LOSS OF LIFE MODELLING FOR A GIS FRAMEWORK TO ANALYSE THE IMPACTS OF DAM BREAKS

by

SAMUEL CHUKWUDI OVU

B.Sc., Ebonyi State University, 2017

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF APPLIED SCIENCE IN

ENGINEERING

UNIVERSITY OF NORTHERN BRITISH COLUMBIA

November 2024

© Samuel Ovu, 2024

ABSTRACT

In recent years, advancements in dam management and construction have greatly contributed to improved dam reliability. However, failures still occur due to natural disasters, human error, and structural deterioration, often leading to severe flash floods. This research presents a GISbased framework aimed at assessing the impacts of dam failures, with a primary focus on loss of life (LOL) estimation. To achieve this, an improved LOL estimation model was proposed using a dataset of historical dam failures, divided by flood severity and evaluated through multivariate regression analysis. Key influencing factors-such as flood depth, population vulnerability, warning time, and evacuation conditions-were systematically weighted using entropy-based methods, allowing for a more accurate representation of risk levels in different failure scenarios. The framework integrates GIS spatial analysis with hydraulic modelling to evaluate potential damages resulting from dam failure. Results from model validation against historical cases demonstrated that the proposed LOL model achieved a strong fit to the observed data with a coefficient of determination (R²) value of 0.99 for both medium-high severity cases and for low-severity cases, outperforming conventional models, such as Graham's model. By integrating LOL prediction with spatial analysis, this framework provides assessment that can be incorporated into hazard mitigation plans, emergency response plans, and can also be used to inform insurance policies and risk mitigation strategies.

TABLE OF CONTENTS

Abs	tract		ii
Tab	le Of C	ontents	iii
List	Of Fig	ures	v
List	Of Tab	les	vi
Ack	nowled	gements	vii
1		Introduction	1
1.1		Research Needs	2
1.2		Objectives	3
1.3		Organization Of Thesis	3
2		Literature Review	5
2.1		Lol Estimation Model Review	8
3		An Improved Estimation Model For Dam Failure Induced Loss Of Life	20
3.1		Introduction	21
3.2		Failure Database	22
	3.2.1	Collection And Processing	22
3.3		Methodology	23
	3.3.1	Classification Of Influencing Factors Affecting Lol	23
	3.3.1	Normalization And Grading Of Influencing Factors.	28
	3.3.2	Entropy Calculation	28
	3.3.3	Weight Calculation, Filtration And Comprehensive Score Calculation	29
	3.3.4	Nonlinear Regression Analysis	29
3.4		Results	31
	3.4.1	Regression Analysis	33

3.5		Conclusion	36
4		Gis Framework to Assess Impacts Of Dam Failure	38
4.1		Introduction	39
4.2		Case Study	41
4.3		Methodology	41
	4.3.1	Data Collection and Preparation	41
	4.3.2	Hec-Ras (Hydrodynamic Modelling)	43
	4.3.3	Lol Estimation Calculation	44
	4.3.4	Tools	44
4.4		Results and Discussion	45
	4.4.1	Development of Integrated Plugin for Qgis	46
	4.4.2	Conclusion	47
5		Conclusion	48
Ref	erences		49
Арј	oendix		54

LIST OF FIGURES

Figure 1: Influencing factors for estimating life loss due to dam break24
Figure 2: Weight of influencing factors for the low severity case
Figure 3: Weight of influencing factors for the medium and high severity case
Figure 4: Comparison of actual LOL, predicted LOL and Graham's model
Figure 5: Overview of the methodology45
Figure 6: HEC-RAS simulation flowchart46
Figure 7: Inundation map with depth of Buffalo Creek flood47
Figure 8: Segment before (a) and after (b) flooding along Buffalo Creek channel downstream.
Figure 9: GIS framework plugin interface for (a) importing layers and (b) loss of life estimation
Figure 10: GIS framework plugin interface sample result50

LIST OF TABLES

Table 1 - Bibliometric Analysis	2
Table 2 - Review of variables for estimating loss of life by different authors	.16
Table 3. Dam failure cases	24
Table 4. Grading standard of influencing factors	.27
Table 5. Regression analysis metrics	34
Table 6. Comparison of the proposed equation and the actual LOL and FL	.35
Table 7. Information on the areas downstream of Buffalo Creek	41

Acknowledgements

I am profoundly grateful to everyone who supported me throughout this journey, making this thesis possible. First, I thank God Almighty for His guidance, grace, and strength throughout this process. Without His blessings, this achievement would not have been possible. I am deeply grateful for the wisdom and perseverance He provided during challenging moments.

I would like to extend my deepest gratitude to my supervisor, Professor Mauricio Dziedzic, for his invaluable guidance, constructive feedback, and unwavering support. His expertise and encouragement have been instrumental in shaping this work. I would also like to extend my sincere appreciation to my supervisory committee members, Prof. Jueyi Sui and Dr. Faran Ali.

I am also thankful to the faculty and staff at the University of Northern British Columbia, for providing a stimulating academic environment and access to the resources necessary for my research. I sincerely appreciate the MITACS interns Ibrahim Salaria and Leticia Spohr who assisted in gathering, producing and organizing part of the data for this study. Their contributions were crucial in building a comprehensive foundation for my research. Special thanks to my colleagues and friends, whose insightful discussions made this experience both enriching and enjoyable.

I am incredibly grateful to my family for their patience, love, and encouragement. Their belief in me has been a constant source of motivation, even during the most challenging times.

Lastly, thank you to everyone who has contributed to the successful completion of this thesis.

1 INTRODUCTION

Flooding, which is one of the most common and destructive natural hazard, is projected to worsen with climate change, higher sea levels and more intense cyclonic systems (Canadian Climate Institute, 2024; Sanders, 2007). Some of the most destructive flash floods stem from dam breaches. Dams play a crucial role in water resources management and offer significant benefits, including irrigation, flood control, hydropower generation, and water supply (ICOLD, 2011). However, their construction and operation also bring serious challenges. Dam failures can trigger catastrophic floods, posing risks to human lives, infrastructure, and the environment. According to Albu et al. (2020), approximately 1–2 dam breaches with fatalities are recorded worldwide each year, with causes often related to natural disasters, human errors, or degradation from long-term neglect.

With the increase in safety awareness and the improvement of the concept in dam risk management, more attention is being given to dam safety and life-threatening risks. This highlights the urgent need for effective dam breach risk evaluation and the development of hazard mitigation plans (Li et al., 2019).

Strategies to cope with flooding such as emergency preparedness, levee projects, floodplain building regulations and insurance all rely on flood predictions. Hence, the effectiveness of these measures is linked to the quality of flood predictions which depend on reliable modelling. Hydraulic modelling, a key tool since the 1940s, has evolved significantly since the early 2000s, especially with the use of GIS for detailed spatial analysis. GIS and hydraulic modelling enable precise simulation of dam breach scenarios, including dam collapse patterns, flood extents, water levels, flow rates, and water velocities (Eleutério, 2013). Studies demonstrate that integrating GIS and hydraulic models facilitates the development of inundation maps and supports flood risk assessments by mapping potentially affected populations and infrastructure (Abdalla, 2009; Ghent, 2013).

Loss of Life (LOL) estimation resulting from dam failure is critical for disaster management, legal accountability, and insurance purposes. By quantifying potential risks, engineers can identify vulnerabilities, enhance dam design, and aid authorities in developing effective emergency response strategies, evacuation zones, and resource allocation plans. Notable models for LOL estimation, such as those by Graham (1999) and more recent developments by Huang et al. (2017) and Mahmoud et al. (2020), have refined predictions by factoring in flood

severity, population vulnerability, and preparedness. These advancements, however, still face limitations in certain regions, outlining the need for continued model refinement.

This project proposes a framework to aid in assessing the consequences resulting from dam failure. By integrating LOL modelling with GIS and hydraulic modelling, the framework aims to enhance the predictive accuracy of dam safety assessments. This approach highlights the importance of modern tools and methodologies in addressing contemporary challenges in dam safety and disaster preparedness, aiming to provide a foundation for more informed decision-making and comprehensive risk management.

1.1 RESEARCH NEEDS

Since the first half of the 19th century, research on the effects of dam failure has been ongoing (Rogers, 1928), but assessment of all impacts has received very little attention. This research is relevant because it would help in understanding the effects of a dam break. Table *1* provides a bibliometric analysis of research studies relating to dam failures across various databases and search parameters. It highlights the scope of research using different search terms, illustrating the breadth and specificity of available literature. While dam failure research is well-represented in the literature, studies focusing specifically on the impact and use of GIS are comparatively sparse. This gap presents an opportunity for further research in integrating GIS for impact assessment.

Search	Parameters	Web of science	Science direct
dam AND break	All fields	4,452	39,646
dam AND break	All fields	972	22,631
AND impact			
dam AND break	All fields	33	1,736
AND impact			
AND GIS			
dam AND break	All fields	80	2,052
AND GIS			
dam AND break	Categories	1,951	20,196
		Water Resources or Engineering	Environmental
		Civil or Environmental	Science or Earth and
		Sciences or Engineering	Planetary Sciences or
		Environmental or Environmental	Engineering
		Studies or Geology.	

Table 1 - Bibliometric Analysis

dam AND break	Categories	508	13,332
AND impact	_	Water Resources or Engineering	Environmental
		Civil or Environmental	Science or Earth and
		Sciences or Engineering	Planetary Sciences or
		Environmental or Environmental	Engineering or
		Studies or Geology.	computer Science
dam AND break	Categories	19	1382
AND impact		Water Resources or Engineering	Environmental
AND GIS		Civil or Environmental	Science or Earth and
		Sciences or Engineering	Planetary Sciences or
		Environmental or Environmental	Engineering or
		Studies or Geology.	computer Science
dam AND break	Categories	48	1644
AND GIS		Water Resources or Engineering	Environmental
		Civil or Environmental	Science or Earth and
		Sciences or Engineering	Planetary Sciences or
		Environmental or Environmental	Engineering or
		Studies or Geology.	computer Science

1.2 OBJECTIVES

The purpose of this project is to conceptualize an open-source GIS framework for identifying and evaluating the potential economic, social, and environmental consequences of dam failure. As a starting point in the development of this framework, an improved loss of life model is developed, implemented, and tested.

1.3 ORGANIZATION OF THESIS

The thesis structure is as follows:

- 1. **Introduction**: This chapter outlines the motivation behind developing a GIS framework for assessing dam failure impacts, particularly loss of life. It introduces the research needs, objectives, and scope of the study.
- 2. Literature Review: This chapter explores existing models and methodologies for estimating loss of life due to dam failures, and also the current methods used to assess the economic impacts of dam breaks. It discusses empirical and agent-based models, their variables, and limitations. The review identifies a need for models that are adaptable to various regional contexts and addresses limitations in data availability.

- 3. An Improved Estimation Model for Dam Failure-Induced Loss of Life: This chapter presents the proposed model in detail. It begins with the failure database, outlining data collection and processing methods. The methodology here includes classifying influencing factors, normalizing data, calculating entropy and weights, and conducting a nonlinear regression analysis to estimate loss of life. Although there is no dedicated "Methodology" chapter in the thesis, the methodology for this model is contained within this chapter, detailing steps such as the classification and scoring of factors influencing life loss. This chapter is an enhanced version of a paper presented at the 2024 Canadian Society of Civil Engineers (CSCE) Conference and will be submitted to a peer reviewed journal.
- 4. GIS Framework to Assess Impacts of Dam Failure: This chapter describes the GISbased framework, focusing on data collection, preparation, and hydrodynamic modelling with HEC-RAS. The methodology section here is specific to implementing the GIS framework and calculating potential life loss using tools integrated with QGIS. Results include the development of a plugin that supports loss of life estimation. This chapter is a modification of a paper presented at the Canadian Dam Association (CDA) 2024 Conference, and now includes the description of a successfully implemented QGIS plugin, which was not available at the time of the conference.
- 5. **Conclusion**: The final chapter presents a general conclusion in addition to those stated in chapters 3 and 4.

Note that Chapters 3 and 4 each incorporate their own methodology sections, detailing specific approaches for different aspects of the study rather than having a unified methodology chapter. This structure allows each chapter to directly address its methodological requirements and applications in context.

2 LITERATURE REVIEW

The potential loss of life (LOL) caused by dam breaches has attracted the attention of many researchers and several methods have been developed that relate to mortality in the inundation area due to dam failure. Generally, empirical models, developed by regression analysis of historical dam breach events, were used to establish the functional relationship between the loss of life and certain key parameters. Brown & Graham (1988) proposed a method for predicting the LOL based on the analysis of the population at risk (P_R) and the warning time (TW). DeKay & McClelland (1993) proposed a model for estimating the nonlinear relationship between life loss and population at risk. In a publication by the US Department of Interior, Graham (1999) added the understanding of the dam break as an influencing factor to propose a procedure for estimating loss of life. Reiter (2001) proposed the RESCDAM method introducing the vulnerability of the population, warning efficiency and rescue condition. The U.S. Department of the interior Bureau of Reclamation (2015) also proposed a new method to replace Graham's method of estimating LOL. Although these methods can be used with ease, they require extensive data, and, therefore, have limited application (Jonkman et al., 2008). Also, they are not particularly precise because most of the methods do not consider all the variables/parameters relevant in the estimation of loss of life. However, they are useful as a screening tool to rank risks (D. Lumbroso et al., 2021). Additionally, the differences in economic and social conditions of different countries, as well as the discrepancy in the time span of statistical data, lead to a decline in the accuracy of these models when global usage is attempted.

Due to the limited applicability of empirical models, physical or agent-based models which focus on the analysis of the LOL mechanism gradually became research hotspots. Assaf and Hartford (2002) developed a virtual reality approach (BC Hydro's Life Safety Model (LSM)) to deal with the problems of failure consequence analysis and emergency planning. This LSM allows the behaviour of the population at risk to be represented, as it simulates the interaction of people with the modelled flood. This allows various scenarios to be investigated that could help reduce the risk to people (e.g., improvements in warning time and development of new evacuation routes). The LSM has been used to assess the potential LOL and propose effective emergency management measures for a number of historical dam failures (Lumbroso et al., 2011; 2021) - Malpasset dam, Canvey Island and Brumadinho tailings dam).

The U.S. Army Corps of Engineers (USACE) and the Australian National Committee on Large Dams (ANCOLD) supported the establishment of a GIS model for estimating dam failure and other life-loss calculation models by Aboelata et al., (2003; 2008). Aboelata (2008) LIFESim model is designed to simulate the entire warning and evacuation process for estimating potential life loss. Lee (2003) estimated life and economic loss by analysing dam-break related parameters, such as warning time and people's risk awareness by performing uncertainty analysis using Monte Carlo simulation.

Agent-based models can simulate different possible scenarios caused by floods and make effective safety decisions. However, these models also have high data requirements which resulted in comprehensive evaluation models recently receiving more attention. The comprehensive evaluation models combine the knowledge of both empirical models and physical models. Jonkman et al. (2008, 2018) developed a new function to estimate the LOL caused by flood disasters in low-lying areas. Their proposed approach takes into account flood characteristics, such as water depth, rise rate and flow velocity; estimation of the number of people exposed (including the effects of warning, evacuation and shelter); and assessment of the mortality amongst those exposed to the flood.

Ehsan (2009) also developed an improved method for LOL estimation by introducing new criteria for defining flood severity using a geometric aggregate (GA). Considering more factors affecting the LOL and the relationships among them, Peng & Zhang (2012) constructed the human risk analysis model (HURAM) based on Bayesian network theory in order to take into account more important parameters and their inter-relationships in a systematic structure, include the uncertainties of these parameters, incorporate information from previous studies and historical data, and to also update the predictions using Bayes' theorem based on available information in specific cases.

Based on data obtained from 14 dam failure cases in China, (Huang et al., 2017) proposed a new method for estimating the LOL using a three-dimensional stratified sampling method. Their method led to better results, when compared to observed values, than those obtained through Graham's method and Dekay and McClelland's method.

Li et al. (2019) analysed the weights of the primary factors that affect the consequences of dam breaks, using set pair analysis and the variable fuzzy set theory. Ge et al. (2019, 2020) constructed a rapid evaluation model based on catastrophe theory and used interval theory to

determine the possible upper and lower limits of LOL caused by dam breaches, rather than a single value. These comprehensive evaluation models focus on the innovation and application of mathematical methods, with a significantly improved level of accuracy when compared with other models. Ge et al. (2022) also modified pre-existing models and aimed to further account for evacuation potential, such as the time required for evacuation and the effective evacuation positions, while estimating LOL caused by a dam breach.

The establishment of an accurate flood-induced loss estimation method involves several issues due to the complexity of components, especially the nature of damage caused by flooding (Dutta et al., 2003). Several studies present detailed modelling of direct losses caused by flooding (Van Der Veen, 2004). However, due to incomplete, inconsistent, or unreported information, the scope/extent of losses is still not fully understood (Meyer et al., 2013; Van Der Veen, 2004). Most of the studies focused on direct tangible losses which can be estimated using replacement costs of damaged assets that can be monetized (Merz et al., 2004; Natho & Thieken, 2018; Yang et al., 2018). In assessing the economic consequence from a dam break, current literature primarily suggests two categories: those based on mathematical methods, and the use of GIS & remote sensing technology, both discussed in the following paragraphs.

The following studies employed evaluation models based on mathematical methods. Tang et al. (1992) investigated the data of several units in residential areas, industrial areas, agricultural areas and commercial areas of Bangkok, Thailand, using multiple regression analysis to estimate the flood-damage functions in terms of depth and flood duration. (Oliveri et al. (2000) proposed an empirical frequency-damage relationship for flood mitigation measures in strongly urbanised drainage areas and also used the structural replacement cost to estimate the average value of a property. Middelmann-Fernandes (2010) studied a combination of techniques to assess the economic cost caused by floods. Notaro et al. (2014) evaluated the uncertainty of water depth-damage function to calculate the flood damage in Cappalermo, Italy. The United Nations International Strategy for Disaster Reduction, UNISDR (2016), now known as UNDRR (United Nations Office for Disaster Risk Reduction) proposed a methodology to estimate the direct economic losses from natural hazards. The main point of the UNDRR's method is to provide a simple approach that allows for the estimation of direct economic losses for a wide range of disasters, based on documented physical damage (i.e., number of affected buildings, amount of destroyed agricultural area, number of livestock lost, etc.). (Natho et al. (2018) adapted and calibrated the UNDRR's model for Germany (as model M-DELENAH),

and it was concluded that: the UNDRR method underestimates the losses in general and that loss documentation needs to be improved to fill in the data gaps.

The following studies employed GIS and remote sensing technology for delineation of flood inundated areas for loss estimation. De Jonge et al. (1996) developed a flood hazard assessment model, simulated the flood depth and damage assessment using GIS technology, and established a flood disaster loss assessment model. The model focused on the socio-economic impacts of flooding, not on the ecological impacts. Haq et al. (2012) combined GIS with socio-economic data, developed a procedure for mapping inundated areas to determine the flooded range, estimate the affected land use/land cover types and the number of people affected. Mohammadi et al. (2014) proposed a model to estimate the economic amount of flood damage using flood depth as the assessment index. The assessment of existing literature points to a number of challenges related to data availability, access and quality.

2.1 LOL Estimation Model Review

The Brown & Graham (1988) procedure for estimating loss of life from a dam failure uses equations that were derived from the analysis of 24 dam failures and major flash floods. The warning time which was the major factor in calculating the loss of life is the time between the initiation of an evacuation warning and the arrival of the floodwater to the population at risk. This is presented as:

When warning time is less than 15 minutes:

$$Loss of Life = .5(PAR)$$
1

When warning time is between 15 and 90 minutes:

$$Loss of Life = PAR^{.6}$$
 2

When warning time is more than 90 minutes:

$$Loss of Life = .0002(PAR)$$
 3

PAR = *population* at *risk*

Dekay and McClelland (1993) further explored the work of Brown & Graham (1988), by including a few events that were not used by the latter. They proposed that loss of life is greater in situations where the flood waters are deep and swift. DeKay and McClelland have a separate equation for high and low force conditions.

