Master's thesis

NTNU
Norwegian University of Science and Technology
Department of Geography

Torgeir Skurdal

Developing semi-automated GIS methods for indicating downstream flood damage potential of critical points

Master's thesis in Geography
Supervisor: Chantel Nixon
Co-supervisor: Knut Alfredsen
May 2022

NTNU
Norwegian University of
Science and Technology

Torgeir Skurdal

Developing semi-automated GIS methods for indicating downstream flood damage potential of critical points

Master's thesis in Geography
Supervisor: Chantel Nixon
Co-supervisor: Knut Alfredsen
May 2022

Norwegian University of Science and Technology
Department of Geography

NTNU
Norwegian University of Science and Technology

Abstract

Surface water floods are already a huge problem affecting many Norwegian municipalities and predictions for increasing temperature, annual precipitation, and extreme weather events means that flooding will continue to be problematic for the foreseeable future.
Regional forecasts are issued by Norwegian Water Resources and Energy Directorate (NVE), but they are very general. One of the main goals of Trygg Elv is to develop methods for converting these regional forecasts, to better the understanding of what they mean on a local scale. Developing more accurate methods for small watersheds is important, to help protect vulnerable infrastructure and keep inhabitants safe. This study has developed, tested, and evaluated methods for indicating the damage potential of critical points in a small watershed in Melhus municipality by developing a GIS-based models for automating the process of water delineation. Critical points are natural or man-made obstacles or constrictions where the deposition of sediments, driftwood and other debris may result in the stream or river changing course. The model is set up to run two different scenarios: critical points blocked, and open flood paths. In this study the damage potential is assumed to be directly correlated to the upstream contributing area of a given point in the watershed. By finding the contributing upstream area, it is possible to make assumptions about the damage potential of a given location. This can help improve preparedness in flood-prone areas, as well as point to locations that should be given extra attention and locate areas where flood protection measures should be considered.

Sammendrag

Overvannsflom er allerede et stort problem som påvirker mange Norske kommuner, og prognoser spår økende temperatur, årlig nedbør og ekstremvær gjør at flom vil fortsette å være problematisk i overskuelig fremtid.
Regionale varsel gis av Norges vassdrags- og energidirektorat (NVE), men disse er ofte svært generelle. Et av hovedmålene til Trygg Elv er å utvikle metoder for å konvertere disse regionale prognosene, for å bedre forstå hva de betyr på lokalt nivå. Utvikling av mer nøyaktige metoder for små nedbørfelt er viktig, for å bidra til å beskytte sårbar infrastruktur, og sikre menneskeliv. I forbindelse med denne oppgaven har det blitt utviklet, testet og evaluert metoder for å indikere skadepotensialet på kritiske punkter i et lite nedbørfelt i Melhus kommune ved å utvikle GIS-baserte modeller for automatisering av vannlinjeberegning. Kritiske punkter er naturlige eller menneskeskapte hindringer eller innsnevringer hvor avsetning av sedimenter, drivved og eller andre flom materialer kan resultere i at bekken eller elva endrer løp. Modellen er satt opp for å kjøre to forskjellige scenarier: kritiske punkter blokkert og åpne flomveier. I denne studien antas skadepotensialet å være direkte korrelert til oppstrøms areal. Ved å finne det medvirkende oppstrømsområdet er det mulig å gjøre antakelser om skadepotensialet på en gitt lokasjon. Dette kan bidra til å bedre beredskapen i flomutsatte områder, samt peke ut punkter som bør vies ekstra oppmerksomhet og lokalisere områder hvor flomsikringstiltak bør vurderes.

Forord

Denne masteroppgaven er skrevet i forbindelse med faget «GEOG3900» som er det avsluttende faget på masterprogrammet i Geografi ved NTNU. Oppgaven teller 60 studiepoeng, og er skrevet som en del av Trygg Elv-prosjektet, som er et prosjekt ved institutt for bygg og miljøteknikk på NTNU.

Hoveddelen av denne oppgaven har bestått av utvikling av sammensatte modeller i ArcGIS Pro for indikasjon av nedstrøms skadepotensial et kritisk punkt kan forårsake i det tilfelle at det blir tilstoppet av sedimenter eller drivgods. Prosessen har til tider vært utfordrende, med lange dager og mange sene kvelder fremfor datamaskinen. Til gjengjeld sitter jeg igjen med verdifulle lærdommer og erfaringer med Geografiske Informasjonssystemer som jeg nå i etterkant ikke ville vært foruten.

Jeg vil rette en stor takk til Chantel Nixon som har vært min veileder på denne oppgaven og bidratt med viktig hjelp og råd under oppgaveskrivingen. Videre vil jeg takke Professor Knut Alfredsen, Professor Oddbjørn Bruland ved institutt for bygg og miljøteknikk, og tidligere Ph.d. student Ana Juárez Gomez for verdifull bistand og faglig rådgivning. Til slutt vil jeg takke kjæresten min Martine, som har holdt ut med meg gjennom denne prosessen, og alltid gir meg støtte og oppmuntring når jeg trenger det mest.

Trondheim,15. Mai 2022

Torgeir Skurdal

Table of content

List of figures:

List of tables:

Abbreviations:

ALS → Airborne LiDAR scan

DEM - Digital Elevation Model

ESRI→ Environmental systems research institute

FKB → Felles Kartdatabase (Common map database)

GIS→ Geographical Information System

GPS → Global positioning system

HEC→ RAS - Hydrologic Engineering Center's River Analysis System

IMU → Inertial measurement unit

LAS → Log Ascii Standard

LGM → Last Glacial Maximum

LiDAR → Light detection and ranging

NOU→ Norges offentlige utredning (Official Norwegian Reports)

NVE → Norges vassdrags og energidirektorat (Norwegian Water Resources and Energy Directorate)

NVDB → Nasjonal VegDataBank (National Road databank)

TIN → Triangulated Irregular Network

UTM → Universal Transverse Mercator

VA → Vann og avløp (water and wastewater management)

1. Introduction

1.1 Trygg Elv (Safe River) project

In our changing climate, an emerging challenge is to maintain strong and resilient infrastructure that can handle more frequent flooding and manage floodwater when streams and rivers overflow their banks. With increased temperatures, more extreme weather is expected, which will lead to increased rainfall, in the future (Myhre et al., 2019). Smaller watercourses, rivers, and streams with a watershed smaller than 50 km^2 in accordance with Norges Vassdrags- og Energi-direktorat (NVE) Guide no. 7-2015 (Stenius et.al, 2015) have a short response time to torrential rainfall events and therefore have the potential to cause substantial damage in a short period of time. During flash floods, rivers have higher velocities and can overflow their banks, potentially causing catastrophic damage via enhanced erosion, transportation, and as floodwaters recede, deposition of sediments. For example, the village of Kvam, Nord-Fron in eastern Norway, experienced a devastating flood in 2013 which caused massive damage to roads, buildings, and other infrastructure. As did the village of Utvik, Stryn in western Norway in 2017 (Figure 1.1).

Hydrological dimensioning, which is the sizing of flood protection measures, such as ditches, culverts, and bridges to prevent flooding, can be challenging in small watersheds, as the available data often are intended for modeling larger watercourses. The watershed is the total contributing area leading water to outlet. Furthermore, many small rivers and streams do not have measuring stations and data are often extrapolated from similar rivers or interpolated from stations nearby. NVE issues flood warnings for larger watercourses, but there is often great uncertainty about exactly where in the region the precipitation will fall. This means that even the best methods, (such as NVE, 2015; “Veileder for flomberegninger i småuregulerte felt”) for flood estimation in small, unmeasured watersheds are associated with great uncertainty. Due to the major damage that small, steep rivers and streams can cause over such short timespans, developing accurate flood modeling methodologies for these streams and rivers will be an important focus area in the future (Bruland, 2020).

To support the development of better tools for flood estimation in small watersheds this thesis will focus specifically on "critical points" in watercourses. Critical points are natural or man-made obstacles or constrictions where the deposition of sediments, driftwood and other debris may result in the stream or river changing course (NVE, 2015). Technical interventions and natural conditions that can lead to flooding with increased water flow, where technical interventions are defined as bridges, culverts, ditches, closed streams, and other man-made features that narrow the watercourse, thereby reducing its capacity (NVE, 2015). Natural conditions that may lead to flooding including narrow points in the watercourse, points and stretches prone to erosion, shallow areas sensitive to mass deposition, portions of the stream that run higher than the surrounding terrain, vegetation in or near the watercourse, and areas with ice problems (Stenius et al, 2015). This thesis will investigate critical points and their potential to cause downstream damage by developing a ModelBuilder model with ArcGIS Pro tools. The model is set up for Kaldvella, a small river in Melhus municipality in Trøndelag, Norway (Figure 2.1), but is constructed in such a way that it can be applied to other rivers and streams with a minimum of modifications. Mapping of critical points will contribute to better preparedness in flood-prone areas and quickly locate sites where flood protection measures should be considered.

