0 Classification 1 0 1 accuracy</pre>	1-8ed1d760e a(method-'f Report: precision 1.00 1.00	8d3>:29: fill') recall 1.00 1.00	f1-score 1.00 1.00 1.00	ng: DataFrame.fillna with 'meth support 268 37 305	od' is deprec
<pre>cipython-input- df - df.filln Accuracy: 1.0 Classification 1 0 1 accuracy macro avg</pre>	1-8ed1d760e a(method-'f Report: precision 1.00 1.00	8d3>:29: fill') recall 1.00 1.00	f1-score 1.00 1.00 1.00 1.00	g: DataFrame.fillna with 'meth support 268 37 305 305	od' is deprec
<pre>cipython-input- df = df.fillm. Accuracy: 1.0 Classification 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0</pre>	1-8ed1d760e a(method-'f Report: precision 1.00 1.00 1.00 1.00	8d3>:29: fill') recall 1.00 1.00 1.00	f1-score 1.00 1.00 1.00 1.00 1.00	ng: DataFrame.fillna with 'meth support 268 37 305 305 305	od' is depre

Figure 4.4 confusion matrix in vertex ai colab.

V. CONCLUSION AND FUTURE SCOPE

In summary, the promise of combining IoT, cloud computing, and machine learning for real-time environmental monitoring is demonstrated by the suggested flood monitoring and alert system. By using the sensors such as DHT11, vibration, raindrop sensor, soil moisture, ultrasonic sensor the system effectively collects and sends environmental data to cloud storage. While Looker Studio analytical tool which provides user-friendly visualization to in decision-making, Google Sheets and Google Cloud Big Query provides data storage and analysis. The mobile application and web application which improves readiness during flood disasters ,monitor data in real-time and receive notifications. The technology generates dependable forecasts with an accuracy of over 93% thanks to a machine learning model that was built on actual data in vertex ai . the system is scalable, effective, and has a significant impact on reducing the effects of floods.

In order to give more coverage and prediction accuracy, future versions of the system will try to extend its deployment throughout many flood-prone areas. The system may produce more accurate forecasts by expanding the number of sensors and including more environmental factors like wind speed, sun radiation, and river flow data. Predictive capacities will be further improved by developments in machine learning models, such as the application of neural networks and real-time model training. Reliability in remote locations will also be increased by enabling offline functioning and integrating energy-efficient solutions like solar power. The goal is to establish a worldwide network of linked flood monitoring systems that are capable of real-time flood prediction and response anywhere.

REFERENCES

1. RFMS: Real-time Flood Monitoring System with wireless sensor networks. Authors - Jong-uk Lee. Jae-Eon Kim, Daeyoung Kim, Poh Kit Chong, Jungsik Kim, Philjae Jang.

2. Tool-Based Mobile Application Applied to the Monitoring System and Flood Alert. Authors - Diogo de Souza, Iago Freitas Cardoso, Marcos Rodrigo Momo.

3. Framework for IOT Based Real-Time Monitoring System of Rainfall Water Level for Flood Prediction Using LSTM Network. Authors- P. William, Oluwadare Joshua Oyebode Gandikota Ramu,Sorabh Lakhanpal, Keerat Kumar Gupta, Hassan M. Al-Jawahry.

4. An Implementation of IoT-Based Weather Monitoring System. Authors - Yung-Chung Tsao, Yin Te Tsai, Yaw-Wen Kuo, Chaokung Hwang.



INTERNATIONAL STANDARD SERIAL NUMBER INDIA







INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

🚺 9940 572 462 应 6381 907 438 🖂 ijircce@gmail.com



www.ijircce.com