Under high force conditions (where 20% or more of flooded residences are either destroyed or heavily damaged) their model is given by equation 4.

$$Deaths = \frac{PAR}{1 + 13.277(PAR^{0.440})e^{[2.982(WT) - 3.790]}}$$
4

Under low force conditions, i.e., where less than 20% of flooded residences are either destroyed or heavily damaged their model is expressed by equation 5.

$$Deaths = \frac{PAR}{1 + 13.277(PAR^{0.440})e^{[0.759(WT)]}}$$
5

When dam failure warnings do not precede the arrival of dam failure flooding in an area, warning time, WT is zero.

DeKay and McClelland (1993) cautioned against using their equations for dams that fail without warning above areas with very large populations at risk.

Graham (1999) proposed a new model seeking to correct the weaknesses in the Brown and Graham (1988), and the DeKay and McClelland (1993) equations. Graham provided fatality rates based on the severity of the flood, warning time and how people respond to the warning. He considered different categories of flood severity and proposed a method that can be used to separate low severity flooding from medium severity flooding by using the DV (level of destructiveness) parameter. (equation 6).

$$DV = \frac{Q_{df} - Q_{2.33}}{W_{df}}$$
 6

Where,

 Q_{df} is the discharge at a particular site caused by dam failure (m³/s).

 $Q_{2.33}$ (m³/s) is the mean annual discharge at the same site, specifically a discharge level that has a recurrence interval of approximately 2.33 years. This interval is chosen because it closely aligns with the median discharge for many river systems, which means it generally represents the flow level that can be considered "normal" or within the channel's safe capacity under typical conditions. This discharge can be easily estimated, and it is an indicator of the safe channel capacity. As discharges increase above this value, there is a greater chance that it will cause overbank flooding.

 $W_{df}(m)$ is the maximum width of flooding caused by dam failure at the same site.

DV is the general level of destructiveness that would be caused by the flooding and not necessarily representative of the depth and velocity and should provide a good indication of the severity of the flooding. As the peak discharge from dam failure increases, the value of DV increases. As the width of the flooding area narrows, the value of DV again increases.

Low flood severity is assumed when structures are exposed to depths less than 10 ft (3.048 m) and DV is less than 50 ft²/s (4.6 m²/s). Medium flood severity is assumed when depth is greater than 3.048 m and DV is more than 4.6 m²/s.

No guidance was provided for high flood severity level.

Reiter (2001) proposed the RESCDAM LOL Estimation Method, building on Graham's (Graham 1999) method but including more factors such as vulnerability factor, living environment factor and rescue factor, as given by equation 7.

$$LOL = PAR \times FAT_{BASE} \times IMPACT \times CORRFAT$$
7

Where

LOL = Loss of life caused by dam break flood.

 P_R = Population at risk.

 FAT_{BASE} = Base fatality rate of P_R, mean values from Graham (1999);

IMPACT = Additional impact factor to account for flood severity impact (SEV), living environment impact (LOC) and vulnerability impact (VUL) derived in the RESCDAM LOL method using public population register information on P_R.

11

CORRFAT = Correction factor to take the warning efficiency and possible emergency/rescue action into account in each sub-area Graham (1999).

Jonkman et al. (2008) proposed three general steps to estimate loss of life:

- Analysis of flood characteristics, such as water depth, flow velocity.
- Estimation of the number of people exposed (including the effects of warning, evacuation and shelter).
- Assessment of the mortality amongst those exposed to the flood.

The number of people exposed to the floodwaters (N_{EXP}) can be estimated by equation 8.

$$N_{EXP} = (1 - F_E) (1 - F_S) N_{P_R} - N_{RES}$$
8

 N_{P_R} = The number of people at risk before the event.

 F_E = The fraction of the population that is evacuated out of the area before the flood.

 F_{S} = The fraction of the (remaining) population that has the possibility to find shelter.

 N_{RES} = The number of people rescued.

Mortality is the number of fatalities divided by the number of exposed people. Mortality functions can be used to relate to flood characteristics (e.g. water depth) and other important factors such as the collapse of buildings. The number of fatalities (N) can be estimated by equation 9.

$$N = F_D N_{EXP}$$
 9

F_D -flood mortality

In the determination of mortality amongst those exposed to the flood, Jonkman proposed an approach in which hazard zones are distinguished. Three hazard zones were identified for a breach of a flood defense protecting a low-lying area.

i. Breach zone

Total destruction of masonry, concrete and brick houses occurs if the product of water depth and flow velocity simultaneously exceeds the criteria given in equation 10.

$$F_D = 1$$
 if $hv \ge 7 m^2/s$ and $v \ge 2 m/s$ 10

hv = depth-velocity product

v = flow velocity

ii. Zone with rapidly rising waters

This zone of concern as people on higher floors or buildings will be endangered due to higher water depths from the rapid rise of water. All the points categorized as being in the rapidly rising zone had rise rates of 0.5 m/hr and the derived function is applicable for situations with a rise rate above this threshold value.

Mortality in the zone with rapidly rising water:

$$F_D(h) = \Phi_N(\frac{\ln(h) - \mu_N}{\mu_N})$$
11

if $(hv \ 2.1 \ m \text{ and } w \ 0.5 hr)$ and $(hv < 7 \ m^2/s \ or v < 2 \ m/s)$

where

 $\Phi_{N} \, \text{is the cumulative normal distribution}$

w =rate of rise of water (m/hr)

 $\mu_N = 1.46$ is the average of the normal distribution

 $\sigma_N = 0.28$ is the standard deviation of the normal distribution.

iii. Remaining zone

Account for fatalities outside the breach and rapidly rising zones. Mortality in the remaining zone:

$$F_D(h) = \Phi_N(\frac{\ln(h) - \mu_N}{\mu_N})$$
12

 $\mu_N = 7.60$

 $\sigma_N = 2.75$

if $(w < 0.5 m/h \text{ or } (w \ 0.5m/h \text{ and } h < 2.1 m))$ and $(hv < 7 \ m^2/s \text{ or } v < 2 m/s)$

Ehsan (2009) improved criteria for defining flood severity using a geometric aggregate (GA) as shown in equation 13.

$$LOL_{i} = PR_{i} \times FAT_{BASE} \times F_{SV} \times F_{age} \times F_{mt} \times F_{st} \times F_{h} \times F_{war} \times F_{ev}$$
13

Where

 LOL_i = Loss of life at a particular location "i" downstream of the dam

 PR_i = Population at risk at a particular location "i" downstream of the dam

 F_{SV} : Flood severity factor (in terms of the probability of life loss due to collapse of buildings).

High Severity very likely 1.0Medium Severity unlikely 0.3Low Severity very unlikely 0.1

 F_{age} : Age risk factor depending on different age groups in P_R; three age groups have been defined, number of people in group A (<10yrs & >=65yrs), B (10-15) yrs and C (15-64) yrs. This factor will change from 1.0 with respect to the likelihood of different age groups

 F_{mt} : Material risk factor. For flood severity indication, materials that are frequently used for house construction such as concrete, bricks, masonry etc.

$$F_{mt} = 1 \times X\% + 1.5 \times Y\% \qquad (general form) \qquad 14$$

Where

X = % of other types of houses

Y = % of very low strength houses

 F_{st} : Storey risk factor: It is assumed that all types of houses (single and more storeys) will be damaged during high flood severity, because upper storeys can also collapse due to the failure of the ground storeys. For medium and low severity cases, more storey houses could provide refuge to people in upper storeys and reduce the overall risk. So, the suggested relation for the storey risk factor is,

$$F_{st} = 1$$
 (for high severity and all types of houses) 15

$$F_{st} = 1 - S\%$$
 (for medium and low severity) 16

S = % of more storey houses

 F_h : Health risk factor: An average value of 1.0 was assumed with FAT_{BASE} for normal healthy P_R. The overall risk would increase with respect to the percentage of disabled persons within the inundation area.

$$F_h = 1 \times H\% + 1.25 \times D\% \qquad (\text{general form}) \qquad 17$$

H = % of P_R with avg. health

D = % of disabled P_R

 F_{war} = Warning factor depending on the initiation of warning and flood travel time. Interpreting Graham (1999) warning definitions:

Warning time	Flood severity understanding	F _{war}
No	No	1
Some (15-60 min)	Vague/unclear	0.7
Adequate (> 60 min)	Precise/clear	0.3

 F_{ev} = Ease of evacuation factor: Depends on the warning efficiency and the available evacuation facilities, transportation. In principle, this factor would be different for urban and rural areas. At the moment, no empirical value or guideline is available for ease of evacuation. So, this factor has been defined quantitatively as the likelihood of no rescue for all combinations of population at risk (P_R) with respect to warning efficiency.

Warning time	Ease of evacuation	F _{ev}
No	No	1
Some (15-60 min)	Some	0.7
Adequate (> 60 min)	Good	0.3

Definition of Parameters/Variables, based on a combination of the various references cited earlier.

- Warning Time: The time available to alert potentially affected individuals before the dam failure occurs. Measured in minutes or hours.
- Dam Size: The size of the dam, specifically its height and capacity to store water, is a crucial factor in determining the potential magnitude of a dam failure. The height and structural integrity of the dam influence the mode of failure, such as overtopping or breach.
- Inundation Mapping: Inundation mapping refers to the process of creating maps that depict the areas likely to be flooded in the event of a dam failure. These maps are based on hydraulic and hydrological modelling, taking into account various factors like topography, flow velocities, and floodwater depths. Inundation maps provide critical information about the extent and spatial distribution of floodwaters, helping emergency responders and communities to understand which areas are at risk and plan evacuation routes accordingly.
- Velocity of Flow: The speed at which the floodwater moves, impacting evacuation and safety measures. Mostly Measured in meters per second (m/s).
- Water Depth: Measured in meters (m), it specifies the depth of floodwater, which influences the severity of the impact.
- Flood severity refers to the extent or magnitude of the flood caused by a dam failure. It is a
 measure of how intense or severe the flood event is, and it plays a critical role in determining
 the potential impact on human lives. The severity of the flood can be influenced by various
 factors, including the volume of water released during the dam failure, the rate of flow, the
 topography of the downstream area, the presence of natural or man-made obstacles that may
 hinder the flow, and the characteristics of the floodwater (e.g., presence of sediments or debris).
 The higher the severity of the flood, the greater the potential risk to human life and property in
 the affected areas.
- Population at Risk (P_R): The number of people living in the downstream area of the dam who could be affected.
- Activities of P_R : Assessed by categorizing the nature of activities (residential, industrial, recreational, etc.) in the downstream area that could increase the vulnerability of the population.
- Monitoring Capabilities: Based on the ability to monitor dam conditions and issue timely alerts.

- Flood Warning System: The existence and effectiveness of a system to warn downstream communities about the potential dam failure.
 - Warning Rate, Extent, and Effect: Measured by determining how many people receive warnings, how widespread the warning coverage is, and the impact of the warning on evacuation.
 - Prior Awareness: Assessed through surveys or interviews to gauge the level of awareness among the affected population about the potential risks.
 - Psychological Impressions: it examines how individuals perceive and respond to the threat, which can impact evacuation decisions.
 - Personal Mobilization Time: The time it takes for individuals to respond to the warning and begin evacuating, Measured in minutes.
 - Evacuation Rate/Ease of Evacuation: Measured by calculating the speed and effectiveness of evacuation processes.
 - Topography: Assessed through geographical surveys and mapping, it relates to the physical features of the terrain, which can affect floodwater flow and evacuation options.
 - Urban vs. Rural: Assessed by categorizing areas into urban and rural, which may influence the density of population and infrastructure.
 - Age: The age distribution of the population at risk, as vulnerability can vary with age.
 - Time of Day: The time when the dam failure occurs, affecting the readiness and response of the population.
 - Environmental Conditions: refer to the prevailing weather, hydrological, and ecological factors that exist at the time of a dam failure. These conditions can significantly influence the dynamics of the flood, the speed and direction of water flow, and the extent of damage caused. Environmental conditions may include rainfall patterns, soil saturation levels, wind speed, temperature, and the presence of other natural hazards in the area (e.g., earthquakes or landslides). Understanding these conditions is crucial for accurate modelling and prediction of the flood's behaviour and the potential impacts on downstream communities.

Tabular Comparison of Variables for Estimating Loss of Life

Table 2 offers a representative overview of significant variables utilized by different LOL models. While some of these variables were acknowledged as important by the authors, they may not have been directly utilized in their model (primarily due to insufficient data in many instances).

	Brow Graha (1988	rown and DeKay & raham McClelland (1993)		Grah (199	Graham (1999)		Reiter (2001)		Assaf (2002)		Aboelata et al., (2003); Aboelata &		3)	Jonkman et al. (2008)		
												(2008)				
Variable	Ι	Μ	Ι	М	Ι	M	Ι	M	Ι	Μ	Ι	M	Ι	M	Ι	М
Dam/reservoir/breach size	*				*		*	*					*	*	*	
Inundation mapping	*				*	*	*	*					*			
Depth of flooding	*		*		*	*	*	*	*	*	*	*	*	*	*	*
Velocity of flow	*		*		*	*	*	*	*	*	*	*	*	*	*	*
Flood severity	*		*	*	*	*	*	*				*	*	*		*
Damage to structures	*				*	*	*	*	*	*		*			*	*
Population at risk (P_R)	*	*	*	*	*	*	*	*	*	*	*	*	*		*	*
Activities of P _R					*		*		*	*						*
Warning time	*	*	*	*	*	*	*				*	*	*	*	*	
Monitoring capabilities	*		*		*						*	*			*	
Flood warning system	*		*		*						*	*				
Warning rate, extent and effect	*				*						*	*			*	
Prior awareness	*				*											
Psychological impressions	*								*	*			*			
Personal mobilization time									*	*	*	*				
Evacuation rate/ease of evacuation	*				*		*		*	*	*	*			*	*
Topography			*												*	
Urban vs. Rural									*							
Age	*								*	*					*	
Time of day	*						*						*	*		
Environmental conditions	*				*								*	*		

Table 2 - Review of variables for estimating loss of life by different authors. (I = identified by the author(s), M= used in the model)

	Es	shan	Johnsto	ne & (2012)	Peng an	Peng and Zhang (2012)		Huang et al. (2017); Mahmoud et al. (2020)			Ge	et
	(2	009)	Lence,	(2012)	Zhang (2012)	Mannouc	i et al. (2020)	(20	18)	(20)	19)
Variable		Μ	Ι	Μ	Ι	Μ	Ι	М	Ι	Μ	Ι	Μ
Dam/reservoir/breach size	*						*	*	*	*	*	*
Inundation mapping												
Depth of flooding	*	*	*	*	*	*	*	*	*	*		
Velocity of flow	*	*	*	*	*	*	*	*	*	*		
Flood lethality	*	*	*	*	*	*	*	*	*	*	*	
Damage to structures		*	*	*	*	*	*	*	*		*	*
Population at risk (P_R)	*	*	*	*	*	*	*	*	*	*	*	*
Activities of P _R												
Warning time	*	*	*		*	*	*	*	*	*	*	*
Monitoring capabilities	*											
Flood warning system	*											
Warning rate, extent and effect			*		*		*	*	*			
Prior awareness			*						*			
Psychological impressions												
Personal mobilization time			*	*					*			
Evacuation rate/ease of evacuation	*	*	*		*	*	*	*	*			
Topography	*								*		*	
Urban vs. Rural	*											
Age	*	*							*			
Time of day			*	*	*	*	*	*	*	*	*	
Environmental conditions							*	*	*		*	

Although different models have yielded results that are within acceptable limits, a review of the variables used for loss of life (LOL) estimation is essential to understand the current practices and identify areas for improvement.

The authors cited in Table 2 have tried to estimate the LOL as accurately as possible with their various models, giving attention to variables/parameters they consider important and discarding less significant ones. Newer models show a significant rise in the number of variables utilized compared to older models.

Warning time, flow velocity, and water depth can be considered the most important parameters in estimating loss of life from a dam failure, as they are used in 80% of the various models in this literature.

Validation of relevant and significant parameters/variables is through empirical studies and data analysis. That would mean analysing historical dam failure events or conducting virtual simulations to understand how specific variables influence the actual outcomes and assessing how well the chosen variables correlate with the observed loss of life.

Our model incorporates additional variables to achieve a more comprehensive estimation of the potential loss of life (LOL) in dam failure scenarios. By integrating these extra parameters, we aim to improve the model's accuracy and provide a more thorough assessment of the potential impacts on human lives. These variables include:

- Dam Failure Mechanism: The specific mode of dam failure (e.g., breach, overtopping) which can significantly affect the characteristics of the flood.
- Dam Type and Design: Different dam types may lead to distinct failure patterns and consequences.
- Sediment Transport: The presence of sediments (debris, slurry in tailings) in the floodwater, affecting flow dynamics and potential hazards.
- Hazard Mapping/inundation mapping: The existence of accurate hazard maps to understand and communicate potential risks.
- Climate Change Impact and environmental conditions: Environmental conditions refer to the prevailing weather, hydrological, and ecological factors that exist at the time of a dam failure. These conditions can significantly influence the dynamics of the flood, the speed and direction of water flow, and the extent of damage caused. Climate change impact focuses on the long-term changes in climate patterns and how they might affect the likelihood and severity of dam failures over time.

- Community Preparedness or prior awareness: The level of preparedness and awareness among the population at risk, influencing their ability to respond effectively.
- Communication Infrastructure: The availability and reliability of communication channels to disseminate warnings and information to at-risk populations, emergency responders, and relevant stakeholders.
- Emergency Response Capabilities: The effectiveness of emergency response services and agencies in managing the crisis.

The novelty of this proposal lies in its comprehensive analysis of the variables used in LOL estimation models and the identification of areas for improvement. It begins by acknowledging the existence of various models that estimate LOL, then proposes a model that integrates additional variables to achieve a more comprehensive estimation of LOL in dam failure scenarios. By expanding the scope of variables considered, it addresses various aspects related to infrastructure, community, dam-specific factors, communication, vulnerability, and climate change. This broader perspective can enhance the accuracy of the LOL estimation model and also provide a more thorough assessment of the potential impacts on human lives in the event of a dam failure.

Generally, this work aims to contribute significantly to the field of dam safety by providing a more robust and comprehensive framework for predicting the potential loss of life and mitigating the impacts of dam failures.

3 AN IMPROVED ESTIMATION MODEL FOR DAM FAILURE INDUCED LOSS OF LIFE

Samuel Ovu¹ and Mauricio Dziedzic^{1,2}

¹ School of Engineering, Faculty of Science and Engineering, University of Northern British Columbia, Prince George, BC, V2N 4Z9 ² Mauricio.Dziedzic@unbc.ca

This chapter is an expanded version of the article "An Improved Estimation Model for Dam Failure Induced Loss of Life" presented in the Canadian Society for Civil Engineers 2024 annual conference and will be submitted to a peer reviewed journal such as Springer's Stochastic Environmental Research and Risk Assessment.

Abstract. The potential loss of life (LOL) resulting from dam failures represents a critical concern in the field of dam safety and disaster management. The accurate estimation of LOL is paramount for informed decision-making, emergency preparedness, and the minimization of human casualties in such catastrophic events. The purpose of this paper is to propose an improved model for LOL estimation specific to North American dam failure cases. The study involves a review of existing literature, selecting a model, and refining the chosen model to improve its predictive capabilities for LOL. The approach categorizes dam failure into subcases based on flood severity and the distance from the dam. It then identifies and filters the more important influencing variables. Subsequently, two empirical equations that serve as the calculation method for LOL formulated through multivariate regression analysis are derived using thirty-two dam failure subcases. The datasets were split into train and test sets, yielding R² values of 0.9949 for low severity cases and 0.9955 for medium-high severity cases on the test sets. Graham's model was selected as a comparison benchmark due to its straightforward formula, established use in LOL estimation, and minimal data requirements. The successful implementation of this model suggests its potential applicability for diverse regions, contributing to improved disaster preparedness and response strategies, as well as enhancing dam safety and community well-being downstream of dams.

Keywords: Flood, Dam breach, life loss prediction.