Extreme rainfall and floods present a major challenge to society in terms of damage to property and critical infrastructure such as power lines, railways and roads. Floods can cut off important societal services such as electricity, clean water, healthcare, and communications. As a result of climate change, extreme weather events are expected to occur with greater frequency and intensity in the future. According to the Norwegian Natural Damage Pool, damage caused by extreme weather events and natural hazards cost ~ 28 billion kroner over the last 10 years (Finans Norge, 2022, p.6). The World Economic Forum counts extreme weather and natural disasters as a result of climate change as the largest risk factor both in terms of probability and impact (Bruland, 2020). In Norwegian municipalities, risk and vulnerability analyses are needed to identify critical areas with respect to flooding due to extreme rainfall events, however there is a lack of instruments and tools to carry out such analyses (Kielland, 2020). As well as a backlog when it comes to infrastructure maintenance (Bruland, 2020). Furthermore, national resources for securing watercourses are limited, Kalsnes et.al (2021) estimates that the cost for securing all buildings residing within hazard zones for floods, landslides and quick clay will cost ~85 billion NOK to make them as safe as the demands for new buildings in TEK17. While at the same time the total budget for flood preventive measures including buildings, rail, and roads, are limited to 720 million NOK in 2022 (Olje- og energidepartementet, 2021). It is therefore important to prioritize measures that maximize cost/utilitarian value.

One of the goals of the Trygg Elv project is to develop methods and tools that will quickly identify parts of watercourses where the payoff of measures will have the greatest impact. As well as to give indications of where it is most critical to take preventive measures, where the river can find new paths, and the potential consequences of this. Using hydraulic models, the project will identify topographical factors that, combined with different precipitation scenarios, could create critical situations in terms of flooding, damage, and risk to people. In a connection with a Geographical Information System (GIS), such information will be used to identify potentially critical areas in a watercourse. Critical areas can be thought of as areas that the municipalities or infrastructure owner should pay extra attention to in the case of a flood. By combining models, geographical data, and field observations with local knowledge, this can help prevent situations that can develop into hazardous situations, endangering critical infrastructure or human lives and thus prevent the worst-case scenarios.
NVE have established plans where municipalities can apply for grants to map critical points in streams and steep watercourses (NVE, 2018). Based on the events of recent years, flooding in even the smallest streams has enormous potential for damaging buildings, infrastructure and endangering people. (Bruland, 2020)

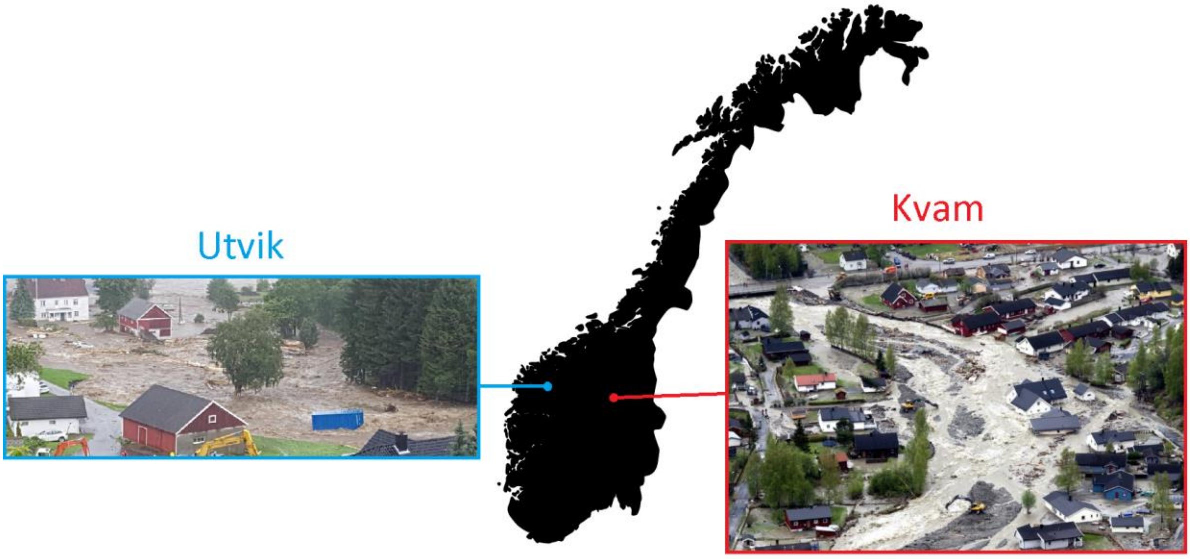


Figure 1.1: showing the damages caused in Utvik in Stryn and Kvam in Nord-Fron.

Even in the smallest municipalities of Norway, there are so many streams, manual mapping is very difficult and extremely time-consuming. Even with good instructions for how this is to be carried out, such surveys require expertise that municipalities do not presently have. The aim of this thesis is therefore to develop a methodology for identifying the downstream damage potential of critical points. This will be achieved by combining hydraulic modeling output, and geographical data such as building data, road data and results from geographical analysis together in a GIS. Through hydraulic modeling, it will be possible to identify different terrain shapes and terrain types to investigate what kind of effects they have in a flood situation and how the topography affects flood watercourses. With this information, it will be possible to develop routines in the municipalities' GIS systems that identify localities that should be given extra attention during extreme rainfall events. The models developed in this study will also be useful in assessing the effectiveness of preventive measures and assessing damage reduction measures and possible evacuation plans in a crisis.

In the Trygg Elv project, a selection of small watercourses has been made in collaboration with Melhus municipality, which will be investigated and mapped. Kaldvella, Gaula and Varmbubekken are all small watercourses that have the potential to do great damage if several favorable flood triggering factors should coincide. This study will investigate Kaldvella (Figure 2.1).

1.2 Earlier studies

There is a large body of research on hydrological modeling in GIS, investigating how DEMs can be used to model water flow in river and stream channels e.g., Bratlie (2013), Sellægg (2016) and Kielland (2020) to mention a few. A variety of different programs, toolsets and flow algorithms have also been developed, these will be presented in Chapters 3 and 4.

Because spatial and temporal variation of torrential rain is, complicated to forecast within any catchment. This makes predicting critical areas during any given flood difficult.
In addition to increased focus on flood related issues by the municipalities, has there been a collective effort by Norwegian government departments to develop new, and better methodologies for mapping flood paths. One such project is called Naturfarer, Infrastructure, Flom og Skred (NIFS; Natural hazards, Infrastructure, Flood and Landslides), which was a collaborative project with NVE, Jernbaneverket (Norwegian railway) and Vegvesenet (The

Norwegian Public Roads Administration). The aim of NIFS was to investigate, evaluate and develop methodologies for handling natural hazards and evaluate their performance. The project was based on comparing the models produced in the project with damage data from the floods that impacted Gudbrandsdalen in 2013 (Viréhn, 2014). Gudbrandsdalen is a large valley that crosses south-central Norway from SE to NW. The Department of Geography, NTNU, also contributed to NIFS with several master theses. Some of these are Viréhn (2014), investigated damage points and determined if the damage was a result of streams and rivers growing beyond their banks, or drainage features being clogged, and if this could be automated in a script.
Henning (2015) Utilized Geographical Information System (GIS) to model flood scenarios using two different flow algorithms. In addition to this, methods for making flood maps using fuzzy logic overlays were tested. A similar study was made by Sellæg (2016) investigating the June 2014 flood in Lillehammer, looking at three different rivers, and comparing the model results with known damages registered by the municipality. Sellæg (2016) based his analysis on two scenarios, open and closed drainage features. Kielland (2020) developed a GIS-tool for stormwater calculations, for use in early-stage planning. The tool was designed to work with few input values. Despite few input values the tool has great potential for producing information about the watershed characteristics, as well as stormwater peak flow for different recurrence intervals. For Kaldvella, in Melhus municipality, Graf (2021) created a hydraulic model to investigate critical points in using ArcGIS and HEC-RAS to investigate flood scenarios of different magnitudes in Kaldvella. Graf (2021) produced flood maps by setting up a hydraulic model for parts of the Kaldvella watershed. These flood maps will be used to compare the output from the tools produced in this study. To supplement the research by Graf, Gomez (2021) set up her own hydraulic model in HEC-RAS, these maps will also be used to validate the output from this study.

In addition to the ones mentioned above, there is a list of reports of older and new dates, such as Solberg (2018) investigating the geology and landscape development along the Gaula river. A risk and vulnerability analysis by Melhus municipality (2016), and a rapport evaluating flood and ice problems in Kaldvella form 2016. The latter two are too old to be interesting for this study, since the terrain has been changed. The main takeaway from the reports is that Kaldvella was sensitive to flooding due to several reasons, such as erosion, heavy rainfall and ice problems in 2016 (Husebye, 2016).

1.3 Research questions

Given the unpredictability of torrential rain and the challenges in mapping the numerous small rivers and streams and their critical points within most municipalities in Norway. This thesis aims to address these challenges for one small river, Kaldvella in Melhus municipality (Figure 2.1), by asking the following questions:

1. *Can GIS methodologies be applied to indicate downstream damage potential of critical points in a small watercourse?*
2. *Is it possible to automate this process in an ArcGIS ModelBuilder workflow?*
3. *Can the model output be used to evaluate if flood protective measures are needed?*

When assessing surface water, a GIS tool is typically used to calculate, among other things drainage lines and low points. ArcGIS Pro is one of several software programs that can be used for such applications, which is elaborated on in Chapter 3. ArcGIS Pro contains several built-in tools and functions that can calculate different parameters, most of which can be automated and modified in a script or ModelBuilder workflow. This study will utilize this to make model builder workflows to ease this process.

The overall goal of this thesis is to develop ArcGIS Pro toolsets, which can be used to automate GIS analysis, and help indicate sensitive areas, and where flood protection should be considered, maximizing cost-utility factors.
A secondary aim is to develop models that provides clear visualization of where surface water management is most needed, while trying to minimize the effort from the user. Emphasis is also placed on documenting the automations of calculations that the tool implements but is not the main focus of the study.

Finally, an assessment will be made to evaluate the advantages and disadvantages of the toolsets in relation to other existing products, made for providing information in early-stage assessment. This will help test and examine the usefulness of the tools developed.

1.4 Study limitations and preconditions

The hydrological analysis presented in this thesis is based on the assumption that surface water flow is controlled by gravity and the features in the terrain. This means that the modeled flood paths are explicitly based on the terrain features and sloping of the digital elevation model. Parameters like rainfall, erosion and infiltration to the ground can be included as weight rasters and are not modeled separately. An elaboration on weight rasters can be found in Chapter 4. It is also important to note that the tools used in the thesis as well as the models developed do not present results with the same accuracy as a hydraulic model. The tools and workflows developed in this thesis are intended for planning and to investigate what will happen when critical points are blocked, as well to give an indication of potential damage that can be caused by critical points. The tools developed in this thesis also operate on the assumption that only one critical point is blocked at one time; multiple critical points being closed at the same time should be considered implemented in the future.

1.5 Thesis Structure

The thesis is structured into eight chapters with associated sub-chapters.
Chapter 1 -The introduction presents some general information on floods and flood damage. **Chapter 2** presents the Gauldalen region, and a detailed description of the Kaldvella watershed. **Chapter 3** Presents the background for this thesis, looking at floods, flood management, and some existing tools. **Chapter 4** Presents the theoretical background for the thesis, presenting Geographical information systems, their associated formats. This chapter also present some tools fundamental to this study. **Chapter 5** explains the methodology, this mainly consists of how the development of the tool in ArcGIS Pro was carried out and tested. **Chapter 6** presents the results, from running the tools and discussing their output. In **Chapter 7** the model and its output are discussed, and the performance is evaluated. **Chapter 8** presents the conclusion of the thesis, in addition to discussing some further work.

2 Study area

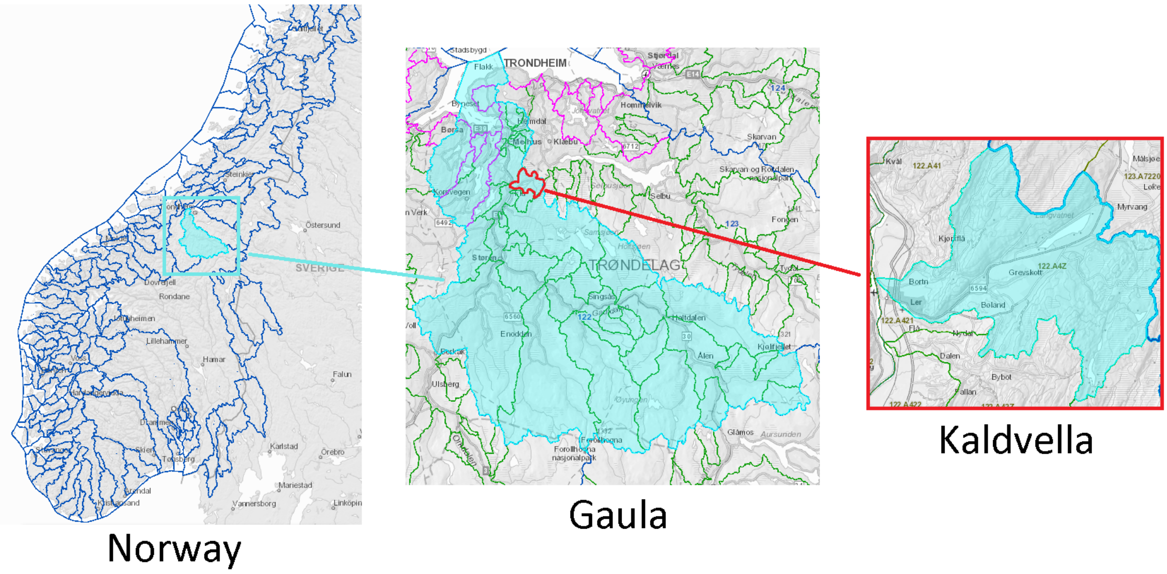


Figure 2.1: illustrates the geographical placement of the study area.

2.1 Gauldalen

Gauldalen is a river valley in Trøndelag County that stretches from Rørosvidda in the east to Gaulosen and the Trondheimsfjord in the west (Fig. 2.1). The river Gaula, which is ~145 km long, flows through the valley. Gauldalen is characterized by a complex topography, much of which falls below marine limit (the highest limit of the marine boundary following the last glacial period). The valley is therefore shaped, carved and eroded by a number of processes, both glacial, glaciofluvial, and fluvial, which at different times have influenced valley topography. This will be discussed in further detail in Chapter 2.3: Geology and landscape development. Europavei 6 (Car) and Dovrebanen (Train), are both central transportation routes in the region, running through the valley (Thorsnæs, 2021).

Trøndelag county is characterized by a varied climate due to its coastal and mountainous inland environments. In the low-lying coastal areas, most floods, including the largest events, occur in autumn and winter. The larger watercourses, with watersheds stretching to mountainous areas, experience spring floods because of snowmelt. In addition, there are many smaller watersheds where floods can potentially occur at any time throughout the year, because of heavy rainfall (Dannevig, 2020).

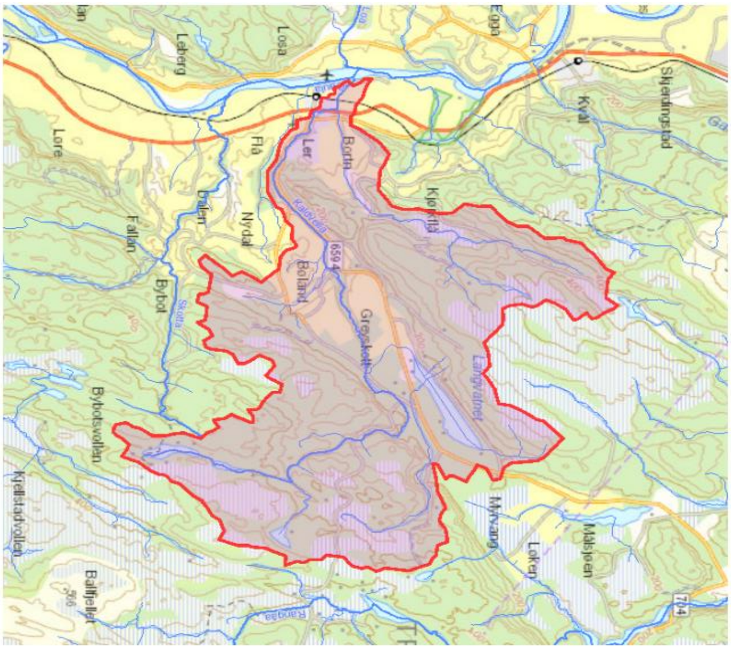
2.2 Kaldvella

This thesis will examine Kaldvella which is a stream that flows through the town of Ler in Melhus municipality (Figure. 2.1). The stream was strategically selected in collaboration with Melhus municipality which is part of the Safe River project. Kaldvella has a watershed of 29.7 km² and therefore falls within the definition of small watersheds (50 km²) in accordance with NVE's Guide no. 7-2015. Guide for flood estimation in small unregulated fields from 2015 (Stenius et. Al., 2015). The field has a total area of 29.7 km², and a river length of 12.7 km. as seen in Figure 2.2.

Previous studies and reports such as Graf (2021), Husebye (2016) and the risk and vulnerability analysis performed by Melhus municipality in 2016 (Melhus kommune, 2016) have shown that Kaldvella is prone to flooding for a number of different reasons including, low infiltration, frequent blocking of the river channel due to erosion, flood debris or anchor ice in winter. Anchor ice is ice that forms on the river bottom during the winter, causing the river to freeze from the bottom up as supercooled water hits solid particles causing it to crystallize. The formation of anchor ice has great potential to block or lower the capacity of the stream channel and cause the water to take alternative paths (Halleraker, 2022).

Stickler, Alfredsen, Linnansaari & Fjeldstad (2010) found that River ice can have a profound effect on in-stream conditions and that ice processes should be considered in stream management and assessment. As a result of anchor ice formation stream environments are transformed from fast-flow to slow-flow, even on short temporal scales. Furthermore, the anchor ice build-up-initiated ice formation due to reduced local water velocities causing ice dams upstream.

In NEVINA it is possible to calculate precipitation fields, field parameters and indices, based on a freely chosen location in a watershed (NEVINA, n.d). In this study the chosen point is the location where Kaldvella outlet into the Gaula river (Figure 2.2).



Norges vassdrags- og energidirektorat

NVE

Kartbakgrunn:	Statens Kartverk
Kartdatum:	EUREF89 WGS84
Projeksjon:	UTM 33N
Beregn.punkt:	263445 E 7016812 N

Nedbørfeltgrenser og feltparametere er automatisk generert og kan inneholde feil. Resultatene må kvalitetssikres.