3.1 INTRODUCTION

According to the International Commission On Large Dams (ICOLD, n.d.), dams play a crucial role in the development and management of water resources, providing a range of benefits that contribute to societal progress. However, their construction and operation are not without challenges. Dam failures can potentially unleash catastrophic floods, leading to the loss of human lives, damage to infrastructure, and severe environmental consequences (Ehsan, 2009; Graham, 1999). The potential loss of life (*LOL*) resulting from dam failures is a critical concern in the field of dam safety and disaster management, and it is imperative to develop accurate and reliable methods for its estimation (Jonkman et al., 2008).

By quantifying the potential risks, engineers can gauge the potential consequences of dam breaches, identify vulnerabilities, and make necessary improvements to dam design and operation, and authorities can further develop effective emergency response plans, make informed decisions regarding evacuation zones and allocate resources efficiently (Public Safety Canada, 2017).

LOL estimation is also vital for legal and insurance purposes, aiding in determining liability, compensation, and insurance coverage for affected individuals and properties.

Despite its critical significance, existing *LOL* estimation models for North American regions still have limitations regarding accuracy and comprehensiveness. Therefore, the aim of this paper is to contribute to this evolving field of *LOL* estimation. This will be achieved by reviewing existing models and refining a chosen model to enhance its predictive capabilities for *LOL*.

Life loss estimation models from dam failure have been developed by several authors. Brown & Graham (1988) proposed a formula based on the size of the population at risk (P_R) from failure and the warning time available for that population. DeKay & McClelland (1993) proposed an equation for *LOL* estimation, considering flood severity, the size of the population at risk and warning time. Graham (1999) improves *LOL* prediction by adding preparedness as a factor. Reiter (2001) proposed the RESCDAM method based on Graham's principle, by introducing more factors, such as vulnerability of the P_R and rescue conditions. Assaf (2002) take the behaviour of the P_R into consideration to develop BC Hydro's life safety model (LSM). Jonkman et al. (2008) considered hydraulic characteristics of the flood, evacuation rate and warning time. Ehsan (2009) introduced a new criteria for classification of flood severity. Peng

& Zhang (2012) developed the human risk analysis model (HURAM), taking into account factors such as evacuation time, distance to dam and time of day. Ge et al. (2019, 2020) used a combination of catastrophe theory and interval theory to select the most important influencing factors and to estimate upper and lower limits of *LOL* rather than a single value. Ge et al. (2022) further improved that work by including the time required for evacuation.

Huang et al. (2017) and Mahmoud et al. (2020) developed a more comprehensive model for predicting *LOL* resulting from dam failures in China. Their approach categorizes dam failure into subcases according to the severity of the flood. It then identifies and filters the more important influencing variables. Subsequently, two empirical equations are formulated through multivariate nonlinear regression. These equations serve as the calculation method for *LOL*, with one tailored for low-severity cases and the other for medium and high-severity cases. Their model shows superior performance when compared with results documented by DeKay & McClelland (1993) and Graham (1999). Thus, Huang et al. (2017); Mahmoud et al. (2020) model was selected as the starting point to propose a similar approach for North America.

3.2 Failure database

3.2.1 Collection and processing

The data employed in this study were collected from various sources (ASDSO, n.d; Graham, 2008; Jay Hilary Kelley et al., 1973; Joseph A. Strahl et al., 1972; Larimer & Department of the Interior, 1973; Logan County Genealogical Society, 1972; Meservy, 1968; National Weather Service, n.d.; NOAA, 1972; Spero et al., 2022; U. S. Census Bureau, n.d.; US Army Corps of Engineers, 2015; USGS & NOAA, 1975; Wahl, 1998; Weather Underground, 2024; Williamm E. Davies et al., 1972): 1) existing data on historic cases available in the literature and GIS databases, 2) existing data from institutional bodies (National committees, governmental agencies, municipal data), and 3) simulation-derived data. The dam failure sites selected for the study were divided into sub-cases based on the severity of the flood, which is indirectly influenced by the distance from the dam, to give an accurate representation at different locations downstream of the dam failure. Therefore, a total of 32 subcases were analysed (10 low-severity subcases and 22 medium or high-severity subcases) and are listed in Table 3. These sites are selected to include dams ranging from low to extreme failure consequences as classified by the Canadian Dam Association (Environment Alberta, 2016). Based on dam height (ICOLD, 2011), the cases include small to large dam ranges (6.1 - 92)

metres) as well as a range of fatality rates (0.0 - 0.5) and reservoir size from 0.49 - 310 million m³ of water.

Incomplete or missing information such as flood severity, not available in existing sources was derived from the outcome (water depth, flow velocity) of two-dimensional hydrodynamic simulations of unsteady flow using HEC-RAS. This was achieved by leveraging a reported, known or estimated severity of flood (S_F) at a specific point along the flood path to estimate conditions at other locations downstream of the flood reach.

3.3 Methodology

The calculation steps are divided into: classification and categorization of the influencing factors; normalization and weight calculation using the entropy method; Multivariate Regression Analysis. LOL equations are obtained for a) low severity and b) medium and high severity cases, respectively.

3.3.1 Classification of influencing factors affecting LOL

A review of variables influencing *LOL* is essential for identifying areas for improvement. Peng, Ge, Huang, Mahmoud (Ge et al., 2019; Huang et al., 2017; Mahmoud et al., 2020; Peng & Zhang, 2012) and several others cited earlier have tried to accurately estimate the *LOL* with their various models, giving attention to variables they consider important and discarding less significant ones, with newer models showing a significant rise in the number of variables utilized compared to older models. In this work, influencing factors are selected based on Mahmoud et al. (2020) classification into four categories: hazard factors, exposure factors, population-related factors and rescue capability factors (Figure 1).



Figure 1. Influencing factors for estimating life loss due to dam break (from Mahmoud et al., 2020).

Table 3. Dam failure cases

* Low flood severity case

 H_D is the height of the dam; S_F is the flood severity; M_B is the dam break mode; T_B is the time of the dam breach; W_B is the weather at the time of the dam breach; V_B is the Vulnerability of Buildings; D_D is the distance from the dam; P_R is the population at Risk; E_C is the Effectiveness of evacuation conditions; T_W is the warning time; U_B is people's understanding of a dam break; Lol is the loss of life resulting from failure of the dam.

	Dam	SUBCASE	H _D (m)	$\begin{bmatrix} S_F \\ (m^2.s^- \\ 1 \end{bmatrix}$	M _B	$\begin{bmatrix} S_W \\ (x10^4 \\ m^3) \end{bmatrix}$	T _B	WB	VB	D _D (km)	P _R	U _B	Tw	E _C	LOL
1	Johnstown South Fork 1	Johnstown	38.30	17.88	1	1890	0.33	0.4	0.835	22.05	37700	1	1	1	2209
2*	Canyon lake dam	Rapid City	6.10	3.71	1	99	0.67	0.8	0.835	11.27	43000	0.5	1	1	238
3	Teton Dam	Mouth of Teton Canyon	92.96	26.95	0.75	31047	0.33	0.2	1	8.05	2	1	1	1	1
4		Wilford	92.96	10.08	0.75	31047	0.33	0.2	0.67	13.52	536	1	0.8	1	5
5		Town of Teton	92.96	8.83	0.75	31047	0.33	0.2	0.67	14.16	1466	1	0.6	0.67	0
6*		Sugar city	92.96	3.71	0.75	31047	0.33	0.2	0.67	19.79	2987	1	0.2	0.67	0
7*		Rexburg	92.96	2.87	0.75	31047	0.33	0.2	0.67	24.62	4767	1	0.2	0.33	0
8	Buffalo Creek	Saunders	13.41	14.16	0.75	49	0.33	0.4	0.835	1.5	429	1	1	1	18
9	Dam	Lorado	13.41	9.22	0.75	49	0.33	0.4	0.835	6.78	618	1	1	1	22
10*		Lundale	13.41	2.37	0.75	49	0.33	0.4	0.835	9.57	997	1	1	1	52
11*		Stowe	13.41	2.62	0.75	49	0.33	0.4	0.835	9.66	128	1	1	1	2
12*		Latrobe	13.41	2.95	0.75	49	0.33	0.4	0.835	13.2	258	1	0.6	0.67	8
13*		Accoville	13.41	1.77	0.75	49	0.33	0.4	0.835	19.32	2070	1	0.6	0.67	0
14	St. Francis, Ca	Edison Camp	60.00	52.86	0.75	4687.2	1	0.2	0.835	29.9	150	1	1	1	84
15*	Mohegan Park Dam (Spaulding Pond Dam)	Norwich	6.10	4.18	0.75	17	0.67	0.8	0.67	3.21	500	0.5	1	1	6
16*	Meadow Pond / Bergeron Pond Dam	Alton, Nh	11.00	0.65	0.75	34.8	0.33	0.2	0.835	1.26	50	0.5	1	1	1
17*	Lee Lake	Lee, Ma	7.60	4.18	0.75	37	0.33	0.2	0.835	4.02	80	0.5	1	0.67	2
18	Baldwin Hills, Ca	Dam to Sanchez Drive	20.10	18.58	0.75	95	0.33	0.2	0.67	0.8	100	0.5	0.2	0.33	0

19		Sanchez Drive onwards inc. Green Village	20.10	13.66	0.75	95	0.33	0.2	0.835	4.8	16400	0.5	0.2	0.33	5
20	Kelly Barnes Dam	Toccoa Falls Bible College	11.60	7.9	0.75	77.7	1	0.8	0.67	1.6	100	0.5	1	1	39
21	Mill River	Williamsburg	13.10	9.75	0.75	227.12	0.33	0.2	0.835	4.82	888	1	1	0.67	138
22	Austin Dam (Bayless Dam) I	Austin, Pa	14	11.15	0.75	86.3	0.33	0.2	0.835	2.41	900	1	1	1	78
23	Laurel Run Dam, Pa	Johnstown	12.80	11.15	1	55.5	1	0.6	0.835	2.4	150	1	1	1	40
24	Timberlake Dam, VA	Lynchburg	10.05	5.57	1	178.7	0.67	0.8	0.835	1.5	7	1	1	1	2
25	Little Deer Creek Dam, UT	-	26.21	14.59	0.75	136	0.33	0.2	1	11.58	50	1	1	1	1
26	Lawn Lake Dam	Roaring River	7.90	16.21	0.75	79.8	0.33	0.2	1	6.437	25	1	1	1	1
27	Cascade Lake Dam	Aspen Glen Campground	5.20	11.24	1	3.1	0.33	0.2	1	11.265	275	1	0.8	0.67	2
28	Lawn Lake & Cascade Valley	Fall River Road / Estes Park	5.20	6.6	1	3.1	0.33	0.2	0.835	19.31	4000	0.5	0.6	0.33	0
29	Bear Wallow Dam	Ashville, NC	10.97	6.97	0.75	4.93	1	0.6	0.835	1.28	8	0.5	1	1	4
30	walnut grove dam	Another New Dam	33.00	33.00	1	1100	1	0.6	1	24.14	125	0.5	1	0.67	85
31	Swift Dam No. 2, MT	Birch Creek	47.85	47.85	1	4193.8	0.33	0.8	0.835	8	250	1	1	0.33	19
32	Castlewood Canyon Dam II	Denver	21.30	21.30	1	617	1	0.8	0.67	60	5000	0.5	0.2	0.33	2

The severity of flooding, S_F is calculated by equation 18 as outlined by (Graham, 1999; Huang et al., 2017). If the flow rate Q and max width of the water surface, W is known, S_F can be determined from equation 19.

$$S_F = hv \tag{18}$$

$$S_F \approx Q_{Top} / W_{Max} \tag{19}$$

Where h = flood depth (m), v = flow velocity (m.s⁻¹), $Q_{Top} =$ peak flood flow (m³.s⁻¹), $W_{Max} =$ top-width of flow (m).

 S_F is the variable used to categorize the severity into low or medium/high. According to Graham (1999); Li et al. (2006), as Cited in Huang et al. (2017), S_F is neglected if it is <0.5 m²/s, while low, medium and high severity cases are based on the range of values, $0.5 \text{ m}^2/\text{s} \le S_F < 4.6 \text{ m}^2/\text{s}$, $4.6 \text{ m}^2/\text{s} \le S_F < 12.0 \text{ m}^2/\text{s}$ and $S_F \ge 12.0 \text{ m}^2/\text{s}$, respectively.

The dam break mode, M_B is ranked based on four frequent break modes (Graham, 1999; Huang et al., 2017):

- o Overtopping
- Poor quality (leakage, internal erosion, tunnel, blockage or obstruction of dam structure, spillway etc.)
- Mismanagement (over storage, poor or lacking maintenance or management)
- \circ Others.

The time of the dam breach, T_B is divided into three main periods: daytime-work period (8:00–20:00); night rest period (20:00–24:00); and midnight sleep period (24:00–8:00) (Li et al., 2019; Mahmoud et al., 2020).

The weather at the time of the dam breach, W_B can contribute to the level of severity and extent of the consequences such as precipitation intensity, visibility, communication, wind speeds and temperature, making it difficult for evacuation and rescue. The weather conditions are reported by Mahmoud et al. (2020) as cited by Huang et al. (2017) and are divided into:

- o Level I (extreme weather: storm, blizzard, typhoon, fog, haze)
- Level II (heavy rain, heavy snow, gale, etc.)
- o Level III (moderate rain, moderate snow, etc.)
- o Level IV (light rain, shower, light snow, etc.)
- o Level V (sunny or cloudy day).

Building Vulnerability, V_B is classified into adobe, wood or masonry and concrete (Mahmoud et al., 2020; Reiter, 2001). Here, wood and masonry are separated into different classes to provide a more precise classification in building vulnerability. Herein, V_B is classified into: Adobe, Wood, Masonry and Concrete.

The distance from the dam, D_D plays a significant role in determining the response time available for affected communities to evacuate and seek safety. Therefore, the greater the D_D , the lesser the *LOL* (Huang et al., 2017).

The population at Risk, P_R is the total number of people within the inundation area (Brown & Graham, 1988).

The grading of all the influencing factors have been defined in Table 4.

Table 4. Grading standard of influencing factors. Modified from (Ge et al., 2019; Huang et al., 2017; Mahmoudet al., 2020; Peng & Zhang, 2012).

Factors	Description	Grade
Weather at breach	Storm	1.0
	Heavy rain/snow	0.8
	Moderate rain/Snow	0.6
	Light rain/Snow	0.4
	Sunny	0.2
Evacuation Condition	Bad	1
	Middle	0.67
	Good	0.33
People understanding	Vague/fuzzy	1.0
	Clear/precise	0.5
Building vulnerability	Adobe	1.0
	Wood	0.835
	Masonry (brick, stone)	0.67
	Concrete	0.33
Warning time (min)	0-15	1.0
	15-30	0.80
	30-45	0.60
	45-60	0.40
	> 60	0.20
Breach time	Midnight	1.0
	Night rest	0.67
	Day time	0.33
Break mode	Overtopping	1.0
	Poor quality	0.75
	Mismanagement	0.50
	Others	0.25

The people's understanding of the dam break, U_B will play a role in *LOL*, as a better or clearer understanding will lead to lesser loss of life (Brown & Graham, 1988). This is divided into clear or precise understanding and vague or fuzzy understanding (Graham, 1999) and can be understood as level of preparedness.

The warning time, T_W refers to the duration between the identification of an imminent dam breach and the actual arrival of floodwaters downstream. The amount of warning time significantly affects the ability of affected communities to respond, evacuate, and seek safety (Huang et al., 2017; Jonkman et al., 2008; Mahmoud et al., 2020). T_W has been divided into: 0–15, 15–30, 30–45, 45–60 and >60 min (Huang et al., 2017; Jonkman et al., 2008).

Effectiveness of evacuation condition, E_C plays a crucial role in determining the potential *LOL* and is influenced by terrain and environment (Huang et al., 2017). This is divided into good, middle and bad evacuation conditions (Johnstone W M et al., 2005).

3.3.1 Normalization and grading of influencing factors.

Influencing factors that do not have a numeric value are normalized by assigning them to a standard grade of values up to 1 (Ge et al., 2019; Huang et al., 2017; Mahmoud et al., 2020; Peng & Zhang, 2012).

The normalization of variables with numeric values, that is H_D , S_w , D_d , S_f , P_R is done using eq. 20 (Mahmoud et al., 2020).

$$r_{ij} = x_{ij} / max \{x_{ij}\}$$
⁽²⁰⁾

where *i* is the dam break case, *j* is the influence factor number, and r_{ij} is the normalized value. After normalization, the normalized judgement matrix is represented by eq. 21.

$$R = \left[r_{ij}\right]_{m \times n} \tag{21}$$

And x_{ij} is the initial value of the influencing factor in the judgment matrix (22).

$$x_{ij} = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}_{m \times n}$$
(22)

Where m is the number of dam break events, that is, m = 10 and 22 for the low subcase and medium and high severity subcases, respectively. The influencing factors are grouped into g=4 modules: hazard factors, exposure factors, population-related factors, and rescue capability factors. For these modules, n = 4, 4, 2, 2, respectively.
3.3.2 Entropy calculation

Mahmoud et al. (2020) present the jth factor entropy (Hj) (Eq. 23).

$$H_{j} = -\frac{\sum_{i=1}^{m} (f_{ij} ln f_{ij})}{lnm}$$
(23)

Where

$$f_{ij} = \frac{r_{ij}}{\sum_{i=1}^{m} r_{ij}}$$
(24)

If any influencing factor affecting a dam break case has an equal probability, the entropy is maximum and equal to 1. As such, its utility for analysis is zero. Therefore, the utility value of an influencing factor depends on the difference between 1 and the information entropy, $1 - H_j$ (Huang et al., 2017; Mahmoud et al., 2020).

3.3.3 Weight calculation, filtration and comprehensive score calculation

The weight of the jth factor entropy is calculated using Eq. 25. Factors with weight <5% can be neglected when calculating the comprehensive score (Huang et al., 2017; Mahmoud et al., 2020).

$$w_{j} = (1 - H_{j}) / \sum_{i=1}^{n} (1 - H_{j})$$
(25)

The comprehensive score is calculated for the four modules with m dam break cases (m = 10, 22) using Eq. 26.

$$Y_k = \sum_{j=1}^n r_{kj} \times w_j \tag{26}$$

3.3.4 Nonlinear regression analysis

Multivariate regression analysis is the final step in this *LOL* estimation model, it involves exploring the relationships between the fatality rate, F_L , and the four modules (Y_1 , Y_2 , Y_3 and Y_4). To capture both linear and nonlinear relationships, we evaluate different combinations of regression functions to optimize for the best fit using the coefficient of determination (R^2). Here, the functions used are reciprocal, square root, cubic, exponential, logarithmic, power, and linear functions, which allow a diverse combination and increase the possibility of achieving a good fit, allowing to explore different relationships between F_L and the modules. The key objective for this combination is that it explores all possible combinations of the seven transformation functions across the four independent variables.

In addition to these regression functions, the four modules are combined using second-degree polynomials.

The number of combinations tested in this analysis highlights the comprehensive nature of the model. With seven different regression functions tested for each of the four variables, the total number of possible combinations becomes 2401.

Each combination was evaluated by randomly splitting the dataset into training and testing subsets, ensuring a more robust and reliable model through data transformation and multiple splits. Unlike our approach, some authors chose to fit their entire datasets without splitting, likely due to limited data availability. While this may create a strong model fit, it risks overfitting, where the model performs well on training data but poorly on unseen data.

To identify the best dataset randomization for optimal model performance, we iteratively explored all possible dataset splits. For the 10 low-severity datasets, we generated 45 unique combinations, while the 22 medium- and high-severity datasets yielded 26,334 combinations. Each iteration ran a loop to find the optimal randomization state based on the highest R² score on the test data. This process was repeated separately for both 70/30 and 80/20 train-test splits to determine the most effective ratio. Python coding was developed to implement this procedure, using the NumPy and Pandas libraries for data manipulation, and Scikit-learn for regression analysis.

3.4 Results

The normalized judgement matrix, $\mathbf{R} = [\mathbf{r}_{ij}]_{m \times n}$ from equation 21 is presented as $\mathbf{R}_1 \dots \mathbf{R}_4$ for low severity and medium/high severity cases for the four influencing factor categories, respectively, in equations 27 & 28.

S/N	$\begin{bmatrix} H_D \end{bmatrix}$	S_F	M _R	S_W		$\begin{bmatrix} T_B \end{bmatrix}$	W_B	V_{P}	D_D	
1	0.0656	0.8876	1	0.0032		1	1	1	0.4578	
2*	1	0.8876	0.75	1		0.4925	0.25	0.8024	0.8038	
3	1	0.6866	0.75	1		0.4925	0.25	0.8024	1	
4	0.1443	0.5670	0.75	0.0016		0.4925	0.5	1	0.3887	
5	0.1443	0.6268	0.75	0.0016		0.4925	0.5	1	0.3924	
6*	0.1443	0.7057	0.75	0.0016		0.4925	0.5	1	0.5361	
7*	0.1443	0.4234	0.75	0.0016		0.4925	0.5	1	0.7847	
8	0.0656	1	0.75	0.0005		1	1	0.8024	0.1304	
9	0.1183	0.1555	0.75	0.0011		0.4925	0.25	1	0.0512	
10*	0.0818	1	0.75	0.0012		0.4925	0.25	1	0.1633	
11*	0.4120	0.3383	1	0.0609		0.3300	0.5	0.8350	0.3675	
12*	1	0.5098	0.75	1		0.3300	0.25	1	0.1342	
13*	1	0.1907	0.75	1		0.3300	0.25	0.6700	0.2253	
14	1	0.1670	0.75	1		0.3300	0.25	0.6700	0.2360	
15*	0.1443	0.2679	0.75	0.0016		0.3300	0.5	0.8350	0.0250	07
$16^* R_1 =$	= 0.1443	0.1744	0.75	0.0016	$R_2 =$	0.3300	0.5	0.8350	0.1130	27
17*	0.6454	1	0.75	0.1510		1	0.25	0.8350	0.4983	
18	0.2162	0.3515	0.75	0.0031		0.3300	0.25	0.6700	0.0133	
19	0.2162	0.2584	0.75	0.0031		0.3300	0.25	0.8350	0.0800	
20	0.1248	0.1495	0.75	0.0025		1	1	0.6700	0.0267	
21	0.1409	0.1844	0.75	0.0073		0.3300	0.25	0.8350	0.0803	
22	0.1506	0.2109	0.75	0.0028		0.3300	0.25	0.8350	0.0402	
23	0.1377	0.2109	1	0.0018		1	0.75	0.8350	0.0400	
24	0.1081	0.1054	1	0.0058		0.6700	1	0.8350	0.0250	
25	0.2819	0.2760	0.75	0.0044		0.3300	0.25	1	0.1930	
26	0.0850	0.3067	0.75	0.0026		0.3300	0.25	1	0.1073	
27	0.0559	0.2126	1	0.0001		0.3300	0.25	1	0.1878	
28	0.0559	0.1249	1	0.0001		0.3300	0.25	0.8350	0.3218	
29	0.1180	0.1319	0.75	0.0002		1	0.75	0.8350	0.0213	
30	0.3550	0.6243	1	0.0354		1	0.75	1	0.4023	
31	0.5147	0.9052	1	0.1351		0.330	0 1	0.8350	0.1333	
32	0.2291	0.4030	1	0.0199		[1.00	1.00	0.6700	1.00	

Mahmoud et al. (2020) elected to neglect the influence of parameters whose calculated weight result is smaller than 5% for the comprehensive score (Y_k) calculation. In the low severity cases S_F , M_B , U_B and V_B have weights less than 5%, while M_B , U_B and V_B are below the threshold in the medium/high severity case. This is presented in Figure 2 & 3 respectively. Here, we chose to utilize all the weights including those below the 5% threshold, to ensure general representation across the categories, and not to allow for cumulatively neglecting a large percentage of the influencing factors.



Figure 2: weight of influencing factors for the low severity case



Figure 3: weight of influencing factors for the medium and high severity case

3.4.1 Regression Analysis

Based on the highest R², the best combination of the regression function for low severity is logarithmic, square root, exponential, square root. For the medium and high severity, power, reciprocal, reciprocal and exponential are the best functions. the proposed equation for low flood severity ($0.5 \text{ m}^2/\text{s} \le \text{S}_F < 4.6 \text{ m}^2/\text{s}$) and medium and high severity cases ($S_f > 4.6 \text{ m}^2/\text{s}$) is presented in Eq. 29 &30 respectively, using second-degree polynomials to combine the four Y's.

$$F_{L,l} = \beta_0 + \beta_1 \times LN(Y1) + \beta_2 \times \sqrt{Y2} + \beta_3 \times e^{Y3} + \beta_4 \times \sqrt{Y4} + \beta_5 \times LN(Y1)^2 + \beta_6 \times LN(Y1) \cdot \sqrt{Y2} + \beta_7 \times LN(Y1) \cdot e^{Y3} + \beta_8 \times LN(Y1) \cdot \sqrt{Y4} + \beta_9 \times (\sqrt{Y2})^2 + \beta_{10} \times \sqrt{Y2} \cdot e^{Y3} + \beta_{11} \times \sqrt{Y2} \cdot \sqrt{Y4} + \beta_{12} \times (e^{Y3})^2 + \beta_{13} \times e^{Y3} \cdot \sqrt{Y4} + \beta_{14} \times (\sqrt{Y4})^2$$
(29)

$$F_{L,m\&h} = \beta_{0} + \beta_{1} \times Y5^{2} + \beta_{2} \times (1/Y_{6}) + \beta_{3} \times (1/Y_{7}) + \beta_{4} \times e^{Y8} + \beta_{5} \times (Y5^{2})^{2} + \beta_{6} \cdot Y5^{2} \cdot (1/Y_{6}) + \beta_{7} \cdot Y5^{2} \cdot (1/Y_{7}) + \beta_{8} \cdot Y5^{2} \cdot e^{Y8} + \beta_{9} \times (1/Y_{6})^{2} + \beta_{10} \cdot (1/Y_{6}) \cdot (1/Y_{7}) + \beta_{11} \cdot (1/Y_{6}) \cdot e^{Y8} + \beta_{12} \times (1/Y_{7})^{2} + \beta_{13} \cdot (1/Y_{7}) \cdot e^{Y8} + \beta_{14} \times (e^{Y8})^{2}$$
(30)

In the regression analysis, some of the predicted values for the fatality rate (FL) were negative. As a negative fatality rate is not physically meaningful in real-world scenarios, these negative predictions likely occurred due to the mathematical nature of the regression model and the limitations of the data available. In such cases the fatality rate was adjusted to zero. It is important to note that this adjustment, though necessary, may slightly bias the model's performance metrics.

The regression model was also trained and tested using a standard approach of splitting the dataset into two subsets: 80% for training and 20% for testing, and 70% for training and 30% for testing. The training set was used to fit the model, while the test set provided an independent evaluation to assess the model's generalization capability. This split ensures that the model is not overfitted to the training data and can be reliably applied to new, unseen data. The 80/20 split provided the best results.

From the result of the regression analysis, the RMSE (Root Mean Square Error) and R² (the coefficient of determination) for the low-severity and medium/high flood severity cases are presented in Table 5.

Data Split		Low severity (10		Medium and high Severity (22			
			datasets)		datasets)		
			RMSE	\mathbb{R}^2		RMSE	R^2
80/20	Train	Combination	1.0807e-	1	Combination	0.01941	0.9904
		No. 32	15		No. 17081		
	Test		0.0009	0.9949		0.0170	0.9955
70/30	Train	Combination	2.9939e-	1	Combination	3.3582e-	1
		No. 42	15		No. 12330	12	
	Test		0.0049	0.8547		0.04086	0.9714
100/0	Train	-	3.6572e-	1	_	0.0180	0.9929
			16				

Table 5. Regression analysis metrics

For comparison, the low, medium, and high-severity datasets were also looked at as a single set and combined in one regression analysis. The results are: RMSE (Train): 0.1108 and R² (Train): 0.6146; RMSE (Test): 0.1595 and R2 (Test): 0.5134. This indicates that combining the datasets does not lead to a good fit.

The iteration with the lowest R^2 value on the test dataset for low severity resulted in: RMSE (Train): 4.6e-14, R^2 (Train): 1; RMSE (Test): 82.9668, R^2 (Test): -4.4e+07. This occurred with the transformation combination: cubic, exponential, cubic, and reciprocal. While the iteration with the lowest R^2 value on the test dataset for medium and high severity cases resulted in

RMSE (Train): 0.002551, R² (Train): 0.9998; RMSE (Test): 11.7263, R² (Test): -2117.72. This occurred with the transformation combination: cubic, reciprocal, cubic and power. This extremely low R² and high RMSE may likely be amplified noise or poor generalization, causing the poor performance on unseen data.

The proposed equation for estimating F_L for low flood severity and medium and high severity cases is calculated in Eq. 31 & 32 respectively. Fatality rate (F_L) is defined as the value of loss of life (*LOL*) divided by population at risk (P_R). Thus *LOL* is calculated by Eq. 33 (Huang et al., 2017).

$$F_{L,m\&h} = -2.2533 + 11.3349 \times Y5^{2} - 0.1300 \times (1/Y_{f_{0}}) - 0.0378 \times (1/Y_{7_{0}}) + 2.6903 \times e^{Y8} + 12.7990 \times (Y5^{2})^{2} - 4.2117 \cdot Y5^{2} \cdot (1/Y_{f_{0}}) + 0.4114 \cdot Y5^{2} \cdot (1/Y_{7_{0}}) - 6.6235 \cdot Y5^{2} \cdot e^{Y8} + 0.0375 \times (1/Y_{f_{0}})^{2} + 0.0009 \cdot (1/Y_{f_{0}}) \cdot (1/Y_{7_{0}}) - 0.1023 \cdot (1/Y_{f_{0}}) \cdot e^{Y8} + 0.0001 \times (1/Y_{7_{0}})^{2} + 0.0098 \cdot (1/Y_{7_{0}}) \cdot e^{Y8} - 0.5143 \times (e^{Y8})^{2}$$

$$(32)$$

$$LOL = F_L \times P_R \tag{33}$$

The comparison between the actual and predicted *LOL* shows that the model results are close to the actual value of *LOL* except in a few cases (Figure 4, Table 6). A further comparison between the fatality rate and loss of life calculated with the proposed method and Graham's model (Graham, 1999) shows the proposed model performs better in predicting the *LOL* (Table 6). Graham's model was chosen as a comparison benchmark due to its straightforward formula and minimal data requirements. Unlike more complex models, Graham's model enables reproducibility and efficient comparison within the limits of available data and resources.



Figure 4: Comparison of actual LOL, predicted LOL and Graham's model.

Case	Subcase	Subcase No.	Actual LOL	Graham LOL	Predicted LOL
Johnstown South Fork 1 (1889)	Johnstown	1	2209	11310	2244
Canyon Lake Dam (1972)	Rapid City	2	238	301	238
	Mouth of Teton Canyon	3	1	2	1
	Wilford	4	5	21	5
Teton Dam (1976)	Teton Town	5	0	513	0
	Sugar city	6	0	0	0
	Rexburg	7	0	0	0
	Saunders	8	18	300	6
	Lorado	9	22	340	41
$\mathbf{D} = \{\mathbf{C} \mid \mathbf{C} \in [\mathbf{L}, \mathbf{D}] \in (1072)$	Lundale	10	52	10	52
Buffalo Creek Dam (1972)	Stowe	11	2	1	2
	Latrobe	12	8	2	8
	Accoville	13	0	14	0
St. Francis, CA (1928)	Edison Camp	14	84	114	84
	Dam to Sanchez Drive	15	0	0	30
Baldwin Hills, CA (1965)	Sanchez Drive onwards inc. Green Village	Johnstown 1 2209 111 Rapid City 2 238 30 Mouth of Teton 3 1 22 Canyon 3 1 22 Wilford 4 5 2 Teton Town 5 0 5 Sugar city 6 0 0 Rexburg 7 0 0 Saunders 8 18 30 Lorado 9 22 34 Lundale 10 52 1 Stowe 11 2 3 Latrobe 12 8 3 Accoville 13 0 1 Edison Camp 14 84 1 Dam to Sanchez 15 0 0 Sanchez Drive 16 5 0 onwards inc. Green 16 5 0 Village 17 39 1 College 17 39 1 Williamsburg 18 138 1	0	0	
Kelly Barnes Dam (1977)	Toccoa Falls Bible College	17	39	14	43
Mill River (1874)	Williamsburg	18	138	124	143
Austin Dam (Bayless Dam) I (1911)	Austin, PA	19	78	126	75
Laurel Run Dam, PV (1977)	Johnstown	20	40	21	40

Table 6. Comparison of the proposed equation and the actual LOL and F_{L} .

Timberlake Dam, VA (1995)	Lynchburg	21	2	1	2
Little Deer Creek Dam, UT (1963)		22	1	38	1
Mohegan Park Dam (Spaulding Pond Dam) (1963)	Norwich	23	6	4	6
Meadow Pond / Bergeron Pond Dam (1996)	Alton, NH	24	1	0	1
Lee Lake (1968)	Lee, MA	25	2	1	2
Lawn Lake Dam (1982)	Roaring River	26	1	19	2
Cascade Lake Dam (1982)	Aspen Glen Campground	27	2	4	0
Lawn Lake & Cascade Valley (11982)	Fall River Road / Estes Park	28	0	40	7
Bear Wallow Dam (1976)	Ashville, NC	29	4	1	4
Walnut Grove Dam (1890)	Another New Dam	30	85	95	85
Swift Dam No. 2, MT (1964)	Birch Creek	31	19	190	17
Castlewood Canyon Dam II (1933)	Denver	32	2	-	0
	-				

3.5 Conclusion

The presented LOL estimation model, developed through a comprehensive review and improvement of existing models, holds promise for improving accuracy of loss of life (LOL) predictions in dam failure scenarios. The paper successfully adapts and applies a model originally developed for China to North American dam failure cases, demonstrating potential cross-regional applicability. Limitations include data constraints which were overcome by simulation and estimation. In some cases, the Population at Risk was not reported, and the missing data were estimated based on watershed area and census data and might be a source of error. Future work should focus on obtaining additional historical data for dam failures in North America and other regions globally to improve model accuracy and develop models for other regions. The incorporation of polynomial terms and combination functions ensures that the model accounts for both simple and complex relationships between variables, leading to a robust regression model that captures key patterns in the data. The model's high R² values, particularly in low-severity cases, suggest strong performance; however, the possibility of overfitting, especially in cases where R² approaches 1, should be considered due to the complexity of the non-linear regression used. The proposed method, with its emphasis on diverse influencing factors, stands as a valuable tool for disaster preparedness and response, contributing to dam safety and community well-being downstream of dams.

4 GIS FRAMEWORK TO ASSESS IMPACTS OF DAM FAILURE

Samuel Ovu, MASc Candidate, School of Engineering, University of Northern British Columbia, Canada Mauricio Dziedzic, P. Eng., Chair, School of Engineering, University of Northern British Columbia, Canada

This chapter is an expanded version of the article "GIS Framework to Assess Impacts of Dam Failure" presented in the Canadian Dam Association 2024 Annual Conference and will be submitted to a peer reviewed journal such as Elseviers's International Journal of Disaster Risk Reduction.

ABSTRACT

Dam safety is of critical concern in water resources management. While traditional approaches to dam safety have been effective for decades, climate change and increasingly extreme weather events necessitate a continuous improvement in risk assessment and emergency planning capabilities. Geographic Information Systems (GIS) have become a valuable tool in dam failure analysis, allowing for modelling scenarios and providing valuable information for emergency planning and risk mitigation. This paper introduces a segmented GIS model being developed as part of a comprehensive framework to assess the impacts of dam breaks. The framework integrates hydraulic modelling, loss of life (LOL) modelling and GIS technologies. It highlights a QGIS plugin that facilitates flood risk analysis and loss of life estimation by integrating HEC-RAS results into QGIS and enables the visualization of flood extents, identification of vulnerable assets, and estimation of potential fatalities. Mortality rate is calculated considering regional dam failure data, population dynamics, time of failure and weather patterns. This approach will contribute to enhance dam safety management and advance the field of dam safety by providing a comprehensive understanding of the potential impacts of dam breaks.

4.1 Introduction

Dams serve as vital components of water resource management systems, facilitating irrigation, flood control, hydropower generation, and municipal water supply. For decades, dam safety has been managed effectively through established practices. However, factors such as aging infrastructure, evolving climate patterns, and population growth in downstream areas lead to the need for re-evaluating dam safety protocols (Li et al., 2019). While traditional methods have been effective, these new challenges require their supplementation with advanced tools and methodologies to enhance existing dam safety practices. These tools offer more comprehensive risk assessment capabilities and improved emergency planning, allowing to proactively address dam safety in the face of contemporary challenges.

Geographic Information Systems (GIS), coupled with hydraulic and hydrologic modelling software, remote sensing technologies, and advanced computational methods, offer unprecedented capabilities for analysing dam safety risks in a comprehensive manner. With GIS being an indispensable tool for understanding and managing complex spatial data (Ghent, 2013), its integration into various fields has revolutionized the way we approach problem-solving, particularly in the field of environmental risk assessment and disaster management. Understanding the role of GIS in dam break assessment is recognizing its capacity to integrate diverse datasets and analytical tools into a cohesive platform. By harnessing spatial data on dam infrastructure, terrain characteristics, hydrological patterns, and population demographics, GIS empowers stakeholders to model and simulate various scenarios, thereby anticipating potential risks and formulating effective mitigation strategies (Katwal, 2018; Mancusi et al., 2015). Through spatial analysis and visualization techniques, GIS facilitates the identification of vulnerable populations, critical infrastructure, and ecologically sensitive areas that may be affected in the event of a dam failure.

Recently, there has been notable progress in the assessment of dam failure hazards, largely credited to the integration of GIS with hydraulic modelling. This integration allows for the seamless transfer of data generated from hydraulic models to GIS platforms, enabling the creation of inundation maps and facilitating further analysis (Abdalla 2009; Pandya and Jitaji 2013). Numerous studies have examined the significance of integrated GIS in managing dam failure hazards: Aboelata et al. (2003) discussed the concept of integrated GIS, proposing a modular GIS model for estimating potential loss of life resulting from natural and dam-failure floods. Similarly, Abdalla (2009) demonstrated the utility of WebGIS through a case study in the Don Valley watershed, Toronto, Canada, highlighting its effectiveness by simulating

different scenarios in delineating different water surface elevations for the assessment of possible impact on critical infrastructure and land use classes.

The quality of spatial data used in modelling is also of importance, as it forms the basis of the geometric file used for simulation and analyses (Solaimani 2009). Advanced spatial data collection technologies, such as LiDAR and high-resolution satellite imagery, have significantly enhanced the accuracy and detail of three-dimensional (3D) spatial information (Doyle et al. 1998; Schultz 2009). Alonso (2015) investigated the impact of LiDAR data on the classification of multispectral imagery to show its potential for extracting buildings and other objects from medium-resolution satellite imagery. The classification result showed a more realistic representation of geographic features compared to those obtained solely from multispectral satellite imagery.

Various software and modelling tools have been employed to predict flooding and manage its consequences. River Analysis System (HEC-RAS) and Hydrologic Modelling System (HEC-HMS) which were developed by the U.S. Army Engineer Hydrologic Engineering Center (HEC), can simulate the water surface profile of rivers and open channels, generate flood hazard maps, and simulate the complete hydrologic processes of watershed systems (Ogras and Onen 2020; Ongdas et al. 2020; Almasalmeh et al. 2022; HEC 2024). Quantum Geographic Information system (QGIS), an open-source GIS application, has also been extensively utilized for flood risk assessment, natural hazard mapping, and detection of flood hazards (Mancusi et al., 2015; Sansare & Mhaske, 2020; Soni & Prasad, 2021). Integration of QGIS with HEC-RAS enabled the mapping of flood-risk buildings resulting from a levee breach causing flooding that affected millions of people in Nepal and India in 2008 (Katwal, 2018). Derdous et al. (2015) studied an approach based on the integration of hydraulic modelling and GIS to assess the risks resulting from a potential failure of a concrete dam in Algeria's North East. Albano et al. (2019) used a similar approach to demonstrate the effectiveness of a GIS-based method for delineating dam-break flood-prone areas, particularly in data-scarce environments and transboundary regions.