Nedbørfeltparametere

Vassdragsnr.:	**122.A4Z**
Kommune.:	**Melhus**
Fylke.:	**Trøndelag**
Vassdrag.:	**Kaldvella**

Feltparametere		
Areal (A)	29.7	km²
Effektiv sjø (A_{SE})	0.44	%
Elvleengde (E_L)	12.7	km
Elvegradient (E_G)	30.5	m/km
Elvegradient$_{1085}$ ($E_{G,1085}$)	38.0	m/km
Helning	10.7	°
Dreneringstetthet (D_T)	1.3	km^{-1}
Feltlengde (F_L)	8.4	km

Arealklasse		
Bre (A_{BRE})	0	%
Dyrket mark (A_{JORD})	10.1	%
Myr (A_{MYR})	6.8	%
Leire (A_{LEIRE})	11.8	%
Skog (A_{SKOG})	75.9	%
Sjø (A_{SJO})	2.5	%
Snaufjell (A_{SF})	0	%
Urban (A_U)	1.2	%
Uklassifisert areal (A_{REST})	3.5	%

Hypsografisk kurve		
Høyde$_{MIN}$	21	m
Høyde$_{10}$	127	m
Høyde$_{20}$	173	m
Høyde$_{30}$	194	m
Høyde$_{40}$	242	m
Høyde$_{50}$	292	m
Høyde$_{60}$	323	m
Høyde$_{70}$	357	m
Høyde$_{80}$	394	m
Høyde$_{90}$	425	m
Høyde$_{MAX}$	525	m

Klima- /hydrologiske parametere		
Avrenning 1961-90 (Q_N)	20.4	l/s*km²
Sommernedbør	410	mm
Vinternedbør	538	mm
Årstemperatur	3.7	°C
Sommertemperatur	9.8	°C
Vintertemperatur	-0.6	°C

Figure. 2.2: watershed parameters for Kaldvella, retrieved from NEVINA.

2.3 Geology and landscape development

Landscape development and the surficial geology of Gauldalen have been most influenced by glaciation, deglaciation, and post-glacial relative sea-level change during the Quaternary period. The Quaternary period (last 2.6 million years) is characterized by extreme climatic changes, with over 40 glacial-interglacial cycles (Elias, 2013). Approximately 20000 years before present, the Fennoscandian Ice Sheet reached its maximum extent, stretching south to Denmark and northern Germany. This is referred to as the Last Glacial Maximum (LGM). It is estimated that the ice thickness reached over 3000 meters. At the time of the LGM ~5.5% of the worlds water was bound up in ice, compared to ~ 1.7% today (Steffen & Wu 2011). The heavy ice sheet compressed the Earth's crust, and the lag in glacio-isostatic rebound (uplift of the crust) following deglaciation around 15000 years before present allowed the sea to flood the coastal areas up to around 200 m above modern sea level. The highest shorelines formed during this time mark what is referred to as the marine limit. Glacio-isostatic rebound of the land is ongoing in Norway (Rohr-Torp, 1994).

The surficial geology of Gauldalen also includes large amounts of marine clay, deposited at the same time as the deltas, but in the deeper waters of the palaeofjord (Figure 2.3). Marine limit in this area is approx. 175-180 meters above sea level (Solberg & Hansen, 2017; Reite 1983, 1985). The resulting landscape can be described as complex, as marine, glacial, glaciofluvial and fluvial processes at different times have left their mark on the landscape.

Just like the rest of Gauldalen, the Kaldvella watershed consists of a mixture of sediment types, as illustrated in Figure 2.4. Marine sediments underlie the town of Ler as it is below the marine limit. Looking at the quick clay maps made by NVE indicate that quick clay fields may be present. Further up in the watershed near Lake Langvatnet, the surficial geology is dominated by glaciofluvial deposits, more specifically, an enormous, ice-contact marine limit delta. At higher elevations above marine limit, the watershed is dominated by thin moraine material overlying bedrock. For fluvial mass transport to happen, there needs to be erodible material available. This is very much the case in Kaldvella, and in the case of extreme precipitation events, these materials will be eroded, transported, and deposited downstream.

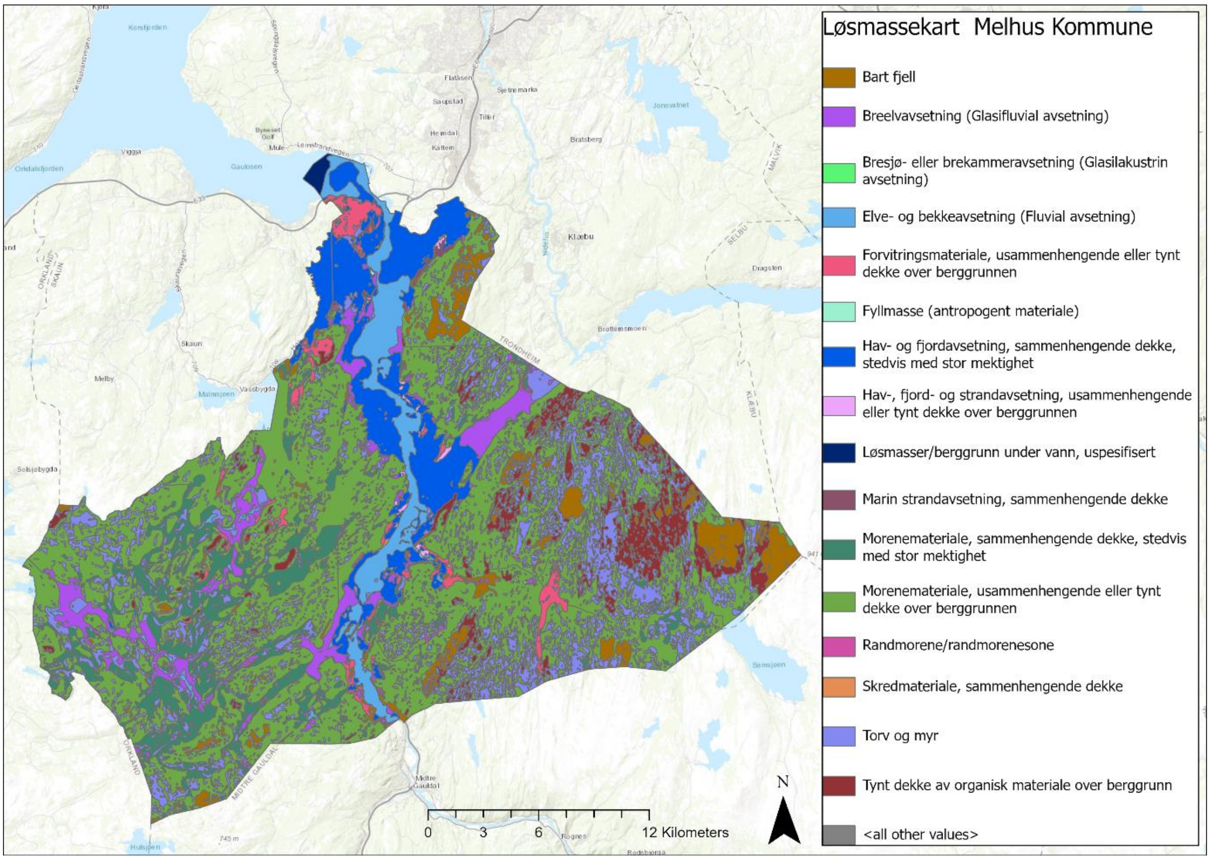


Figure 2.3: Map of quaternary deposits in Melhus.

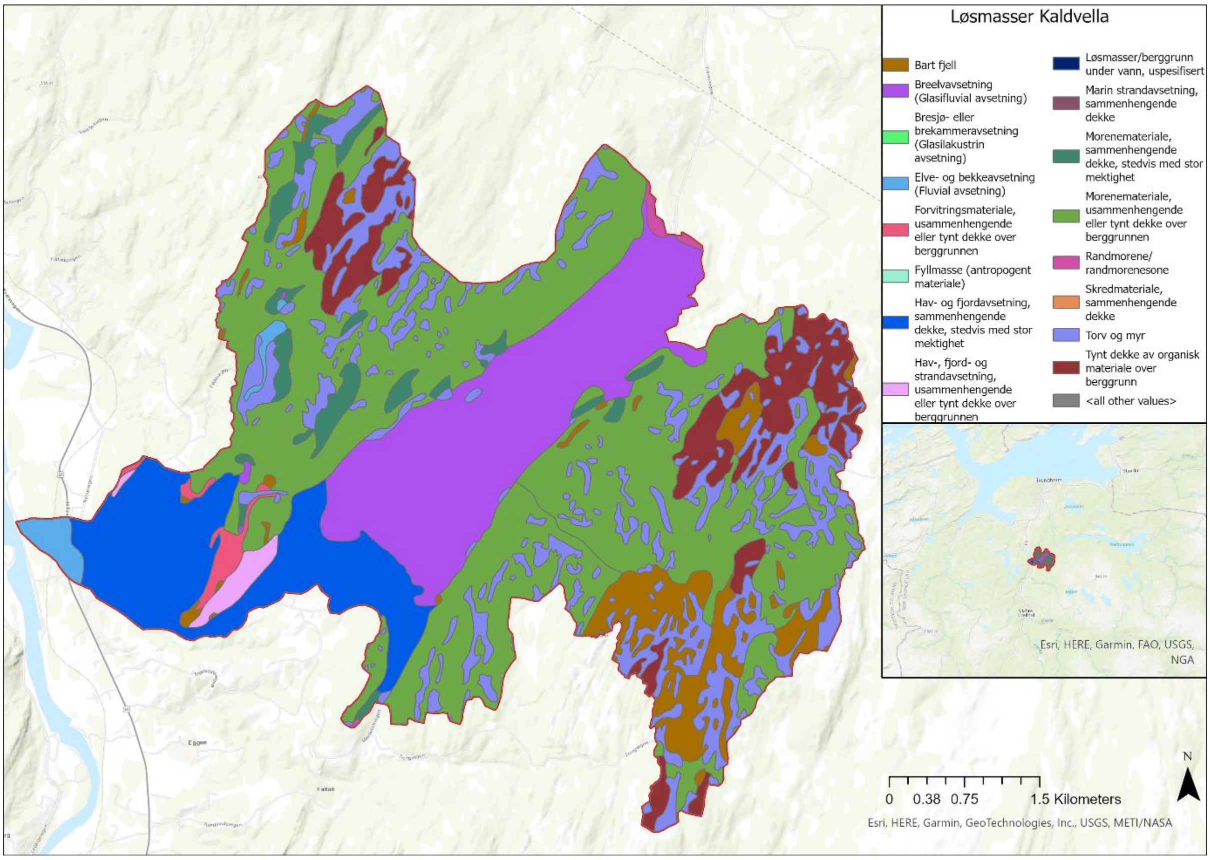


Figure 2.4: Map of quaternary deposits in Kaldvella.