The studies mentioned thus far have effectively showcased the diverse functionalities enabled by GIS integration. Therefore, the aim of this paper is to introduce a segmented framework under development for dam break assessment. This framework integrates hydraulic modelling, LOL modelling, and GIS technologies. Also, the components of the segmented GIS framework illustrate how each facet contributes to a comprehensive understanding of dam failure risks and their consequences. In addition, we highlight the practical implications of integrating GIS into dam safety management practices, emphasizing the potential for enhanced preparedness and response in the face of evolving threats.

4.2 EXAMPLE OF APPLICATION

Buffalo Creek dam located N 37°47′50″ W 81°39′50″ in Logan County, West Virginia, USA. The Buffalo Creek flood is considered as one of the worst dam break-related disasters in American history (NOAA, 1972a; Webb Jeffrey, 2021). On the 26th of February 1972, an impoundment dam for coal-mine waste broke, releasing over 130 million gallons (app. 492 million liters) of dark floodwater. The flood travelled over 20 km downstream, killing 125 people and leaving over 3000 homeless. Information about this area is presented in Table 7. This site's data, including a Digital Elevation Model (DEM) with resolution of 1m, was used for hydrologic and hydrodynamic simulations that allowed obtaining the flood severity parameter to inform the loss of life model in the GIS framework presented in the following sections.

	Height of Dam, m	Distance to Dam site, km	Population
Saunders		1.5	429
Lorado		6.78	618
Lundale	13.41	9.57	997
Stowe		9.66	128
Latrobe		13.2	258
Accoville		19.32	2070

Table 7. information on the areas downstream of Buffalo Creek

4.3 METHODOLOGY

4.3.1 Data collection and preparation

Depending on the availability of data, hydrologic simulation might not be required if the inflow hydrograph is available. HEC-HMS is used for the hydrologic simulation to obtain hydrograph results. Data required for hydrologic modelling in HEC-HMS include a Digital Elevation Model (DEM), dam and spillway parameters, Land Use Land Cover data (LULC), soil map (soil type data), climate data (precipitation), and dam breach parameters. The data sources can vary depending on factors such as availability, scale, and specific project requirements. Here are some common sources for each type of data:

• Elevation Models:

- Provincial and national mapping agencies (e.g., Government of BC Geographic Data Services, United States Geological Survey (USGS), Ordnance Survey in the UK)
- Satellite or aerial imagery (e.g., NASA's Shuttle Radar Topography Mission, commercial satellite providers)
- LiDAR surveys
- Open data repositories (e.g., NASA Earthdata, USGS EarthExplorer)
- Land Use Land Cover (LULC) Data:
 - National or regional land use/land cover datasets (e.g., National Land Cover Database in the United States, CORINE Land Cover in Europe)
 - Remote sensing imagery (e.g., Landsat, Sentinel)
 - Academic research institutions
- Soil Maps (Soil Type Data):
 - National or regional soil surveys (e.g., NRCS Soil Survey in the United States, European Soil Data Centre)
 - Government agencies specializing in soil science
 - Soil research institutions and universities
 - Remote sensing techniques combined with ground-truthing
- Climate Data (Precipitation):
 - Meteorological stations operated by national meteorological agencies
 - Global climate datasets (e.g., WorldClim)
 - Remote sensing data (e.g., satellite-derived precipitation estimates)
 - Climate reanalysis datasets (e.g., ERA5, NCEP/NCAR Reanalysis)
- Dam and Spillway Parameters:
 - o Design documents and engineering drawings provided by dam owners/operators
 - Regulatory agencies responsible for dam safety
 - Dam safety databases maintained by government agencies
 - Site surveys conducted by engineering firms or consultants
- Dam Breach Parameters:
 - Engineering studies and reports on dam safety and risk assessments
 - Historical data on dam failures and breach events
 - Hydraulic modelling studies specific to dam breach scenarios
 - Empirical equations and guidelines for estimating dam breach parameters

After the data collection and preparation, a hydrological simulation is made using HEC-HMS to obtain the flow hydrograph. This hydrograph result is then used in the hydraulic simulation. Finally, flood water depth, velocity, and inundation/flood-affected area were obtained from the hydraulic simulation, and these are used to make flood inundation maps. Figure 5 shows a summary of the procedure followed to achieve the objective of this study. Collected data may need to be modified or used to create additional information to be used for modelling.



Figure 5: Overview of the methodology.

4.3.2 HEC-RAS (hydrodynamic modelling)

Dam break simulations were conducted using HEC-RAS, a widely utilized hydraulic modelling tool for simulating water surface profiles in rivers, streams, and other reservoir systems. Notably, 2D hydrodynamic modelling is widely used for flood inundation modelling (Katwal, 2018). Originally introduced in 1995, it has undergone several iterations since its inception. Version 6.5 of HEC-RAS was used in this study to model flood extents. Figure 6 shows a schematic flowchart of the hydraulic modelling using HEC-RAS.



Figure 6: HEC-RAS simulation flowchart (from Phyo et al. 2023)

4.3.3 LOL Estimation Calculation

The LOL estimation model and the validation of the model's accuracy has been discussed in the previous chapter.

4.3.4 Tools

The major tool used in this study for spatial analysis and visualization is QGIS. It is an opensource geographic information system that provides a comprehensive environment for spatial data analysis, visualization, and modelling. Integration of simulation outputs in QGIS facilitated the spatial representation and analysis of dam break consequences, including the visualization of flood extents, identification of affected areas, and assessment of impacts. One main advantage of QGIS is its ease in the development of new plugins. In this study, a plugin is being developed and the interface is created with the aid of the Python programming language. The plugin is first created in QGIS, then the user interface is designed with QT-Designer, then functionality is added to the plugin through Python scripts on PyCharm. Loss of life (LOL) model calculation is also automated in QGIS with Python codes.

4.4 RESULTS AND DISCUSSION

The results of the HEC-RAS simulation are the floodwater discharge, maximum water surface elevation, floodwater volume, velocity, water depth and water surface elevation profile along Buffalo Creek. The output map for the maximum flooding depth along the downstream reach for the time of breach is shown in Figure 7 (the different subcase/ settlements are marked by the black lines across the flood path). Figure 8 shows downstream settlements before and after flooding of the same section, respectively. The simulation results are exported in raster format. The exported rasters are polygonised in QGIS and finally overlayed with other data layers.



Figure 7: Inundation map with depth of Buffalo Creek flood



Figure 8: Segment before (a) and after (b) flooding along Buffalo Creek channel downstream.

4.4.1 Development of Integrated Plugin for QGIS

The plugin allows users to import multiple raster layers (Figure 9a). To determine the inundated area, the plugin uses QGIS's basic statistical calculation tools to measure the area of polygons. Identifying affected assets involves intersecting data, such as buildings, crops, and infrastructure, with the inundation layer. The resulting inundation map highlights zones and assets impacted during the maximum flood discharge.

The plugin employs a previously developed loss of life estimation model (Ovu and Dziedzic, 2024) (as shown in Figure 9b). By considering various factors such as population density, building occupancy, flood intensity, and time of occurrence, the model provides an estimate of the potential loss of life.

amework LOL		×	Q Framework LOL			
mports and database Los	s of life estimation Result Help		Imports and database Loss of	flife estimation Result H	elp	
ow velocity raster layer	ome \home \ovu\Downloads \Velocity (Max) Logan County. Terrain. Terrain. merged dem. tif 🚳					
ater depth raster layer	shome\home\ovu\Downloads\Depth (Max) logan county.Terrain.Terrain.merged dem.tif 🚳		HD (m) Height of the dam	SF (m^2/s) Severity of the	flood People's und	derstanding of the break (UB)
vation layer			DD ((km) Distance from the D	D PR Population at R	isk O Clear	/ Precise
pulation distribution			Building vulnerability, VB	Breach time, TB	Warning time, Tw	Evacuation Condition, EC
ilding layer			O Adobe	Midnight (00:00 - 08:00)	0-15	O Bad
nd cover layer			O Wood	Pautime (00:00 - 20:00)	(i) 15-30	O Middle
ldlife habitat data			O Masonry (Brick, Stone)	 Night (20:00 - 24:00) 	0 30-45 0 45-60	◯ Good
ansport Infrastructure layer			Concrete	() Hight (20:00 - 24:00)	() >60	
	Add all layers to map		Break mode, MB Weather at Breach, WB			
			Overtopping Poor quality (leakage, jar internal erosion etc.)	ns of dam, tunnel, spillway,	C Level I (storm, bli Level II (heavy ra	zzard, typhoon, fog) in, heavy snow, gale)
			Mismanagement (Overag	e storage, poor maintenance ce or management etc.)	Level III (moderat	te rain, moderate snow
			O Others		C Level V (sunny or	doudy day)
oject Title						
	Create Database			Run Loss of	life model	

Figure 9: GIS framework plugin interface for (a) importing layers and (b) loss of life estimation.

The outcomes generated by the plugin will vary based on the data layers imported (Figure 10). The plugin's flexibility allows it to accommodate a wide range of data inputs. Future work on expanding the plugin functionalities will include allowing the user to select different hydrodynamic modelling software and estimate other environmental, social, and economic consequences resulting from dam failure, such as the social disruptions and impacts on vegetation/crops, bridges, roads, human and environmental health, water availability and quality, and animals/wildlife within the inundation zone.



Figure 10: GIS framework plugin interface sample result.

4.4.2 Conclusion

This study introduces a segmented framework for dam break assessment that integrates hydrodynamic modelling, LOL estimation, social, economic, and environmental impacts, and GIS technologies. The QGIS plugin developed as part of this study automates several critical calculations, including the inundated area and loss of life estimations, thereby enhancing the efficiency and accuracy of flood risk analysis. While still in the initial stages, the plugin shows significant promise as a tool for dam safety management.

Future work will focus on refining the plugin, improving user interface design, and validating the model's accuracy with several case studies, seeking to ensure that this plugin becomes a valuable asset for dam safety professionals and emergency planners.

5 CONCLUSION

This study successfully developed a GIS-based framework for assessing life loss resulting from dam failures. By combining GIS and hydrodynamic modelling, the framework visualizes the associated risks to downstream populations and infrastructure. The proposed model, based on empirical data and refined regression analysis, demonstrated higher accuracy and adaptability in estimating loss of life across various dam failure scenarios. Ultimately, this framework provides a valuable tool for enhancing dam safety, improving emergency preparedness, and informing hazard mitigation plans. It offers a robust, adaptable approach for evaluating dam failure risks, promoting better-informed decision-making, and supporting efforts to minimize disaster impacts on communities and the environment.

As this project is still in its early development, potential future work could involve exploring advanced visualization techniques, such as Virtual Reality (VR), to enhance assessment capabilities. By allowing users to virtually experience the potential effects of a dam break, VR can offer an immersive understanding of flood extents and impact severity. Expanding the framework's applicability to include more variables that impact flood severity, such as climate change projections and evolving urban development patterns in downstream areas. Integrating this framework with real-time dam monitoring systems could transform it into an automated risk assessment tool capable of issuing live updates and warnings to support rapid, informed responses.

REFERENCES

- Abdalla, R. (2009). Distributed GIS Approach for Flood Risk Assessment. *International Journal on Advances in Security*, 2(2), 182–189.
- Aboelata, M., & Bowles, D. S. (2008). LIFESim: A tool for estimating and reducing life-loss resulting from dam and levee failures. Association of Dam Safety Officials - Dam Safety 2008, January 2008.
- Aboelata, M., Bowles, D. S., & Mcclelland, D. M. (2003). A Model for Estimating Dam Failure Life Loss. Proceedings of the Australian Committee on Large Dams Risk Workshop, Launceston, Tasmania, Australia., May.
- Albano, R., Mancusi, L., Adamowski, J., Cantisani, A., & Sole, A. (2019). A GIS tool for mapping dam-break flood hazards in Italy. *ISPRS International Journal of Geo-Information*, 8(6). https://doi.org/10.3390/ijgi8060250
- Albu, L.-M., Enea, A., Iosub, M., & Breabăn, I.-G. (2020). Dam Breach Size Comparison for Flood Simulations. A HEC-RAS Based, GIS Approach for Drăcşani Lake, Sitna River, Romania. *Water*, 12(4), 1090. https://doi.org/10.3390/w12041090
- Almasalmeh, O., Mourad, K. A., & Eizeldin, M. (2022). Simulating flash floods using remote sensing and GIS-based KW-GIUH hydrological model. *Arabian Journal of Geosciences*, 15(19). https://doi.org/10.1007/S12517-022-10852-6
- Alonso, M. C., & Malpica, J. A. (2015). Satellite imagery classification with LIDAR data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, XXXVIII, 730–735.
- ASDSO. (n.d.-a). Buffalo Creek Dam (West Virginia, 1972). Retrieved March 8, 2024, from https://damfailures.org/case-study/buffalo-creek-dam-west-virginia-1972/
- ASDSO. (n.d.-b). Canyon Lake Dam (South Dakota, 1972). Association of State Dam Safety Officials. Retrieved March 8, 2024, from https://damfailures.org/case-study/canyon-lake-dam-south-dakota-1972/
- ASDSO. (n.d.-c). Teton Dam (Idaho, 1976). Retrieved March 8, 2024, from https://damfailures.org/case-study/teton-dam-idaho-1976/
- Assaf, H. (2002). A Virtual Reality Approach to Public Protection and Emergency Preparedness Planning in Dam Safety Analysis System Dynamics Simulation for Assessment of Hydropower Dam Safety View project. *Canadian Dam Association Conference Proceedings 2002*. https://www.researchgate.net/publication/258513135
- Brown, C. A., & Graham, W. J. (1988). Assessing the Threat to Life from Dam Failure. *JAWRA Journal of the American Water Resources Association*, 24(6), 1303–1309. https://doi.org/10.1111/J.1752-1688.1988.TB03051.X
- Canadian Climate Institute. (2024, September). Climate Change and Floods. https://climateinstitute.ca/wp-content/uploads/2024/09/Fact-sheet_-Floods_CanadianClimateInstitute.pdf
- DeKay, M. L., & McClelland, G. H. (1993). Predicting Loss of Life in Cases of Dam Failure and Flash Flood. *Risk Analysis*, 13(2), 193–205. https://doi.org/10.1111/j.1539-6924.1993.tb01069.x
- Derdous, O., Djemili, L., Bouchehed, H., & Tachi, S. E. (2015). A GIS based approach for the prediction of the dam break flood hazard - A case study of Zardezas reservoir "skikda, Algeria." *Journal of Water and Land Development*, 27(1), 15–20. https://doi.org/10.1515/jwld-2015-0020
- Doyle, S., Dodge, M., & Smith, A. (1998). The potential of Web-based mapping and virtual reality technologies for modelling urban environments. *Computers, Environment and Urban Systems*, 22(2), 137–155. https://doi.org/10.1016/S0198-9715(98)00014-3
- Dutta, D., Herath, S., & Musiake, K. (2003). A mathematical model for flood loss estimation.

Journal of Hydrology, 277(1–2), 24–49. https://doi.org/10.1016/S0022-1694(03)00084-2 Ehsan, S. (2009). Evaluation of Life Safety Risks Related to Severe Flooding. In *self-published by Instituts für Wasserbau der Universität Stuttgart*.

Eleutério, J. (2013). Flood risk analysis: impact of uncertainty in hazard modelling and vulnerability assessments on damage estimations *To cite this version: HAL Id: tel-*00821011 impact of uncertainty in hazard modelling and vulnerability assessments.

Environment Alberta. (2016). Canadian Dam Association (CDA) Consequence Classification Ratings for Dams. https://open.alberta.ca/dataset/e598d71f-9baa-4f33-98d1-2417f9bf7d93/resource/08db72bd-6fef-48d4-8c62-72c33c44d9a3/download/cdaclassificationratingsdams-apr2016.pdf

Ge, W., Jiao, Y., Sun, H., Li, Z., Zhang, H., Zheng, Y., Guo, X., Zhang, Z., & van Gelder, P. H. A. J. M. (2019). A method for fast evaluation of potential consequences of dam breach. In *Water (Switzerland)* (Vol. 11, Issue 11). https://doi.org/10.3390/w11112224

Ge, W., Jiao, Y., Wu, M., Li, Z., Wang, T., Li, W., Zhang, Y., Gao, W., & van Gelder, P. (2022). Estimating loss of life caused by dam breaches based on the simulation of floods routing and evacuation potential of population at risk. *Journal of Hydrology*, 612, 128059. https://doi.org/10.1016/J.JHYDROL.2022.128059

Ge, W., Wang, X., Li, Z., Zhang, H., Guo, X., Wang, T., Gao, W., Lin, C., & Gelder, P. van. (2020). Interval Analysis of the Loss of Life Caused by Dam Failure. *Journal of Water Resources Planning and Management*, 147(1), 04020098. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001311

Ghent, E. O. (2013). Application of Remote Sensing and Geographical Information Systems in Flood Management: A Review. *Research Journal of Applied Sciences, Engineering and Technology*, 6(10), 1884–1894. https://doi.org/10.19026/rjaset.6.3920

Graham, W. J. (1999). Loss of life caused by dam failure. http://www.usbr.gov/ssle/dam_safety/risk/Estimating life loss.pdf

Graham, W. J. (2008). The Teton Dam Failure- An Effective Warning and Evacuation. https://damfailures.org/wp-content/uploads/2015/07/075_The-Teton-Dam-Failure.pdf

Haq, M., Akhtar, M., Muhammad, S., Paras, S., & Rahmatullah, J. (2012). Techniques of Remote Sensing and GIS for flood monitoring and damage assessment: A case study of Sindh province, Pakistan. *The Egyptian Journal of Remote Sensing and Space Science*, 15(2), 135–141. https://doi.org/10.1016/J.EJRS.2012.07.002

HEC. (2024). HEC-RAS Features. USACE HEC. https://www.hec.usace.army.mil/software/hec-ras/features.aspx

Huang, D., Yu, Z., Li, Y., Han, D., Zhao, L., & Chu, Q. (2017). Calculation method and application of loss of life caused by dam break in China. *Natural Hazards*, 85(1), 39–57. https://doi.org/10.1007/s11069-016-2557-9

ICOLD. (n.d.). ICOLD CIGB > General Synthesis. Retrieved February 17, 2024, from https://www.icold-cigb.org/GB/world_register/general_synthesis.asp

ICOLD. (2011). Constitution. https://www.icoldcigb.org/userfiles/files/cigb/institutional_files/constitution2011.pdf

Jay Hilary Kelley, Dan Kealy, Charles D. Hylton, J., Elizabeth V. Hallanan, John Ashcraft, Julian F. Murrin, William E. Davies, Robert B. Erwin, & Ira S. Latimer, J. (1973). The Buffalo Creek Flood and Disaster: Official Report from the Governor's Ad Hoc Commission of Inquiry. http://129.71.204.160/history/disasters/buffcreekgovreport.html

Johnstone, W. M., & Lence, B. J. (2012). Use of Flood, Loss, and Evacuation Models to Assess Exposure and Improve a Community Tsunami Response Plan: Vancouver Island. *Natural Hazards Review*, 13(2), 162–171. https://doi.org/10.1061/(asce)nh.1527-6996.0000056

Johnstone W M, Sakamoto D, Assaf H, & Bourban S. (2005). Architecture, modelling

framework and validation of BC Hydro's virtual reality life safety model. *International Symposium on Stochastic Hydraulics*.