The bottom of the Kaldvella watershed is characterized by ravines. These are common features all through Gauldalen. Ler is primarily underlain by marine sediments, the uppermost of which are glaciofluvial sands, gravels, and cobbles reworked by coastal processes when sea level was higher. The sediments were transported by glacial rivers and deposited into the then fjord during the ice melt of the last ice age. During the postglacial glacio-isostatic adjustment (uplift of the land), the rivers and streams incised down through the marine sediments towards a falling base level. Evidence for this includes ridges and fluvial terraces on both sides of the Gauldalen valley. Smaller rivers and streams have eroded these ridges and terrasses and ravines of varying size can be seen all throughout the landscape (Stusdal, 2005).

2.4 Climate

One of the most dramatic effects of ongoing climate change in Norway so far (and expected to increase in the future) are floods and extreme precipitation events. In recent years Norway has seen many examples of such events. Examples of this are Kvam 2013, Utvik 2017, Skjåk 2018, Estimated to be a flood above 50-year return interval (Skjåk kommune, 2021). The Flood event in Kvam, 2013 is estimated to be a 50–100-year flood (Roald, 2015)
see figure 1.1 above for reference to get an understanding of the damage. There are uncertainties surrounding the flood in Utvik and is estimated to be between ~300 year and 1000+ year flood (Leine, 2017).

Flash floods can severely damage buildings and infrastructure such as roads and railways due to their erosive power and because they are difficult to predict in time and space and it is therefore difficult to know where to best place flood protection.
With the effects of climate change the average temperatures are expected to increase in Norway. Climate change is also predicted to increase the average temperature in Norway by around 4.5 degrees by the year 2100, with the largest temperature increase expected during winter months. As a result of rising temperatures, annual precipitation, and the number of days with heavy precipitation are also expected to increase. For annual total precipitation, an increase of 18% is predicted for the end of the century, and days with heavy rainfall are expected to increase by 19%. The increase in heavy rainfall events with durations shorter than one day can be significantly greater (Hanssen-Bauer et al., 2015). Increased rainfall in general will lead to more frequent cases of floods and landslides. The risk of landslides

depends on the local terrain conditions, but the weather can act as a catalyst, especially in steep terrain and small watersheds where the response time is fast (Kalsnes et.al, 2021). Intense precipitation events can increase both frequency and extent of landslides, floods, and debris flow. The outcome is that the climate of the future will include greater rainfall, more snowmelt, more floods, and more landslides and avalanches (Norwegian Environmental Agency, 2021).

Increased erosion because of more frequent and larger rain floods can trigger quick clay landslides, but there is a high degree of uncertainty about the degree to which erosion contributes to accelerating quick clay slides. Increased runoff and stray water could also lead to an increased risk of erosion and sediment transport, small fields are particularly sensitive to this. Erosion and sedimentation will be at least as great of a challenge as the flood water as sediment-laden floodwaters often lead to an increased risk of clogged drainage. An increase in floods in precipitation dominated rivers is expected, due to increased probability of cloudburst. Rivers that are dominated by snowmelt will generally have floods that decrease in size over time. For the large inland waterways dominated by snowmelt, such as Gudbrandsdalen and Østerdalen, a reduction in spring floods of up to 50% is expected. On the other hand, watercourses dominated by rain floods could expect an increase of up to 60% (Hanssen-Bauer et al., 2015). Kaldvella is a small river and falls into the first category (NVE 2016).

3. Background

3.1 Floodtypes

Norway is dominated by two types of floods: meltwater floods and rainfall floods. Snowmelt in combination with precipitation is dominant for winter and spring floods, while torrential rain is the driving factor for autumn floods (Stenius, Glad & Wilson, 2014). The large river floods in Norway are mainly caused by rain, often in combination with snowmelt in the spring season. Meteorological conditions, such as the distribution of precipitation in time and space, together with temperature have the greatest significance for the size of the flood, but the storage capacity of the affected watershed, which is affected by topography, vegetation, ground conditions, lake percentage, and rivers and streams, is also important. In general, the size of the watershed will be of great importance for the intensity, which is runoff per unit area. This means that small fields often have greater intensity than larger fields. In steep fields the runoff is faster than in flatter fields. Fields that are dominated by hard surfaces such as bedrock or a high degree of urbanization will have dense surfaces which further results in both runoff happening faster and that a smaller proportion of the water will infiltrate the ground, compared to fields dominated by bogs or forests, where there are large amounts of loose material in the ground. Lakes in the watershed can have a flood-reducing effect, if they are connected to the main watercourse in the field, otherwise the lake percentage in the field will have little effect (NVE, 2022).

The condition of the watershed will also have a significant impact. If water flows on water-saturated snow or frozen ground, this will increase the risk of flooding in the case of a heavy precipitation event. The same applies to water-saturated soils. If the ground is dry, empty water reservoirs will have a braking effect, provided that the precipitation intensity does not exceed the infiltration capacity (NVE, 2020).

3.1.1 Floods and recurrence intervals.

Flood sizes are usually stated with or annual probability, also referred to as the recurrence interval. The recurrence interval indicates how often a flood of the same magnitude occurs on average over many years. A flood with a recurrence interval of 200 years, also called a 200-year flood, occurs on average every 200 years. Each year, the probability of a 200-year flood is equal to 1/200 (0.5%). This does not exclude two 200-year floods from recurring at shorter intervals. Calculation of recurrence intervals for floods is based on historical observations (Direktoratet for byggkvalitet, n.d.).

3.2 Floods in small watercourses

In small watercourses, floods can develop quickly, and the response time can be very fast, and, in some cases, there are only a few hours between downpour and flood peak. Floods in smaller watercourses have the potential to cause major damage as a result, even though precipitation falls over a limited geographical area. These floods are problematic to predict and pose a potential threat to both human lives and critical infrastructure. Floods can occur when damming culverts and low-hanging bridges, as well as closed streams get clogged. Problems arise when culverts and closed streams are not dimensioned for this type of rainfall, and the capacity is exceeded. This will cause them to be easily blocked by eroded material,

propellants, or driftwood. Impaired capacity due to foreign bodies is normal. When snow melts in the winter, ice can also clog culverts and closed runs.
In all rivers where one finds loose materials such as clay, sand, gravel, and rock in the riverbed and at the edges, the water will erode and transport the masses down the watercourse. The loose materials are deposited either on flat sections in the river course itself, on the floodplain, or in outlet areas where the river hits still or lower energy water, such as lakes or larger rivers (NVE, 2020).

All human interventions can lead to changes in drainage conditions and drainage direction in and between watersheds. Human interventions include culverts, bridges, stream closures, sealing surfaces, e.g., asphalting, because of increased urbanization in the area, or forests being converted to agricultural land. Construction of forest roads, clear-cutting and drainage of wet areas (bogs and other wetlands) will also have a destabilizing effect on natural drainage, increasing the risk of flooding, mass displacement and erosion. If the capacity for drainage measures is not sufficient, the stream will find alternative flood routes. When handling surface water NVE recommends keeping streams open rather than laying them in pipes. If open stream is not an option, the floodwater should be provided with an alternative "safe" flood path. If an alternative flood path does not exist, the culvert should be dimensioned, to be able to safely remove the water (NVE, 2008).

3.2.1 Surface runoff floods

Surface runoff floods are direct runoff over hard surfaces after heavy rainfall, which often lead to local floods, especially in densely populated areas. Surface water occurs when the precipitation intensity surpasses the infiltration capacity of the soil. Surface runoff floods are often characterized by impermeable surfaces such as roads, sidewalks, and roofs. They often coincide with errors, damage, or overload of the surface runoff drainage-system. In cold climates such as in Norway, partially permeable areas such as gravel roads, grass slopes and hard-packed agricultural land are especially sensitive to surface water floods when the ground freezes and the surfaces become completely impermeable. In the case of rain on frozen ground, runoff will increase, at the same time as periods of varying frost and thawing increase the probability of damage to or problems with the surface water drainage systems. In line with the Norwegian Water Directive, surface water should be handled as close to the source as possible. This should be done in such a way that the problem is handled in its entirety, so that the problem is not simply moved downstream (NOU 2015:16).

3.3 Early-stage planning

Spatial planning that considers natural hazards is the most important tool for reducing the risk and damages caused by natural hazards. The best and simplest way to prevent damage is to avoid building in areas exposed to natural hazards, however development still occurs in such areas, and it is therefore critical to ensure sufficient control measures are in place.

The NVE guidelines; Flom og skred I arealplaner (2014), describes the natural hazards that can pose a threat to buildings and infrastructure and how these threats should be managed, addressed, and considered in the planning stages. In relation to this a Risk and vulnerability analysis shall be carried out to highlight sensitive areas. NVEs guidelines describe the requirements for how adequate security can be achieved. The guidelines will form the basis for NVE's input and statements regarding area plans. NVE can lay down objections to plans where the danger of floods and landslides are not well enough mapped or not taken attention to, in a satisfactory manner. The safety requirements for buildings in relation to natural hazards are given in TEK10 and are not a part of The NVE guidelines (NVE, 2014).