- Jonkman, S. N., Godfroy, M., Sebastian, A., & Kolen, B. (2018). Brief communication: Loss of life due to Hurricane Harvey. *Natural Hazards and Earth System Sciences*, 18(4), 1073–1078. https://doi.org/10.5194/NHESS-18-1073-2018
- Jonkman, S. N., Vrijling, J. K., & Vrouwenvelder, A. C. W. M. (2008). Methods for the estimation of loss of life due to floods: A literature review and a proposal for a new method. *Natural Hazards*, *46*(3), 353–389. https://doi.org/10.1007/s11069-008-9227-5
- Joseph A. Strahl, Herbert Lieb, & Lawrence Longsdorf. (1972). Black Hills Flood of June 9, 1972. https://www.weather.gov/media/publications/assessments/NDSR 72-1.pdf
- Karim Solaimani. (2009). Flood forecasting based on geographical information system | Request PDF. *African Journal of Agricultural Research*. https://www.researchgate.net/publication/228659164_Flood_forecasting_based_on_geog raphical_information_system
- Katwal, D. (2018). Mapping Flood Risk Buildings: Using QGIS and HEC-RAS. http://www.theseus.fi/handle/10024/140303
- Larimer, O. J., & Department of the Interior, U. S. G. S. (1973). Flood of June 9-10, 1972, At Rapid City, South Dakota.pdf. https://pubs.usgs.gov/ha/511/plate-1.pdf
- Lee, J.-S. (2003). Uncertainties in the predicted number of life loss due to the dam breach floods. *KSCE Journal of Civil Engineering*, 7(1), 81–91. https://doi.org/10.1007/bf02841991
- Li, W., Li, Z., Ge, W., & Wu, S. (2019). Risk evaluation model of life loss caused by dambreak flood and its application. *Water (Switzerland)*, 11(7). https://doi.org/10.3390/w11071359
- Logan County Genealogical Society. (1972). Ancestree, Tragedy on Buffalo Creek. https://loganwv.us/wp-content/uploads/2013/09/Tragedy-on-Bullalo-Creek-February-26-1972-Volume-34-Issue-2.pdf
- Lumbroso, D., Davison, M., Body, R., & Petkovšek, G. (2021). Modelling the Brumadinho tailings dam failure, the subsequent loss of life and how it could have been reduced. *Natural Hazards and Earth System Sciences*, *21*(1), 21–37. https://doi.org/10.5194/NHESS-21-21-2021
- Lumbroso, D. M., Sakamoto, D., Johnstone, W., Tagg, A., & Lence, B. L. (2011). The Development of a Life Safety Model to Estimate the Risk Posed to People by Dam Failures and Floods.

https://eprints.hrwallingford.com/836/1/HRPP473_TheDevelopmentOfALifeSafetyMod el_2011.pdf

- Mahmoud, A. A., Wang, J. T., & Jin, F. (2020). An improved method for estimating life losses from dam failure in China. *Stochastic Environmental Research and Risk Assessment*, 34(8), 1263–1279. https://doi.org/10.1007/s00477-020-01820-1
- Mancusi, L., Albano, R., & Sole, A. (2015). FloodRisk: a QGIS plugin for flood consequences estimation. *Geomatics Workbooks*, *February*, 483–496. https://doi.org/10.13140/RG.2.1.4215.7846
- Merz, B., Kreibich, H., Thieken, A., & Schmidtke, R. (2004). Estimation uncertainty of direct monetary flood damage to buildings. *Natural Hazards and Earth System Sciences*, 4(1), 153–163. https://doi.org/10.5194/NHESS-4-153-2004
- Meservy, O. K. (1968). Voices from the Past: The Community of Wilford, Fremont County, Idaho. https://content.byui.edu/file/04b51b10-86b9-44ac-8fef-0a557c426692/1/mssi50_019_OliverKingsberryMeservy.pdf
- Meyer, V., Becker, N., Markantonis, V., Schwarze, R., Van Den Bergh, J. C. J. M., Bouwer, L. M., Bubeck, P., Ciavola, P., Genovese, E., Green, C., Hallegatte, S., Kreibich, H.,

Lequeux, Q., Logar, I., Papyrakis, E., Pfurtscheller, C., Poussin, J., Przyluski, V., Thieken, A. H., & Viavattene, C. (2013). Review article: Assessing the costs of natural hazards-state of the art and knowledge gaps. *Natural Hazards and Earth System Science*, *13*(5), 1351–1373. https://doi.org/10.5194/nhess-13-1351-2013

- Middelmann-Fernandes, M. H. (2010). Flood damage estimation beyond stage-damage functions: An Australian example. *Journal of Flood Risk Management*, *3*(1), 88–96. https://doi.org/10.1111/J.1753-318X.2009.01058.X
- Mohammadi, S. A., Nazariha, M., & Mehrdadi, N. (2014). Flood Damage Estimate (Quantity), Using HEC-FDA Model. Case Study: The Neka River. *Procedia Engineering*, 70, 1173–1182. https://doi.org/10.1016/J.PROENG.2014.02.130
- Natho, S., & Thieken, A. H. (2018). Implementation and adaptation of a macro-scale method to assess and monitor direct economic losses caused by natural hazards. *International Journal of Disaster Risk Reduction*, 28, 191–205. https://doi.org/10.1016/J.IJDRR.2018.03.008
- National Weather Service. (n.d.). The Black Hills Flood of 1972. US Department of Commerce, NOAA, National Weather Service. Retrieved March 8, 2024, from https://www.weather.gov/unr/1972-06-09
- NOAA. (1972a). Report to administrator NOAA on Buffalo Creek (West Virginia) disaster.
- NOAA. (1972b). The Black Hills Flood of June 9, 1972.
 - https://www.weather.gov/media/publications/assessments/NDSR_72-1.pdf
- Notaro, V., De Marchis, M., Fontanazza, C. M., La Loggia, G., Puleo, V., & Freni, G. (2014). The Effect of Damage Functions on Urban Flood Damage Appraisal. *Procedia Engineering*, 70, 1251–1260. https://doi.org/10.1016/J.PROENG.2014.02.138
- Ogras, S., & Onen, F. (2020). Flood Analysis with HEC-RAS: A Case Study of Tigris River. Advances in Civil Engineering, 2020. https://doi.org/10.1155/2020/6131982
- Oliveri, E., & Santoro, M. (2000). Estimation of urban structural flood damages: the case study of Palermo. *Urban Water*, 2(3), 223–234. https://doi.org/10.1016/S1462-0758(00)00062-5
- Ongdas, N., Akiyanova, F., Karakulov, Y., Muratbayeva, A., & Zinabdin, N. (2020). Application of hec-ras (2d) for flood hazard maps generation for yesil (ishim) river in kazakhstan. *Water (Switzerland)*, 12(10), 1–20. https://doi.org/10.3390/W12102672
- Pandya, P. H., & Dixitsinh Jitaji, T. (2013). A Brief Review of Method Available for Dam Break Analysis. *PARIPEX-Indian Journal of Research*, *2*(4), 117–118.
- Peng, M., & Zhang, L. (2012). Analysis of human risks due to dam-break floods-part 1: A new model based on Bayesian networks. *Natural Hazards*, 64(1), 903–933. https://doi.org/10.1007/s11069-012-0275-5
- Phyo, A. P., Yabar, H., & Richards, D. (2023). Managing dam breach and flood inundation by HEC-RAS modeling and GIS mapping for disaster risk management. *Case Studies in Chemical and Environmental Engineering*, 8(August), 100487. https://doi.org/10.1016/j.cscee.2023.100487
- Public Safety Canada. (2017). An Emergency Management Framework for Canada. May, 26.
- Reiter. (2001). Development of Rescue Actions Based on. Rescdam, June 1999.
- Rogers, J. D. (1928). The 1928 St. Francis Dam Failure and Its Impact on American Civil Engineering. https://web.mst.edu/~rogersda/st_francis_dam/St-Francis-Dam-for-ASCE-Press.pdf
- Sanders, B. F. (2007). Evaluation of on-line DEMs for flood inundation modeling. Advances in Water Resources, 30(8), 1831–1843. https://doi.org/10.1016/j.advwatres.2007.02.005
- Sansare, D. A., & Mhaske, S. Y. (2020). Natural hazard assessment and mapping using remote sensing and QGIS tools for Mumbai city, India. *Natural Hazards*, 100(3), 1117–1136. https://doi.org/10.1007/S11069-019-03852-5

- Schultz, G. (2009). Use of remote sensing data in a GIS environment for water resources management. *IAHS-AISH Publication (1997) 242 3-15*.
- Soni, S., & Prasad, A. D. (2021). Detection of flood hazard using Qgis. 77, 899–905. https://doi.org/10.1007/978-981-15-5195-6_65
- Spero, H., Calhoun, D., & Schubert, M. (2022). Simulating the 1976 Teton Dam Failure using Geoclaw and HEC-RAS and comparing with Historical Observations. http://arxiv.org/abs/2206.00766
- Tang, J. C. S., Vongvisessomjai, S., & Sahasakmontri, K. (1992). Estimation of flood damage cost for Bangkok. *Water Resources Management*, 6(1), 47–56. https://doi.org/10.1007/BF00872187/METRICS
- The United Nations International Strategy for Disaster Reduction, U. (2016). Concept note on Methodology to Estimate Direct Economic Losses from Hazardous Events to Measure the Achievement of Target C of the Sendai Framework. https://www.preventionweb.net/files/47137 indicatorsforglobaltargetcconceptno.pdf

Tineke de Jonge, Matthijs Kok, & Marten Hogeweg. (1996). Modelling floods and damage assessment using GIS.

https://www.researchgate.net/publication/265919679_Modelling_floods_and_damage_as sessment using GIS

- U. S. Census Bureau. (n.d.). Number of Inhabitants, Idaho. Retrieved March 8, 2024, from https://www2.census.gov/prod2/decennial/documents/37779058v2p12ch2.pdf
- U.S. Department of the interior Bureau of Reclamation. (2015). RCEM Reclamation Consequence Estimating Methodology.
 - https://www.usbr.gov/ssle/damsafety/documents/RCEM-CaseHistories2015.pdf
- US Army Corps of Engineers. (2015, March 27). *Historical Vignette: The Rapid City Flood, June 1972*. https://www.nwo.usace.army.mil/Media/Fact-Sheets/Fact-Sheet-Article-View/Article/581806/historical-vignette-the-rapid-city-flood-june-1972/
- USGS, & NOAA. (1975). The Black Hills-Rapid City Flood of June 9-1-, 1972: A Description of the Storm and Flood. https://pubs.usgs.gov/pp/0877/report.pdf
- Van Der Veen, A. (2004). Disasters and economic damage: Macro, meso and micro approaches. *Disaster Prevention and Management: An International Journal*, 13(4), 274–279. https://doi.org/10.1108/09653560410556483

Wahl, T. L. (1998). Prediction of Embankment Dam Breach Parameters: A Literature Review and Needs Assessment. https://www.azwater.gov/sites/default/files/1998_Prediction-of-Embankment-Dam-Breach-Parameters.pdf

- Weather Underground. (2024). Rapid City, SD Weather History. https://www.wunderground.com/history/daily/us/sd/rapid-city/KRAP/date/1972-6-10
- Webb Jeffrey. (2021, March 3). The Tragedy at Buffalo Creek JSTOR Daily. https://daily.jstor.org/the-tragedy-at-buffalo-creek/

William E. Davies, James F. Bailey, & Donovan B. Kelly. (1972). West Virginia's Buffalo Creek Flood: A study of the Hydrology and Engineering Geology. https://damfailures.org/wp-content/uploads/2022/05/IR_USGS_Baffalo-Creek.pdf

Yang, S., He, H., Chen, W., & Wang, L. (2018). Direct tangible damage assessment for regional snowmelt flood disasters with HJ-1 and HR satellite images: a case study of the Altay region, northern Xinjiang, China. *Natural Hazards*, 94(3), 1099–1116. *https://link.springer.com/article/10.1007/s11069-018-3458-x*

APPENDIX

The codes used in this research are attached below:

LOL CALCULATION (FROM EQUATION 20 - 26)

import numpy as np import pandas as pd import logging

Set up logging

logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')

def save_to_excel(matrix, file_name):
 try:
 pd.DataFrame(matrix).to_excel(file_name, index=False, header=False)
 except Exception as e:
 logging.error(f'Failed to save {file_name}: {e}")

def process_matrix(XX, file_prefix):

try:

Find the maximum of each column
CC = np.max(XX, axis=0)

```
# Create a diagonal matrix with the maximum values
BB = np.diag(CC)
amalamal = BB
```

Normalize the matrix
YY = np.matmul(XX, np.linalg.pinv(amalamal))

Export normalized matrix to Excel save_to_excel(YY, f'{file_prefix}_RR.xlsx')

Sum the columns of the normalized matrix sum_YY = np.sum(YY, axis=0)

Create a diagonal matrix from the sum of the columns PP = np.diag(sum YY)

Further normalize the matrix
ff = np.matmul(YY, np.linalg.pinv(PP))

Check if the sum of each column in the final matrix is 1
sumff = np.sum(ff, axis=0)

Export the final normalized matrix to Excel save_to_excel(ff, f'{file_prefix}_ff.xlsx')

Compute the natural logarithm of each element in ff, handling zeros

 $ff_log = np.where(ff > 0, np.log(ff), 0)$

Multiply each element in ff by its logarithm
ff_final = ff * ff_log

Export the modified matrix to Excel save_to_excel(ff_final, f'{file_prefix}_final.xlsx')

Calculate the sum of each column in ff_final sum_ff_final = np.sum(ff_final, axis=0)

Get the count of elements in each column (number of rows) count_columns = ff_final.shape[0]

Compute the natural logarithm of the count of elements
ln_count = np.log(count_columns)

Compute the negative of the sum of each column divided by the ln of the count final_result = -sum_ff_final / ln_count

Export the final results to Excel save_to_excel(final_result.reshape(-1, 1), f'{file_prefix}_result.xlsx')

Subtract each value in final_result from 1
adjusted_result = 1 - final_result

Calculate the sum of the adjusted results
sum_adjusted_result = np.sum(adjusted_result)

Export the sum of the adjusted results to Excel save_to_excel(np.array([sum_adjusted_result]).reshape(1, -1), f'{file_prefix}_sum_adjusted_result.xlsx')

Divide each value in adjusted_result by sum_adjusted_result normalized_adjusted_result = adjusted_result / sum_adjusted_result

Export the normalized adjusted results as a single row to Excel save_to_excel(normalized_adjusted_result.reshape(1, -1), f'{file prefix} normalized adjusted result.xlsx')

Check if the sum of normalized adjusted results is 1
sum_normalized_adjusted_result = np.sum(normalized_adjusted_result)

if np.isclose(sum_normalized_adjusted_result, 1):
 logging.info(fThe sum of normalized adjusted results for {file_prefix} is 1.')
else:

logging.warning(fThe sum of normalized adjusted results for {file_prefix} is not 1, it is {sum_normalized_adjusted_result}.')

Load the RR matrix

RR = pd.read_excel(f'{file_prefix}_RR.xlsx').values

```
print("CC shape:", CC.shape)
print("BB shape:", BB.shape)
print("YY shape:", YY.shape)
print("Adjusted normalized result shape:", normalized adjusted result.shape)
```

```
# Initialize a result list
multiplication_sum_results = []
```

```
# Ensure that normalized_adjusted_result is a 1D array
normalized_adjusted_result = normalized_adjusted_result.flatten()
```

```
# Perform the column-wise multiplication and summation
for i in range(YY.shape[0]):
    if normalized_adjusted_result.size ==YY.shape[1]: # Check for correct dimensions
    row_result = np.sum(YY[i, :] * normalized_adjusted_result)
    multiplication_sum_results.append([row_result])
    else:
    raise ValueError("Mismatch in dimensions between YY and
```

```
normalized adjusted result")
```

```
# Convert results to numpy array and export to Excel
multiplication_sum_results = np.array(multiplication_sum_results)
save_to_excel(multiplication_sum_results,
f{file prefix} multiplication sum results.xlsx')
```

Print the result

logging.info(f'The multiplication and summation results for {file_prefix} are saved to {file_prefix}_multiplication_sum_results.xlsx')

```
return np.sum(ff, axis=0)
```

```
except Exception as e:
```

```
logging.error(f"Error processing matrix {file_prefix}: {e}")
```

```
# Define the matrices
```

```
\begin{split} &XX1 = \text{np.array}([[6.1, 3.71, 1, 99], [92.96, 3.71, 0.75, 31047], [92.96, 2.87, 0.75, 31047], \\ & [13.41, 2.37, 0.75, 49], [13.41, 2.62, 0.75, 49], [13.41, 2.95, 0.75, 49], \\ & [13.41, 1.77, 0.75, 49], [6.096, 4.18, 0.75, 17], [11, 0.65, 0.75, 34.8], \\ & [7.6, 4.18, 0.75, 37]]) \\ &XX2 = \text{np.array}([[0.67, 0.8, 0.835, 11.27], [0.33, 0.2, 0.67, 19.79], [0.33, 0.2, 0.67, 24.62], \\ & [0.33, 0.4, 0.835, 9.57], [0.33, 0.4, 0.835, 9.66], [0.33, 0.4, 0.835, 13.2], \\ & [0.33, 0.4, 0.835, 19.32], [0.67, 0.8, 0.67, 3.21], [0.33, 0.2, 0.835, 1.26], \\ & [0.33, 0.2, 0.835, 4.02]]) \\ &XX3 = \text{np.array}([[43000, 0.5], [2987, 1], [4767, 1], [997, 1], [128, 1], [258, 1], [2070, 1], \\ & [500, 0.5], [50, 0.5], [80, 0.5]]) \\ &XX4 = \text{np.array}([[1, 1], [0.2, 0.67], [0.2, 0.33], [1, 1], [1, 1], [0.6, 0.67], [0.6, 0.67], \\ & [1, 1], [1, 1], [1, 0.67]]) \end{split}
```

XX5 = np.array([[38.3, 17.882, 1, 1890], [92.96, 26.95, 0.75, 31047], [92.96, 10.08, 0.75, 31047],

 $[92.96, 8.83, 0.75, 31047], [13.41, 14.16, 0.75, 49], [13.41, 9.22, 0.75, 49], \\ [60, 52.86, 0.75, 4687.2], [20.1, 18.58, 0.75, 95], [20.1, 13.66, 0.75, 95], \\ [11.6, 7.9, 0.75, 77.7], [13.1, 9.75, 0.75, 227.12], [14, 11.15, 0.75, 86.3], \\ [12.8, 11.15, 1, 55.5], [10.05, 5.57, 1, 178.7], [26.21, 14.59, 0.75, 136], \\ [7.9, 16.21, 0.75, 79.8], [5.2, 11.24, 1, 3.1], [5.2, 6.6, 1, 3.1], \\ [10.97, 6.97, 0.75, 4.93], [33.00, 33.00, 1, 1100], [47.85, 47.85, 1, 4193.8], \\ [21.30, 21.30, 1, 617]])$

$$\begin{split} XX6 &= \text{np.array}([[0.33, 0.4, 0.835, 22.05], [0.33, 0.2, 1, 8.05], [0.33, 0.2, 0.67, 13.52], \\ & [0.33, 0.2, 0.67, 14.16], [0.33, 0.4, 0.835, 1.5], [0.33, 0.4, 0.835, 6.78], \\ & [1, 0.2, 0.835, 29.9], [0.33, 0.2, 0.67, 0.8], [0.33, 0.2, 0.835, 4.8], \\ & [1, 0.8, 0.67, 1.6], [0.33, 0.2, 0.835, 4.82], [0.33, 0.2, 0.835, 2.4100], \\ & [1, 0.6, 0.835, 2.4], [0.67, 0.8, 0.835, 1.5], [0.33, 0.20, 1, 11.58], \\ & [0.33, 0.2, 1, 6.437], [0.33, 0.2, 1, 11.265], [0.33, 0.2, 0.835, 19.31], \\ & [1, 0.8, 0.67, 60]]) \end{split}$$

$$\begin{split} XX7 &= np.array([[37700, 1], [2, 1], [536, 1], [1466, 1], [429, 1], [618, 1], [150, 1], \\ & [100, 0.5], [16400, 0.5], [100, 0.5], [888, 1], [900, 1], [150, 1], [7, 1], \\ & [50, 1], [25, 1], [275, 1], [4000, 0.5], [8, 0.5], [125, 0.5], [250, 1], \\ & [5000, 0.5]]) \end{split}$$

 $\begin{aligned} XX8 &= np.array([[1, 1], [1, 1], [0.8, 1], [0.6, 0.67], [1, 1], [1, 1], [1, 1], [0.2, 0.33], \\ & [0.2, 0.33], [1, 1], [1, 0.67], [1, 1], [1, 1], [1, 1], [1, 1], [1, 1], \\ & [0.8, 0.67], [0.6, 0.33], [1, 1], [1, 0.67], [1, 0.33], [0.2, 0.33]]) \end{aligned}$