3.3.1 The three-step strategy

The three-step strategy was developed to reduce or eliminate surface runoff problems and has gradually become the guiding principle for Norwegian planners since its introduction in Norsk Vann in 2008 (Lindholm, 2008). The strategy involves dividing the various measures for surface water management into distinct groups, depending on how much water they can handle and to what extent the surface water can be handled locally, this is illustrated in Figure 3.1. The strategy is most used in urban fields, but with Kaldvella running so close to several houses, which makes flood protection difficult. Bringing in the concepts presented below into the planning can therefore be valuable.The three steps are dependent on the size of the precipitation event rating from low (step 1) to high (step 3).

- Step 1: Impede runoff through infiltration.
    - Step 2: Hinder runoff through flood control structures.
        - Step 3: Ensure safe flood paths to recipients. In this setting the recipient is the waterbody that is the end receiver of the water discharge, these can be lakes, larger rivers, ocean, bogs etc. In the case of Kaldvella the recipient is Gaula.

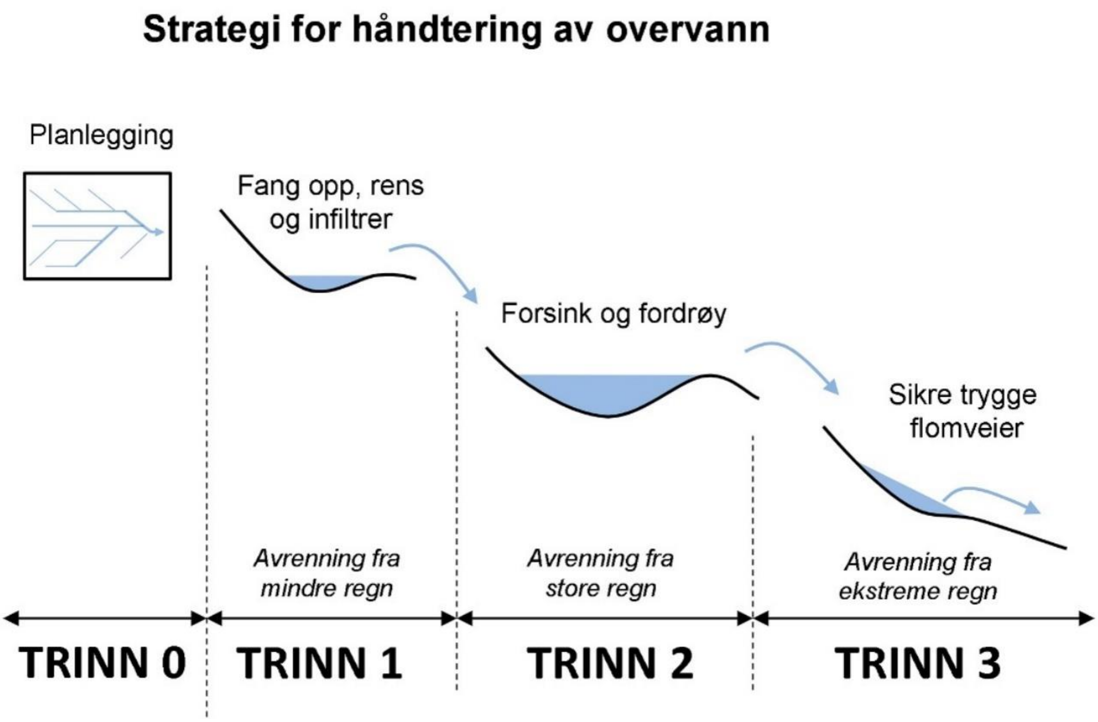


Figure 3.1: Schematic illustration of the three-step strategy (Lindholm, 2008). Modified illustration by (Paus, 2018).

Measures such as green roofs and other permeable surfaces will be able to capture and infiltrate smaller water amounts. During smaller precipitation events, step 1 alone will be able to collect most of the runoff, thus reducing or eliminating the need for steps 2 and 3. The amount of water that is not handled by step 1 is forwarded to flood control features whose function is to hold back and impede the water as long as possible before it is sent on. This will help reduce the flood peaks by lowering the size at the maximum water discharge to the recipient. NVE, 2020 underlines that the field conditions e.g. The number of permeable surfaces, amount of vegetation, steepness in the watershed etc. will be deciding factors, and that these conditions can vary vastly between watersheds, meaning that also the boundaries between the steps is not always clear. To exemplify this, Step 2 is often combined with step 1 in the form of rain beds and infiltration basins where the water is both infiltrated and diverted. Other measures included in step 2 can be diversion reservoirs, open ponds, and controlled floodplains. When the rain event is of such a nature that it cannot be retained by steps 1 and 2, the remaining water should be led away to the recipient via planned flood routes (step 3). NVE, 2020 also points out that the planning often is limited to the build site or planning area, but in most cases the watershed extends well beyond this area, and that this is often not considered before it is too late (NVE, 2020).

3.4 Hydrological modeling with GIS

There are several GIS-based tools that can be used for terrain analysis. Such analyses can provide valuable information in infrastructure planning, as well as visualizing the risk and consequences of surface water flooding. A simple tool is drainage line analysis. The drainage line is a visualization of how the water will flow across the terrain. GIS tools can also be used to calculate the size of watersheds, which is defined as the upstream area that contributes to the defined outlet point. At present there are several tools and programs that can be used to perform these tasks. Some examples of these are Quantum GIS (QGIS) and GRASS GIS. Another tool that was recently launched in Norway is SCALGO Live, which is presented further in Chapter 3.5.1 The platform most widely used however, is Esri's ArcGIS/ArcMap platforms (ESRI, 2015). Esri is the product owner of ArcGIS Pro and the system that is used in this thesis.

One standout weakness of many surface-water models, such as ArcGIS, is that water discharge pipelines are not accounted for. In urban areas this can have a substantial effect, and terrain-based water models, like the ArcGIS, do not account for external factors such as infiltration, evaporation, or the hydraulic properties of the water. To deal with this problem several programs and tools have been developed that operate separately from the flood models. Examples of such tools are MIKE urban, SWMM, and ROSIE. Where ROSIE can be used as an extension in ArcGIS but combining the surface water flow and the pipelines is often complex. MIKE Flood was developed to achieve this task, but it is the most complete and complex modeling program of them all, and requires a very competent user, as well as many input values (Nordeidet, Schow & Killerich, 2015).
Norges Offentlige Utredning (NOU) 2015:16; Surface water in cities and towns (Overvann I byer og tettsteder) only recommends using combined models in areas where hydrological and hydraulic relations are complexly intertwined, resulting in high damage potential and probability of surface runoff.

3.5 Existing tools

3.5.1 SCALGO Live.

The tool most similar to what this thesis is aiming to produce, which is a GIS tool for evaluating downstream damage potential, is probably SCALGO Live. SCALGO Live is a

licensed mapping service developed by the Danish software company SCALGO. The tool was launched in Norway back in 2018 and is one of the newest tools used for flood mapping. The main function of SCALGO Live is to convey surface water
Data, such as drainage lines and water accumulation areas, in a visual format. The analysis is based on the national elevation models from Kartverket (Norway's National Mapping Authority) and is continuously updated.

In the SCALGO Live interface the elevation model can be modified using a range of tools so the terrain data can better represent reality. Features like embankments, terrain drops, or water management features such as culverts and flood banks may be added. The ability to modify the terrain and add or remove features allows one to model future changes in the terrain. SCALGO Live also allows for a quick and clear overview of flood risk in an area (SCALGO, n.d.). However, the analysis is limited and does not account for factors like infiltration, local rain variations, or water drainage pipes. SCALGO Live is also not able to perform hydraulic modeling, but this might be implemented in the future.

3.5.2 HEC-RAS

The hydraulic flood modeling software HEC-RAS is a free, open-source program developed by the United States Army Corps of Engineers (Brunner, 2021). The main function of the program is modeling of channel flow in streams and rivers to analyze flooding. In addition to this the newer versions of HEC-RAS also include the RAS-mapper which is HEC-RAS' built-in function that allows the user to visualize the output data directly in HEC-RAS. RAS-mapper works like any other GIS program and can display the output as graphics.

HEC-RAS can take a variety of different input data and the program can perform a variety of analyses, such as sediment transport, water quality, and steady and unsteady flow scenarios. Analyses in HEC-RAS can also be run in both one and two dimensions. The HEC-RAS model is based on different calculations compared to SCALGO Live and ArcGIS Pro. HEC-RAS is based on a grid format DEM, that allows the user to add breaklines, both manually and from file import. The user can also add profile lines to calculate water depth. Unlike ArcGIS Pro HEC-RAS are based on more advanced algorithms allowing HEC-RAS to take the waters properties into account, as well as depth, water speed, and friction against the terrain (Brunner, 2021).

4. Theory

4.1 Geographical Information Systems (GIS)

Since the 1960s Geographical Information Systems (GIS) have been used in a wide range of fields. It is difficult to make a precise definition of the term GIS due to the range of complexities, formats, and applications, but broadly, GIS is an umbrella term for computer-based tools that store, analyze and visualize geographical information (Rød, 2015, p11-14). Geographical information is data with spatial and sometimes temporal attributes that connect them to a specific location in the world. Spatial attributes include X, Y, and Z coordinates, that is, latitude, longitude, and elevation above mean sea level, and can take on a variety of geometrical shapes, such as polygons, lines, or points (Rød, 2015, p.14-15). Longley et al (2015) underlines that geographical data in reality is “Infinitely complex” (Longley et al, 2015, p.61) while what can be included in a map is not. A map will therefor always be a strategical simplification of reality, only including the elements that the author of the map wants to highlight (Rød, 2015, p.18-19).

4.2 GIS file-formats

In a GIS, geographical data is assigned to two main categories: raster and vector (Fig. 4.1). The raster format is pixel based, while the vector format is based on vertices and a mathematical expression containing its x, y and z values. Geographical data in vector format can further be subdivided into three classes: points, lines, and polygons. As a rule of thumb, point data is best for illustrating discrete features that are too small to be illustrated as lines or polygons, features that have no physical dimensions, or represent “abstract” locations such as place names. Line features are best for illustrating linear features. These are features that have no or very small surface area, such as roads, railways, or powerlines. Polygon features are objects that have large surface areas, such as forests, farmland, or larger water features such as lakes. Polygon format is mostly used to illustrate two-dimensional features (Longley et al. 2015, p. 68).

Raster data is pixel based, where each pixel is assigned a value. Raster data can be divided into integer or float type rasters which means that the raster cells contain either whole or decimal values. Raster format is mostly used to illustrate continuous data without clear boundaries, such as elevation, precipitation, or temperature (Longley et al. 2015, p. 67).

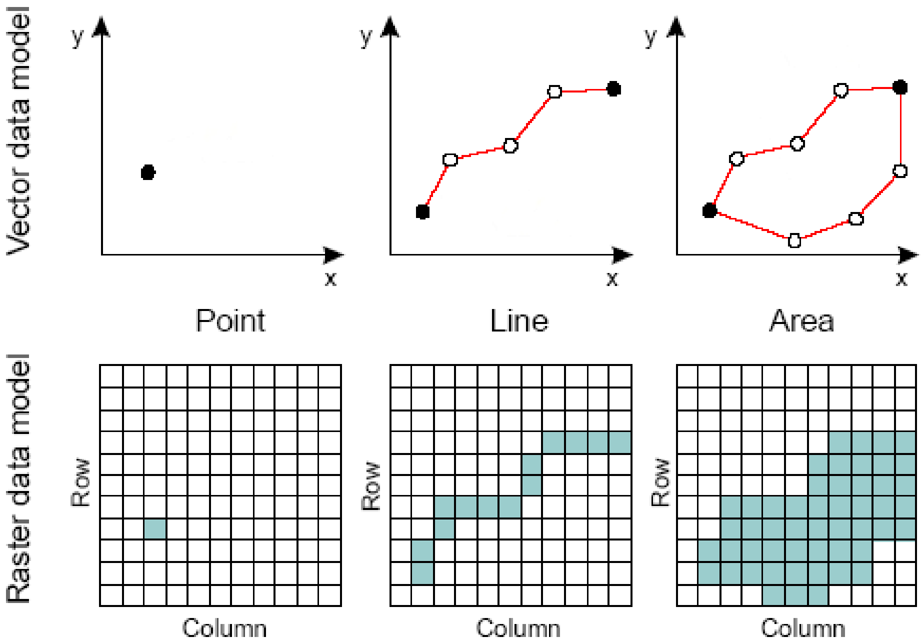
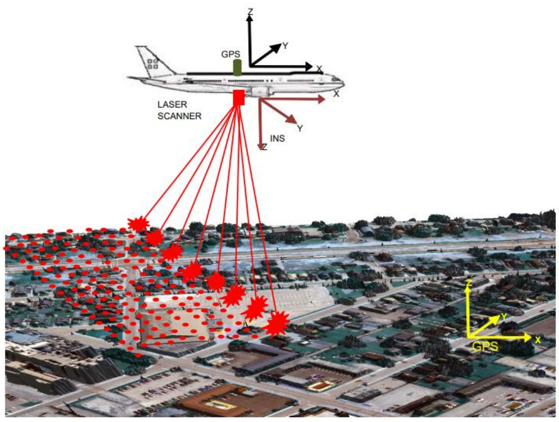


Figure 4.1: Vector and raster data formats, taken from Jukil & Al-Hadad (2017).

4.3 LiDAR

LiDAR (Light Detection And Ranging) is a actove remote sensing technology that uses electromagnetic radiation (laser) to map surfaces in high topographic detail (Figures 4.2 and 4.3). LiDAR consists of an active sensor that emits electromagnetic radiation in the form of, infrared, green, ultraviolet, or visible light. Since the speed of light is constant, the distance to the ground can be calculated with equation 1 (below). Equation 1 is used for calculating the time difference between transmitted and received signal (Liu, 2008). In the literature this is referred to as a laser round trip or two-way travel time.



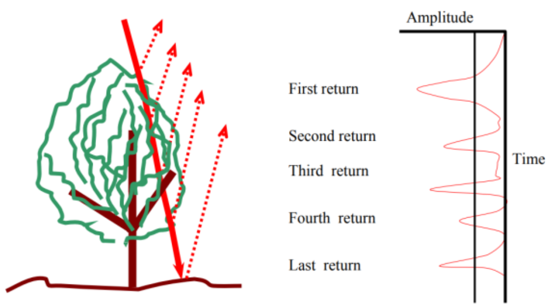


Figure 4.2: Illustrating the concept of ALS (Lohani, 2008). Figure 4.3: Illustrating LiDAR point return (Lohani, 2008).

Equation l: S = V/T

where S is the distance, V is the velocity and T is the time (two-way travel time) (Baltsavias, 1999)
There are several types of LiDAR, but the term is usually associated with Airborne LiDAR Scanning (ALS). This is typically a multispectral sensor mounted on an airplane, helicopter, or drone. Airborne LiDAR enables fast data retrieval of topographic details with high accuracy (Liu, 2008). ALS is the kind used in this thesis.

LiDAR scanners typically consist of a LiDAR sensor, which emits laser pulses with wavelengths between 1000-1600 nm down to the ground. A LiDAR scanner typically emits thousands of laser pulses every second. The signal is then reflected by the substrate, and returns to the sensor, which registers the topography. The time from when the pulse is transmitted until it is reflected and registered by the sensor is used to calculate the distance to the ground. Measurements of distance and direction are done using GPS (Global positioning system) and an IMU (Inertial Measurement Unit). LiDAR produces measurements with high accuracy and resolution using the GPS and IMU. IMU is used to measure the exact position, trajectory, and direction, while GPS is used to identify the X, Y and Z position of the vessel. As a reference point, the GPS in the aircraft is connected to a GPS station on the ground, which helps achieve accuracy down to the centimeter level. (Muhadi. et al, 2020)

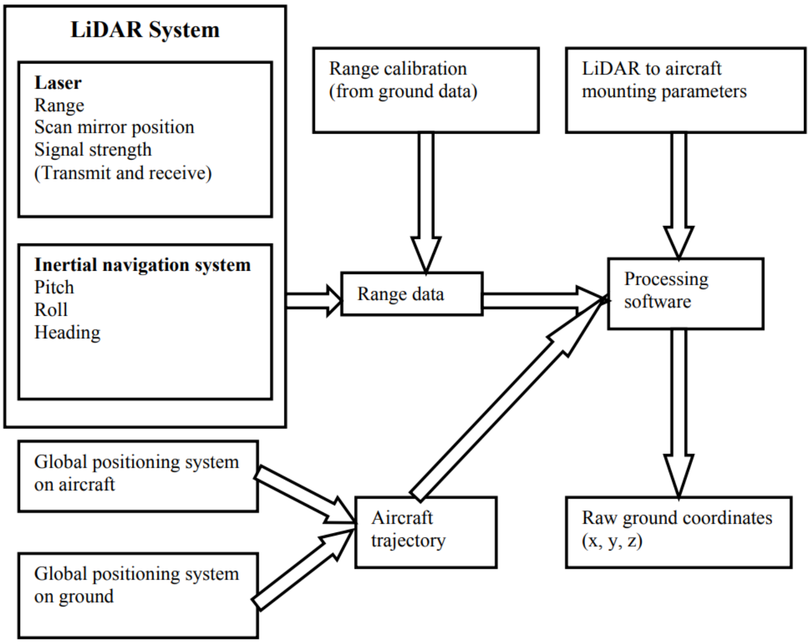


Figure: 4.4 Illustrating the workflow for Aerial LiDAR Scan (Lohani, 2008).

In addition to generating digital elevation models (DEMs) LiDAR has several other applications such as forestry, biology, and natural resource management (e.g., mining) (Muhadi. et al, 2020)

The biggest advantage of LiDAR is that it enables the collection of data for relatively large areas rendered with high point density. LiDAR also has good accuracy compared to, for example, photogrammetry. Photogrammetry is the composition of several photographs to determine the size and position of a photographed object. In photogrammetry, there is a greater risk of vertical inaccuracy due to the parallax that occurs when moving the camera. It is therefore important to have sufficient overlap of the images to achieve a satisfactory result. If the overlap is not sufficient, buildings and other objects that are at the outer edge of the camera lens may appear higher than they actually are, due to the object being photographed at an inclined plane (Aber et al, 2019). The production time for elevation data produced from LiDAR is also usually faster than traditional photogrammetry. In photogrammetry, one is to a greater extent dependent on manually controlling the composition of overlapping aerial photographs using control points on the ground in order to be able to develop terrain models. Despite the disadvantages of using photogrammetry, for example with a drone to produce digital elevation models, it has a number of applications, and can be an effective tool for generating elevation models of small areas, in this case small watercourses. (Muhadi. et al, 2020)

LiDAR also presents some disadvantages, primarily that it does not provide any information about the landscape on the ground. LiDAR cannot distinguish between, for example, agricultural areas or moss-covered mountain landscapes. Another weakness is that LiDAR measures approximately directly from above, which results in more points per square meter being registered in flat terrain than in steep terrain. This means that points in flat terrain are often overrepresented and areas in steep terrain or areas with complex topography are underrepresented (Hengl & Evans, 2009).
Processing of LiDAR data can often present problems as it generates a large amount of point data (The workflow for ALS data capture is illustrated in Figure 4.4). If the computer that is to handle this data does not have sufficient processing power, this will take a long time. The final problem is classification and filtering of the data points to distinguish between ground points and non-ground points. Filtering out non-ground points is fundamental to being able to develop accurate elevation models. To ensure high accuracy, several methods and algorithms of varying complexity have been developed, although none are considered to be 100% accurate (Romano, 2004). It is also worth noting that some of the weaknesses of LiDAR can in some cases be compensated by p, and vice versa (Muhadi. et al, 2020).