Process each matrix

for i, matrix in enumerate([XX1, XX2, XX3, XX4, XX5, XX6, XX7, XX8], start=1): process_matrix(matrix, f'XX{i}')

REGRESSION RELATIONSHIP (LOW SEVERITY) EQUATION 29 - 32

import numpy as np import pandas as pd from sklearn.preprocessing import PolynomialFeatures from sklearn.linear_model import LinearRegression from sklearn.metrics import mean_squared_error, r2_score from sklearn.model_selection import train_test_split import itertools

```
# Data
```

 $\begin{aligned} FL &= np.array([0.005534884, 0, 0, 0.052156469, 0.015625, 0.031007752, 0, 0.012, 0.02, 0.025]) \\ Y1 &= np.array([0.052858348, 0.995486301, 0.98829137, 0.059986756, 0.062128105, 0.064954685, 0.05484752, 0.054523056, 0.038262511, 0.059139302]) \\ Y2 &= np.array([0.711537492, 0.580781003, 0.68514609, 0.44562505, 0.447569741, 0.524060922, 0.656299915, 0.5351082, 0.182662125, 0.242299318]) \\ Y3 &= np.array([0.983339522, 0.100471427, 0.140487444, 0.05573442, 0.036198511, 0.039121029, 0.079856435, 0.027900932, 0.017784523, 0.01845895]) \\ Y4 &= np.array([1, 0.331173601, 0.23628206, 1, 1, 0.619536494, 0.619536494, 1, 1, 0.907899387]) \end{aligned}$

```
# Combine the dependent variables into a single matrix Ys = np.vstack([Y1, Y2, Y3, Y4]).T
```

```
# Split the data into training and test sets
Ys_train, Ys_test, FL_train, FL_test = train_test_split(Ys, FL, test_size=0.20,
random_state=32)
```

```
# Transformation functions
def reciprocal(x):
    return np.where(x == 0, 0, 1 / x)

def square_root(x):
    return np.sqrt(x)

def cubic(x):
    return x ** 3

def exponential(x):
    return np.exp(x)

def logarithmic(x):
    return np.where(x == 0, 0, np.log(x))

def power(x):
    return np.power(x, 2)

# Transformations dictionary
transformations = {
```

```
'reciprocal': reciprocal,
  'square root': square root,
  'cubic': cubic,
  'linear': lambda x: x,
  'exponential': exponential,
  'logarithmic': logarithmic,
  'power': power
}
# Degree of the polynomial (for non-linear fitting)
degree = 2 \# You can change this value
# Generate all possible combinations of transformations
all combinations = list(itertools.product(transformations.keys(), repeat=Ys train.shape[1]))
# Fit models and calculate metrics for all combinations
combination results = []
for combination in all combinations:
  transformed_Ys_train = []
  transformed Ys test = []
  for i, transform name in enumerate(combination):
     transform func = transformations[transform name]
     transformed Ys train.append(transform func(Ys train[:, i]).reshape(-1, 1))
    transformed Ys test.append(transform func(Ys test[:, i]).reshape(-1, 1))
  combined data train = np.hstack(transformed Ys train)
  combined data test = np.hstack(transformed Ys test)
  # Polynomial transformation
  poly = PolynomialFeatures(degree=degree)
  poly combined data train = poly.fit transform(combined data train)
  poly combined data test = poly.transform(combined data test)
  # Non-linear (Polynomial) regression
  model = LinearRegression()
  model.fit(poly combined data train, FL train)
  FL pred train = model.predict(poly combined data train)
  FL pred test = model.predict(poly combined data test)
  rmse train = np.sqrt(mean squared error(FL train, FL pred train))
  rmse test = np.sqrt(mean squared error(FL test, FL pred test))
  r2 train = r2 score(FL train, FL pred train)
  r2 test = r2 score(FL test, FL pred test)
```

combination_results.append((combination, rmse_train, r2_train, rmse_test, r2_test))

Convert the results to a DataFrame

```
combination_results_df = pd.DataFrame(combination_results, columns=['Transformations', 'RMSE Train', 'R2 Train', 'RMSE Test', 'R2 Test'])
```

Find the best combination based on R2 on the test set best_combination = combination_results_df.loc[combination_results_df['R2_Test'].idxmax()]

print("All Combinations Results:")
print(combination_results_df)

print("\nBest Combination:")
print(best_combination)

Extract the best combination details best_transformations = best_combination['Transformations']

Transform the training and test data with the best transformations best_transformed_Ys_train = [] best_transformed_Ys_test = []

for i, transform_name in enumerate(best_transformations):
 transform_func = transformations[transform_name]
 best_transformed_Ys_train.append(transform_func(Ys_train[:, i]).reshape(-1, 1))
 best_transformed_Ys_test.append(transform_func(Ys_test[:, i]).reshape(-1, 1))

combined_data_best_train = np.hstack(best_transformed_Ys_train)
combined_data_best_test = np.hstack(best_transformed_Ys_test)

Polynomial transformation for the best model
poly_combined_data_best_train = poly.fit_transform(combined_data_best_train)
poly_combined_data_best_test = poly.transform(combined_data_best_test)

Perform regression on the best combined data model = LinearRegression() model.fit(poly_combined_data_best_train, FL_train) FL_pred_best_train = model.predict(poly_combined_data_best_train) FL_pred_best_test = model.predict(poly_combined_data_best_test)

coefficients_best = model.coef_
intercept_best = model.intercept_

Output the best combined regression results print("\nBest Combined Regression Results:") print(f"Coefficients: {coefficients_best}") print(f"Intercept: {intercept_best}") print(f"RMSE (Train): {best_combination['RMSE_Train']}") print(f"R2 (Train): {best_combination['R2_Train']}") print(f"RMSE (Test): {best_combination['RMSE_Test']}") print(f"R2 (Test): {best_combination['R2_Test']}")

Calculate FL for each row in the test set using the best combination relationship

```
FL_pred_test_rows = []
for i in range(len(FL_test)):
    row_data = []
    for j, transform_name in enumerate(best_transformations):
        transform_func = transformations[transform_name]
        row_data.append(transform_func(Ys_test[i, j]))
    row_data = np.array(row_data).reshape(1, -1)
        poly_row_data = poly.transform(row_data)
        FL_pred_row = model.predict(poly_row_data)
    # If the predicted FL is negative, set it to zero
    FL_pred_row = max(FL_pred_row[0], 0)
    FL pred_test rows.append(FL pred_row)
```

FL_pred_test_rows = np.array(FL_pred_test_rows)

print("\nPredicted FL for each row in the test set:")
print(FL_pred_test_rows)

print("Training Ys:")
print(Ys_train)
print("Training FL:")
print(FL_train)

print("Testing Ys:")
print(Ys_test)
print("Testing FL:")
print(FL_test)

REGRESSION RELATIONSHIP (MEDUIM & HIGH SEVERITY) EQUATION 29 - 32

import numpy as np import pandas as pd from sklearn.preprocessing import PolynomialFeatures from sklearn.linear_model import LinearRegression from sklearn.metrics import mean_squared_error, r2_score from sklearn.model_selection import train_test_split import itertools

Data

FL = np.array([0.058594164, 0.5, 0.009328358, 0, 0.041958042, 0.035598706, 0.56, 0, 0.000304878, 0.39, 0.155405405, 0.0866666667, 0.2666666667, 0.285714286, 0.02, 0.04, 0.007272727, 0, 0.5, 0.68, 0.076, 0.0004]) Y5 = np.array([0.154513604, 0.949493671, 0.917375734, 0.914995922, 0.056810268, 0.949493671, 0.917375734, 0.914995922, 0.056810268, 0.949493671, 0.917375734, 0.914995922, 0.056810268, 0.949493671, 0.917375734, 0.914995922, 0.056810268, 0.949493671, 0.917375734, 0.914995922, 0.056810268, 0.949493671, 0.917375734, 0.914995922, 0.056810268, 0.949493671, 0.917375734, 0.914995922, 0.056810268, 0.949493671, 0.917375734, 0.917375734, 0.914995922, 0.056810268, 0.949493671, 0.917375734, 0.917375734, 0.914995922, 0.056810268, 0.949493671, 0.917375734, 0.91737574, 0.91737574, 0.917375744, 0.917375744, 0.917444, 0.91744, 0.91744, 0.91744, 0.91744, 0.9174444, 0.9174444, 0.917444, 0.917444, 0.917444, 0.917444, 0.9174444, 0.90.047405252, 0.325555465, 0.07885526, 0.069488321, 0.042158839, 0.051963343, 0.053052748, 0.051262519, 0.038332261, 0.083683737, 0.051080339, 0.035946569, 0.027112709, 0.037517409, 0.155020445, 0.282934817, 0.099571713]) $Y_6 = np.array([0.393736933, 0.201655533, 0.253430168, 0.259905885, 0.185805681,$ 0.239230353, 0.537018648, 0.124725276, 0.166984771, 0.405518234, 0.167187137, 0.142802012, 0.363187809, 0.349126653, 0.237373164, 0.185334701, 0.234185897, 0.313801435, 0.351855303, 0.584946107, 0.355997351, 0.996427484]) 0.028790789, 0.015042005, 0.436632155, 0.015042005, 0.047878735, 0.048189109, 0.028790789, 0.025092176, 0.026204346, 0.025557735, 0.032023842, 0.115913268, 0.012662478, 0.015688616, 0.031377231, 0.141777694]) Y8 = np.array([1, 1, 0.898356491, 0.634424772, 1, 1, 1, 0.263931719, 0.263931719, 1, 0.83771179, 1, 1, 1, 1, 1, 0.736068281, 0.467218737, 1, 0.83771179, 0.670505754, 0.263931719])

Combine the dependent variables into a single matrix Ys = np.vstack([Y5, Y6, Y7, Y8]).T

Split the data into training and test sets (Ys, FL, test_size=0.30, random_state=12330, or test_size=0.20, random_state=17081) Ys_train, Ys_test, FL_train, FL_test = train_test_split(Ys, FL, test_size=0.20, random_state=17081)

Transformation functions
def reciprocal(x):
 return np.where(x == 0, 0, 1 / x)
def square_root(x):
 return np.sqrt(x)

def cubic(x): return x ** 3

def exponential(x):
 return np.exp(x)

```
def logarithmic(x):
  return np.where(x == 0, 0, np.log(x))
def power(x):
  return np.power(x, 2)
# Transformations dictionary
transformations = {
  'reciprocal': reciprocal,
  'square root': square root,
  'cubic': cubic.
  'linear': lambda x: x,
  'exponential': exponential,
  'logarithmic': logarithmic,
  'power': power
}
# Degree of the polynomial (for non-linear fitting)
degree = 2 \# You can change this value
# Generate all possible combinations of transformations
all combinations = list(itertools.product(transformations.keys(), repeat=Ys train.shape[1]))
# Fit models and calculate metrics for all combinations
combination results = []
for combination in all combinations:
  transformed Ys train = []
  transformed Ys test = []
  for i, transform name in enumerate(combination):
     transform func = transformations[transform name]
     transformed Ys train.append(transform func(Ys train[:, i]).reshape(-1, 1))
     transformed_Ys_test.append(transform_func(Ys_test[:, i]).reshape(-1, 1))
  combined data train = np.hstack(transformed Ys train)
  combined data test = np.hstack(transformed Ys test)
  # Polynomial transformation
  poly = PolynomialFeatures(degree=degree)
  poly combined data train = poly.fit transform(combined data train)
  poly combined data test = poly.transform(combined data test)
  # Non-linear (Polynomial) regression
  model = LinearRegression()
  model.fit(poly combined data train, FL train)
  FL_pred_train = model.predict(poly_combined data train)
  FL pred test = model.predict(poly_combined_data_test)
```
rmse_train = np.sqrt(mean_squared_error(FL_train, FL_pred_train))
rmse_test = np.sqrt(mean_squared_error(FL_test, FL_pred_test))
r2_train = r2_score(FL_train, FL_pred_train)
r2_test = r2_score(FL_test, FL_pred_test)

combination_results.append((combination, rmse_train, r2_train, rmse_test, r2_test))

Convert the results to a DataFrame combination_results_df = pd.DataFrame(combination_results, columns=['Transformations', 'RMSE_Train', 'R2_Train', 'RMSE_Test', 'R2_Test'])

Find the best combination based on R2 on the test set best_combination = combination_results_df.loc[combination_results_df['R2_Test'].idxmax()]

print("All Combinations Results:")
print(combination_results_df)

print("\nBest Combination:")
print(best combination)

Extract the best combination details
best transformations = best combination['Transformations']

Transform the training and test data with the best transformations best_transformed_Ys_train = [] best_transformed_Ys_test = []

for i, transform_name in enumerate(best_transformations):
 transform_func = transformations[transform_name]
 best_transformed_Ys_train.append(transform_func(Ys_train[:, i]).reshape(-1, 1))
 best_transformed_Ys_test.append(transform_func(Ys_test[:, i]).reshape(-1, 1))

combined_data_best_train = np.hstack(best_transformed_Ys_train)
combined_data_best_test = np.hstack(best_transformed_Ys_test)

Polynomial transformation for the best model
poly_combined_data_best_train = poly.fit_transform(combined_data_best_train)
poly_combined_data_best_test = poly.transform(combined_data_best_test)

Perform regression on the best combined data model = LinearRegression() model.fit(poly_combined_data_best_train, FL_train) FL_pred_best_train = model.predict(poly_combined_data_best_train) FL_pred_best_test = model.predict(poly_combined_data_best_test)

coefficients_best = model.coef_
intercept_best = model.intercept_

Output the best combined regression results
print("\nBest Combined Regression Results:")

print(f"Coefficients: {coefficients_best}")
print(f"Intercept: {intercept_best}")
print(f"RMSE (Train): {best_combination['RMSE_Train']}")
print(f"R2 (Train): {best_combination['R2_Train']}")
print(f"RMSE (Test): {best_combination['RMSE_Test']}")
print(f"R2 (Test): {best_combination['R2_Test']}")

Calculate FL for each row in the test set using the best combination relationship FL_pred_test_rows = []

```
for i in range(len(FL_test)):
    row_data = []
    for j, transform_name in enumerate(best_transformations):
        transform_func = transformations[transform_name]
        row_data.append(transform_func(Ys_test[i, j]))
        row_data = np.array(row_data).reshape(1, -1)
        poly_row_data = poly.transform(row_data)
        FL_pred_row = model.predict(poly_row_data)
```

```
# If the predicted FL is negative, set it to zero
FL_pred_row = max(FL_pred_row[0], 0)
FL_pred_test_rows.append(FL_pred_row)
```

```
FL_pred_test_rows = np.array(FL_pred_test_rows)
```

print("\nPredicted FL for each row in the test set:")
print(FL_pred_test_rows)

print("Training Ys:")
print(Ys_train)
print("Training FL:")
print(FL_train)

print("Testing Ys:")
print(Ys_test)
print("Testing FL:")
print(FL test)

Assuming best_transformations and best_combination from previous steps

```
# Variables corresponding to the Ys
variables = ['Y5', 'Y6', 'Y7', 'Y8']
```

```
# Apply the best transformations to the training data
best_transformed_Ys_train = []
for i, (transform_name, var) in enumerate(zip(best_transformations, variables)):
    transform_func = transformations[transform_name]
    best_transformed_Ys_train.append(transform_func(Ys_train[:, i]).reshape(-1, 1))
```

Combine the transformed variables for training

combined_data_best_train = np.hstack(best_transformed_Ys_train)

Polynomial transformation for the best model
poly = PolynomialFeatures(degree=2)
poly_combined_data_best_train = poly.fit_transform(combined_data_best_train)

Fit the regression model model = LinearRegression() model.fit(poly combined data best train, FL train)

Get the coefficients and intercept coefficients_best = model.coef_ intercept_best = model.intercept_

Get the feature names from polynomial transformation
poly_feature_names = poly.get_feature_names_out(variables)

Map the coefficients to the features
print("\nMapping of Polynomial Features to Coefficients:")
for feature, coef in zip(poly_feature_names, coefficients_best):
 print(f"{feature}: {coef}")

Output the best regression results
print("\nBest Combined Regression Results:")
print(f"Intercept: {intercept_best}")
print(f"RMSE (Train): {best_combination['RMSE_Train']}")
print(f"R2 (Train): {best_combination['R2_Train']}")
print(f"RMSE (Test): {best_combination['RMSE_Test']}")
print(f"R2 (Test): {best_combination['R2_Test']}")

PLUGIN CODE

The full code and file can be found here: <u>lol_framework</u> https://gounbc.sharepoint.com/:f:/r/sites/Impactsofdambreaks/Shared%20Documents/Framework%20plugin/lol_ framework?csf=1&web=1&e=zb0Ej9

Or

https://github.com/ovuchukwu/LOL-Framework/tree/2d84fa1f600a9388169db9dabcd88cc0c09dcba6/lol_framework

-*- coding: utf-8 -*-,,,,,, LOLFrameworkDialog A QGIS plugin This plugin estimates life loss resulting from dam failure Generated by Plugin Builder: http://g-sherman.github.io/Qgis-Plugin-Builder/ _____ : 2024-11-12 begin git sha : \$Format:%H\$: (C) 2024 by UNBC copyright : ovu@unbc.ca email * * This program is free software; you can redistribute it and/or modify * * it under the terms of the GNU General Public License as published by * * the Free Software Foundation; either version 2 of the License, or * (at your option) any later version. * ,,,,,, import os import numpy as np import pandas as pd import logging

from qgis.PyQt import uic from qgis.PyQt import QtWidgets from .lol framework dialog base import Ui LOLFrameworkDialogBase

import math

This loads your .ui file so that PyQt can populate your plugin with the elements from Qt Designer

FORM_CLASS, _ = uic.loadUiType(os.path.join(

os.path.dirname(__file__), 'lol_framework_dialog_base.ui'))

def classify_and_grade_input(input_data):

Classifies and grades the input data based on predefined grading standards.

Args:

input_data (dict): User input where keys are influencing factors (e.g., 'HD', 'MB', 'WB') and

values are the provided inputs.

Returns:

list: Graded and normalized inputs.

.....

grading_standards = {

'WB': {'Level I (storm, blizzard, typhoon, fog)': 1.0, 'Level II (heavy rain, heavy snow, gale)': 0.8, 'Level III (moderate rain, moderate snow)': 0.6, 'Level IV (light rain, shower, light snow)': 0.4, 'Level V (sunny or cloudy day)': 0.2},

'MB': {'Overtopping': 1.0, 'Poor quality (leakage, internal erosion, tunnel, blockage or obstruction of dam structure, spillway etc.)': 0.75, 'Mismanagement (Overage storage, poor maintenance, dams without maintenance or management etc.)': 0.5, 'Others': 0.25},

'TW': {'0-15': 1.0, '15-30': 0.8, '30-45': 0.6, '45-60': 0.4, '>60': 0.2},

'VB': {'Adobe': 1.0, 'Wood': 0.835, 'Masonry(Brick/Stone)': 0.67, 'Concrete': 0.33},

'EC': {'Bad': 1.0, 'Middle': 0.67, 'Good': 0.33},

'UB': {'Vague/Fuzzy': 1.0, 'Clear/Precise': 0.5},

'TB': {'Midnight (00:00 - 07:59:59)': 1.0, 'Night (20:00 - 23:59:59)': 0.67, 'Daytime (08:00 - 19:59:59)': 0.33},

}

```
normalized_data = []
```

```
for factor, value in input_data.items():
```

if factor in grading_standards: # Non-numeric factors
 graded_value = grading_standards[factor].get(value, None)
 if graded_value is None:
 raise ValueError(f"Invalid value '{value}' for factor '{factor}'.")
 normalized_data.append(graded_value)
 else: # Numeric factors

normalized data.append(value)

```
return normalized data
```

pass

def assign_to_matrix(processed_data, SF, matrices_low, matrices_high):

Assigns processed data to the appropriate matrix group based on SF.

Args:

```
processed data (list): Graded and normalized data.
    SF (float): Severity of flood.
    matrices low (list): [XX1, XX2, XX3, XX4].
    matrices high (list): [XX5, XX6, XX7, XX8].
  ,,,,,,
  target matrices = matrices low if SF < 4.6 else matrices high
  modules = [4, 4, 2, 2] # Module sizes
  start idx = 0
  for i, matrix in enumerate(target_matrices):
    module size = modules[i]
    new row = processed data[start_idx:start_idx + module_size]
    if len(new row) != module size:
       raise ValueError(f"Data length mismatch for module {i + 1}: Expected
{module size}, got {len(new row)}.")
    target matrices[i] = np.vstack([matrix, new row])
    start idx += module size
```

pass

def validate_inputs(input_data, expected_fields):

~ . . .

Validates input data against expected fields.

Args:

input_data (dict): User inputs where keys are field names and values are provided inputs.

expected_fields (dict): Dictionary with field names as keys and value types (e.g., float, str) as values.

Returns:

bool: True if all inputs are valid, raises ValueError otherwise.

```
.....
```

for field, field_type in expected_fields.items():

```
if field not in input_data:
```

raise ValueError(f"Missing input: {field}")

```
if not isinstance(input_data[field], field_type):
```

raise ValueError(f"Invalid type for {field}. Expected {field_type.__name__}.") return True

def initialize_matrices():

Initializes matrices for low and high severity cases.

Returns:

dict: Dictionary containing 'low' and 'high' severity matrices.

return {

```
"low": [np.empty((0, 4)), np.empty((0, 4)), np.empty((0, 2)), np.empty((0, 2))],
"high": [np.empty((0, 4)), np.empty((0, 4)), np.empty((0, 2)), np.empty((0, 2))]
}
```

def process_matrix_in_memory(matrix, display=True):

......

Processes a matrix in memory and optionally displays the normalized adjusted result and multiplication sum results.

Args:

matrix (np.array): The matrix to process.

display (bool): Whether to display the normalized adjusted result and multiplication sum results.

Returns:

dict: Contains the processed results (normalized adjusted result and multiplication sum results).

.....

try:

Perform matrix operations CC = np.max(matrix, axis=0) BB = np.diag(CC) YY = np.matmul(matrix, np.linalg.pinv(BB))

sum_YY = np.sum(YY, axis=0)
PP = np.diag(sum_YY)
ff = np.matmul(YY, np.linalg.pinv(PP))

```
ff_log = np.where(ff > 0, np.log(ff), 0)
ff_final = ff * ff_log
```

```
sum_ff_final = np.sum(ff_final, axis=0)
count_columns = ff_final.shape[0]
ln_count = np.log(count_columns)
final result = -sum ff_final / ln_count
```

```
adjusted_result = 1 - final_result
sum_adjusted_result = np.sum(adjusted_result)
normalized_adjusted_result = adjusted_result / sum_adjusted_result
```

```
# Correct computation for multiplication sum results
# Ensure normalized_adjusted_result is a 1D array
normalized_adjusted_result = normalized_adjusted_result.flatten()
```

```
# Compute multiplication sum results
multiplication_sum_results = []
for i in range(YY.shape[0]):
    if normalized_adjusted_result.size == YY.shape[1]: # Check for correct dimensions
    row_result = np.sum(YY[i, :] * normalized_adjusted_result)
```

```
multiplication_sum_results.append([row_result])
else:
    raise ValueError("Mismatch in dimensions between YY and
normalized_adjusted_result.")

# Convert results to numpy array
multiplication_sum_results = np.array(multiplication_sum_results)

# Log and print results
if display:
    #logging.info("==== Results for Processed Matrix ===="")
    #logging.info(f"Normalized Adjusted Result:\n {normalized_adjusted_result}")
    logging.info(f"Multiplication Sum Results:\n {multiplication_sum_results}")

# print("\n=== Results for Processed Matrix ===="")
#print("\n=== Results for Processed Matrix ===="")
#print("\ne== Results for Processed Matrix ===="")
#print("\n=== Results for Processed Matrix ===="")
#print("\ne== Results for Processed Matrix ====""")
#print("Multiplication Sum Results:", multiplication sum results)
```

```
#sys.stdout.flush() # Ensure it appears in the console
```

return {

```
"normalized_adjusted_result": normalized_adjusted_result,
"multiplication sum results": multiplication sum results,
```

```
}
```

```
except Exception as e:
```

```
logging.error(f"Error processing matrix in memory: {e}")
return None
```

def calculate_fatality_rate_low(Y):

Calculates fatality rate FL for low severity using the provided equation.

Args:

Y (list or array): Multiplication sum results [Y1, Y2, Y3, Y4].

Returns:

```
float: Calculated fatality rate FL (low severity).

"""
Y1, Y2, Y3, Y4 = Y
FL = (
1.1270 +
1.2347 * np.log(Y1) +
0.4266 * np.sqrt(Y2) +
1.0421 * np.exp(Y3) -
1.3593 * np.sqrt(Y4) -
0.0629 * (np.log(Y1)*2) -
0.1314 * np.log(Y1) * np.sqrt(Y2) -
0.5220 * np.log(Y1) * np.exp(Y3) -
1.1981 * np.log(Y1) * np.sqrt(Y4) -
1.9268 * (np.sqrt(Y2)*2) -
```

```
0.0792 * np.sqrt(Y2) * np.exp(Y3) +

1.6188 * np.sqrt(Y2) * np.sqrt(Y4) -

0.8520 * (np.exp(Y3)**2) +

0.6823 * np.exp(Y3) * np.sqrt(Y4) -

2.2349 * (np.sqrt(Y4)**2)

)

return EL
```

return FL

def calculate_fatality_rate_high(Y):
 """

Calculates fatality rate FL for high severity using the provided equation.

Args:

Y (list or array): Multiplication sum results [Y1, Y2, Y3, Y4].

Returns:

```
float: Calculated fatality rate FL (high severity).
.....
Y1, Y2, Y3, Y4 = Y
FL = (
  -2.2533 +
  11.3349 * Y1**2 -
  0.1300 * (1 / Y2) -
  0.0378 * (1 / Y3) +
  2.6903 * np.exp(Y4) +
  12.7990 * (Y1**2)**2 -
  4.2117 * Y1**2 * (1 / Y2) +
  0.4114 * Y1**2 * (1 / Y3) -
  6.6235 * Y1 * 2 * np.exp(Y4) +
  0.0375 * (1 / Y2)**2 +
  0.0009 * (1 / Y2) * (1 / Y3) -
  0.1023 * (1 / Y2) * np.exp(Y4) +
  0.0001 * (1 / Y3)**2 +
  0.0098 * (1 / Y3) * np.exp(Y4) -
  0.5143 * (np.exp(Y4)**2)
)
return FL
```

def calculate_fatality_rate(severity, Y):

Calculates the fatality rate FL based on severity.

Args:

severity (str): Either 'low' or 'high'. Y (list or array): Multiplication sum results [Y1, Y2, Y3, Y4].

Returns:

float: Fatality rate FL.

.....

```
if severity == "low":
    return calculate_fatality_rate_low(Y)
elif severity == "high":
    return calculate_fatality_rate_high(Y)
else:
    raise ValueError("Severity must be 'low' or 'high'.")
```

```
class LOLFrameworkDialog(QtWidgets.QDialog, FORM CLASS):
  def init (self, parent=None):
     ""Constructor."""
     super(LOLFrameworkDialog, self). init (parent)
     # Set up the user interface from Designer through FORM CLASS.
     # After self.setupUi() you can access any designer object by doing
     # self.<objectname>, and you can use autoconnect slots - see
     # http://qt-project.org/doc/qt-4.8/designer-using-a-ui-file.html
    # #widgets-and-dialogs-with-auto-connect
     self.ui = Ui LOLFrameworkDialogBase()
     self.setupUi(self)
     #self.ui.pButtonRunLOLModel.clicked.connect(self.onpButtonRunLOLModelClicked)
     self.pButtonRunLOLModel.clicked.connect(self.pbuttonrunlolmodelclicked)
     self.matrices = initialize matrices()
     self.pButtonExport.clicked.connect(self.save matrices to file)
     # Initialize matrices and reset inputs
     self.reset inputs and matrices()
     self.XX1 = np.array([[6.1, 3.71, 1, 99], [92.96, 3.71, 0.75, 31047], [92.96, 2.87, 0.75,
31047],
          [13.41, 2.37, 0.75, 49], [13.41, 2.62, 0.75, 49], [13.41, 2.95, 0.75, 49],
         [13.41, 1.77, 0.75, 49], [6.096, 4.18, 0.75, 17], [11, 0.65, 0.75, 34.8],
         [7.6, 4.18, 0.75, 37]])
     self.XX2 = np.array([[0.67, 0.8, 0.835, 11.27], [0.33, 0.2, 0.67, 19.79], [0.33, 0.2, 0.67,
24.62],
          [0.33, 0.4, 0.835, 9.57], [0.33, 0.4, 0.835, 9.66], [0.33, 0.4, 0.835, 13.2],
          [0.33, 0.4, 0.835, 19.32], [0.67, 0.8, 0.67, 3.21], [0.33, 0.2, 0.835, 1.26],
          [0.33, 0.2, 0.835, 4.02]])
     self.XX3 = np.array([[43000, 0.5], [2987, 1], [4767, 1], [997, 1], [128, 1], [258, 1],
[2070, 1],
          [500, 0.5], [50, 0.5], [80, 0.5]])
     self.XX4 = np.array([[1, 1], [0.2, 0.67], [0.2, 0.33], [1, 1], [1, 1], [0.6, 0.67], [0.6, 0.67],
          [1, 1], [1, 1], [1, 0.67]])
     self.XX5 = np.array([[38.3, 17.882, 1, 1890], [92.96, 26.95, 0.75, 31047], [92.96,
10.08, 0.75, 31047],
          [92.96, 8.83, 0.75, 31047], [13.41, 14.16, 0.75, 49], [13.41, 9.22, 0.75, 49],
          [60, 52.86, 0.75, 4687.2], [20.1, 18.58, 0.75, 95], [20.1, 13.66, 0.75, 95],
          [11.6, 7.9, 0.75, 77.7], [13.1,
                                             9.75, 0.75, 227.12], [14, 11.15, 0.75, 86.3],
```

[7.9, 16.21, 0.75, 79.8], [5.2, 11.24, 1, 3.1], [5.2, 6.6, 1, 3.1],

[10.97, 6.97, 0.75, 4.93], [33.00, 33.00, 1, 1100], [47.85, 47.85, 1, 4193.8], [21.30, 21.30, 1, 617]])

self.XX6 = np.array([[0.33, 0.4, 0.835, 22.05], [0.33, 0.2, 1, 8.05], [0.33, 0.2, 0.67, 13.52],

[0.33, 0.2, 0.67, 14.16], [0.33, 0.4, 0.835, 1.5], [0.33, 0.4, 0.835, 6.78],

[1, 0.2, 0.835, 29.9], [0.33, 0.2, 0.67, 0.8], [0.33, 0.2, 0.835, 4.8],

[1, 0.8, 0.67, 1.6], [0.33, 0.2, 0.835, 4.82], [0.33, 0.2, 0.835, 2.4100],

[1, 0.6, 0.835, 2.4], [0.67, 0.8, 0.835, 1.5], [0.33, 0.20, 1, 11.58],

[0.33, 0.2, 1, 6.437], [0.33, 0.2, 1, 11.265], [0.33, 0.2, 0.835, 19.31],

[1, 0.6, 0.835, 1.28], [1, 0.6, 1, 24.14], [0.33, 0.8, 0.835, 8],

[1, 0.8, 0.67, 60]])

self.XX7 = np.array([[37700, 1], [2, 1], [536, 1], [1466, 1], [429, 1], [618, 1], [150, 1], [100, 0.5], [16400, 0.5], [100, 0.5], [888, 1], [900, 1], [150, 1], [7, 1], [50, 1], [25, 1], [275, 1], [4000, 0.5], [8, 0.5], [125, 0.5], [250, 1], [5000, 0.5]])

self.XX8 = np.array([[1, 1], [1, 1], [0.8, 1], [0.6, 0.67], [1, 1], [1, 1], [1, 1], [0.2, 0.33], [0.2, 0.33], [1, 1], [1, 0.67], [1, 1], [1, 1], [1, 1], [1, 1], [1, 1], [0.8, 0.67], [0.6, 0.33], [1, 1], [1, 0.67], [1, 0.33], [0.2, 0.33]])

#self.comboBoxMB.addItems(['Overtopping', 'Poor quality (leakage, internal erosion, tunnel, \nblockage or obstruction of dam structure, spillway \netc.)', 'Mismanagement (Overage storage, poor maintenance, \ndams without maintenance or management etc.)', 'Others'])

self.comboBoxMB.setToolTip("Select the dam break mode.")

self.comboBoxMB.addItems([

'Overtopping',

'Poor quality (leakage, internal erosion, tunnel, blockage or obstruction of dam structure, spillway etc.)',

'Mismanagement (Overage storage, poor maintenance, dams without maintenance or management etc.)',

'Others'

])

self.comboBoxWB.addItems(['Level I (storm, blizzard, typhoon, fog)', 'Level II (heavy rain, heavy snow, gale)', 'Level III (moderate rain, moderate snow)', 'Level IV (light rain, shower, light snow)', 'Level V (sunny or cloudy day)'])

self.comboBoxVB.addItems(['Adobe', 'Wood', 'Masonry(Brick/Stone)', 'Concrete'])

self.comboBoxTW.addItems(['0-15', '15-30', '30-45', '45-60', '>60'])

self.comboBoxEC.addItems(['Bad', 'Middle', 'Good'])

self.comboBoxUB.addItems(['Vague/Fuzzy', 'Clear/Precise'])

self.comboBoxTB.addItems(['Midnight (00:00 - 07:59:59)', 'Daytime (08:00 - 19:59:59)', 'Night (20:00 - 23:59:59)'])

#clear/reset

def reset_inputs_and_matrices(self):

Resets all input fields and calculated matrices.

Clears any user input and resets calculated matrices in memory.

Clear line edits

self.lineEditHD.clear()
self.lineEditSF.clear()
self.lineEditPR.clear()
self.lineEditSW.clear()
self.lineEditDD.clear()

Reset combo boxes self.comboBoxMB.setCurrentIndex(0) self.comboBoxWB.setCurrentIndex(0) self.comboBoxVB.setCurrentIndex(0) self.comboBoxTW.setCurrentIndex(0) self.comboBoxEC.setCurrentIndex(0) self.comboBoxUB.setCurrentIndex(0) self.comboBoxTB.setCurrentIndex(0)

Reinitialize matrices (empty them out)
self.matrices = initialize_matrices()

```
def pbuttonrunlolmodelclicked(self):
  # Collect user inputs
  try:
    input data = \{
       'HD': float(self.lineEditHD.text()),
       'SF': float(self.lineEditSF.text()),
       'MB': self.comboBoxMB.currentText(),
       'SW': float(self.lineEditSW.text()),
       'TB': self.comboBoxTB.currentText(),
       'WB': self.comboBoxWB.currentText(),
       'VB': self.comboBoxVB.currentText(),
       'DD': float(self.lineEditDD.text()), # DD is numeric
       'PR': float(self.lineEditPR.text()),
       'UB': self.comboBoxUB.currentText(),
       'TW': self.comboBoxTW.currentText(),
       'EC': self.comboBoxEC.currentText(),
    }
    # Validate inputs
```

expected_fields = {
 'HD': float, 'SF': float, 'MB': str, 'SW': float,
 'TB': str, 'WB': str, 'VB': str, 'DD': float,
 'PR': float, 'UB': str,
 'TW': str, 'EC': str
}
validate_inputs(input_data, expected_fields)
Normalize and classify inputs
SF = input_data['SF']

processed data = classify and grade input(input data)

```
# Dynamic matrix handling
       if not hasattr(self, "matrices"):
         self.matrices = initialize matrices()
       matrix group = "low" if SF < 4.6 else "high"
       assign to matrix(processed data, SF, self.matrices["low"], self.matrices["high"])
       predefined matrices = {
         "low": [self.XX1, self.XX2, self.XX3, self.XX4],
          "high": [self.XX5, self.XX6, self.XX7, self.XX8],
       }
       # Combine user-inputted data with predefined matrices
       for matrix group, user matrices in self.matrices.items():
         for i, user matrix in enumerate(user matrices):
            if user matrix.size > 0: # Append user-inputted data if available
              predefined matrices[matrix group][i] = np.vstack(
                 [predefined matrices[matrix group][i], user matrix])
       # Create dictionaries to store results
       multiplication sum results = \{\}
       fatality rates = {}
       # Process matrices and compute multiplication sum results and fatality rates
       for matrix group, combined matrices in predefined matrices.items():
         if matrix group == ("low" if SF < 4.6 else "high"):
            # Process matrices for the selected severity
            for i, combined matrix in enumerate(combined matrices, start=1):
              result = process matrix in memory(combined matrix, display=False)
              if result:
                 key = f'' {matrix group} XX{i}''
                 multiplication sum results[key] = result["multiplication sum results"]
            # Calculate fatality rate for the selected severity
            try:
              Y1 = multiplication sum results[f" {matrix group} XX1"][-1][0] # Last row
of selected matrix
              Y2 = multiplication sum results[f" {matrix group} XX2"][-1][0] # Last row
of selected matrix
              Y3 = multiplication sum results[f" {matrix group} XX3"][-1][0] # Last row
of selected matrix
              Y4 = multiplication sum results[f" {matrix group} XX4"][-1][0] # Last row
of selected matrix
              Y = [Y1, Y2, Y3, Y4]
              FL = calculate fatality rate(matrix group, Y)
              fatality rates[matrix group] = FL
              logging.info(f"Fatality Rate ({matrix group.capitalize()} Severity): {FL}")
              # Update labelFatality with the calculated Fatality Rate
```

self.labelFatality.setText(f"{FL:.4f}") # Display Fatality Rate # Now multiply Fatality Rate by PR and update labelLOL PR = input data['PR']result Lol = FL * PRself.labelLOL.setText(f"{round(result Lol)}") # Display Fatality Rate * PR # Switch to the Results tab self.tabWidget.setCurrentIndex(2) # Now, call method to save matrices and results #self.save matrices to file(fatality rates, multiplication sum results, predefined matrices) except KeyError as e: logging.error(f"Error extracting {matrix_group} severity results: {e}") self.labelLOL.setText(f"Error: {e}") except ValueError as ve: self.labelLOL.setText(f"Input Error: {ve}") logging.error(f"Validation error: {ve}") except Exception as e: self.labelLOL.setText(f"Processing Error: {e}") logging.error(f"Unexpected error: {e}") def save matrices to file(self, fatality rates, multiplication sum results, predefined matrices, filename=None): """Saves matrices, results, and fatality rates to an Excel file.""" try: # Allow user to save file using QFileDialog if not filename: from PyQt5.QtWidgets import QFileDialog filename, = QFileDialog.getSaveFileName(self, "Save Matrices", "", "Excel Files (*.xlsx)") if not filename: self.labelLOL.setText("Save operation canceled.") return

Write matrices and results to file with pd.ExcelWriter(filename) as writer: for matrix_group, combined_matrices in predefined_matrices.items(): for i, combined_matrix in enumerate(combined_matrices, start=1): sheet_name = f"{matrix_group}_XX{i}" pd.DataFrame(combined_matrix).to_excel(writer, sheet_name=sheet_name, index=False, header=False)

Write multiplication sum results
for key, mult_sum_result in multiplication_sum_results.items():
 sheet_name = f' {key}_mult_sum_result"

pd.DataFrame(mult_sum_result).to_excel(writer, sheet_name=sheet_name, index=False, header=False)

Write fatality rates

```
fatality_rate_df = pd.DataFrame.from_dict(fatality_rates, orient="index",
columns=["Fatality Rate"])
```

fatality_rate_df.to_excel(writer, sheet_name="Fatality Rates", index=True, header=True)

self.labelArea.setText(f"Matrices, results, and fatality rates successfully exported to {filename}.")

except Exception as e:

self.labelArea.setText(f"Error saving matrices, results, and fatality rates: {e}") logging.error(f"Error saving matrices, results, and fatality rates: {e}")