LiDAR data is usually stored in the LAS format. This is the industry standard format for storing LiDAR point clouds. Each point contains information or a value that describes the points, in addition to this the points may also contain x, y and z coordinates, as well as many more attributes describing the points (Esri, n.d.*n*).

4.4 Raster interpolation

LiDAR datasets consist of points with irregular distances, and although the point density is generally good, there will always be areas where the point density is not sufficient. These may be areas where point registration is thin, like steep slopes, or areas covered in dense vegetation, where the laser pulse is unable to penetrate to the ground. In such cases, a method called interpolation is applied to estimate the values in these areas (Esri, n.d.*b*).

Interpolation predicts values of cells in a raster where no point samples were collected. It can be used to predict unknown values for different geographic point data, such as elevation values to create a better DEM. Interpolation assumes that spatially distributed objects are spatially correlated. This means that close objects tend to share similar features and properties. As an example, if it is raining on one side of the street, it can be said with great amounts of certainty that it is also raining on the other side of the street. Based on this logic, it is possible to understand that the values of points close to each other are more likely to be equal than points that are further away. Correlation fades with distance, and this is what forms the basis of interpolation (Esri, n.d.*d*). There are several interpolation methods that might be applicable in this thesis, these are presented below.

4.4.1 Natural neighbor

The Natural neighbor interpolation method uses an algorithm to find the closest subset. The input is queried, and weights are applied based on the proportional area to interpolate a new value. it is local since it is based on the nearest point. This also means that the value that it produces is guaranteed to end up within the measured values. This means that natural neighbor interpolation will not generate peaks, pits, ridges or valleys that are not already present in the dataset. The generated surface will pass through the points, resulting in a smooth output. Closest point is generated by the help of a Voronoi diagram (Figure 4.5). In the world of GIS Voronoi diagrams are often referred to as Thiessen polygons (Longley et.al, 2015, p.313).

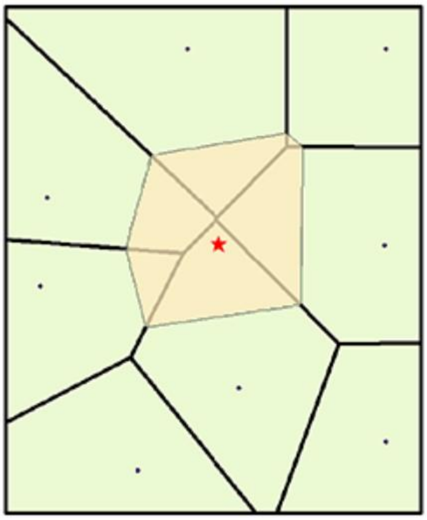


Figure 4.5: Illustrating the principles of Natural neighbor interpolation (Esri, *n.d.k*)

4.4.2 IDW (Inverse distance weighting)

The Inverse Distance Weighting interpolation method relies on spatial autocorrelation, also known as Tobler's first law of geography, which assumes that objects close to each other are more similar than those further away (Longley et.al, 2015, p.314). To be able to predict the value of an area that is not measured, IDW uses the measured values of the points around the area to be predicted (Figure 4.6). The point data that is closest to the place to be predicted has the greatest impact on the result. IDW assumes that each measured point has a local area of influence that decreases with spatial distance. The influential points are marked in red in figure 4.5 below. It assigns a greater weighting to close points, where the weighting decreases as a function of the distance, hence the name inverse distance weighting (Esri, n.d.*e*).

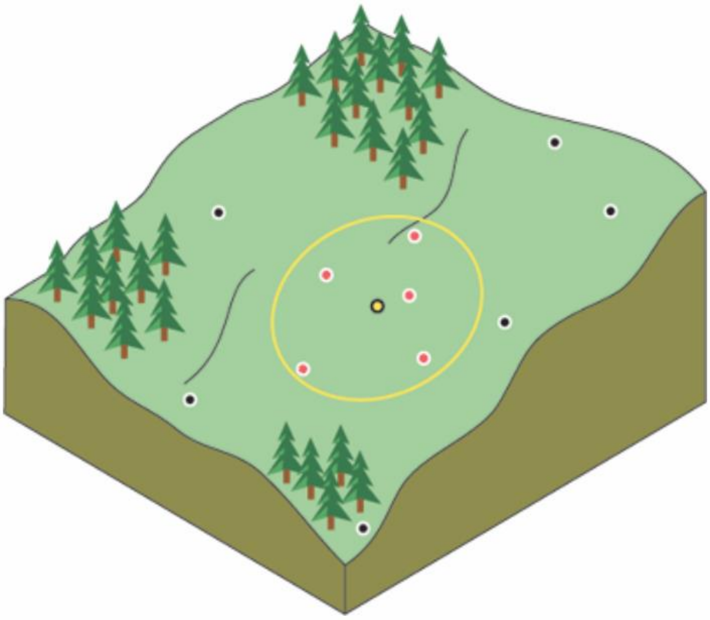


Figure 4.6: Illustrating the principles of Inverse Distance Weighting (Esri, n.d.*e*). The points highlighted in red color are within the decided search radius and will be used as the basis for interpolating the value of the yellow point.

4.4.3 Kriging

Kriging generates a surface from the z values of a set of points from a defined area. An interactive investigation of the spatial behavior of the represented phenomenon then follows, before selecting the best estimation method for generating the output surface. Kriging is unlike other interpolation methods in the ArcGIS interpolation toolset, as it is based on autocorrelation, which is the statistical relationship between the measured points. Kriging can thus both form an estimated surface and say something about the validity or accuracy of the produced estimates. Kriging makes assumptions that the points are spatially correlated, based on the distance and direction between the sample points (Figure 4.7). This is used to explain variations in the surface (Longley et.al, 2015, p.316). The kriging interpolation process consists of multiple steps; statistical analysis of the data and variogram modeling to predict the surface. The use of kriging is most appropriate to use when it is known that spatially correlated distance or directional bias exist in the data. (Esri, n.d.*g*).

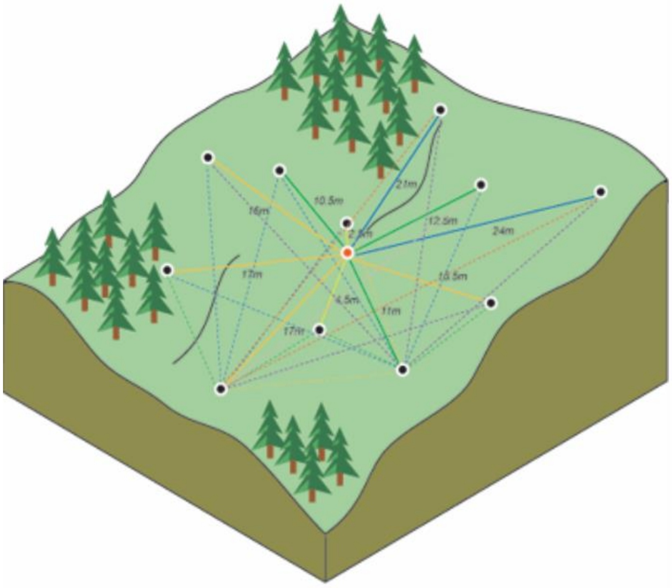


Figure 4.7: Illustrating the principles of Kriging Interpolation. Unlike natural neighbor and IDW interpolation, kriging is based in spatial statistics (autocorrelation), taking all points into consideration, and investigating the geostatistical relationship between the input locations when interpolating the new location (Esri,n.d.*g.*).

4.5 Digital Elevation Model (DEM)

A DEM is a model representation of the earth surface. Normally, LiDAR measurements will produce Digital Surface Models (DSM), which contain information about all objects on the scanned surface (Figure 5.2). Filtering out objects such as trees and buildings produce a model that represents the ground elevation. In this thesis, DEM is used as the term for a model representing the earth's surface elevation. In short, DEM is a grid in Cartesian space,

where each cell is assigned an elevation value representing earth's elevation at a given point. A DEM is a mathematical model, meaning that the model is a statistical representation of a continuous surface. The model is made up of a large number of points or cells with known x, y and z coordinates. The process of preparing a DEM consists of three main tasks. The first is to measure the earth's surface, then generate a model based on the points. The last task consists of validating and dealing with sources of error (Muhadi. et al, 2020).
Digital Elevation Models can be further divided into two categories, TIN (Triangulated Irregular Networks), and Raster. Raster is a regular data matrix consisting of rows and columns (Figure 4.8). Each cell has a height value Z and a longitude value X and a latitude value Y. One of the advantages of raster based DEMs is the simple structure that is compatible with data processing. As a result, relatively simple algorithms can be used. Rød (2015) recommends using TIN for visualization, and raster format for analysis. The analyses around which this thesis is built are based on DEMs with a raster structure.

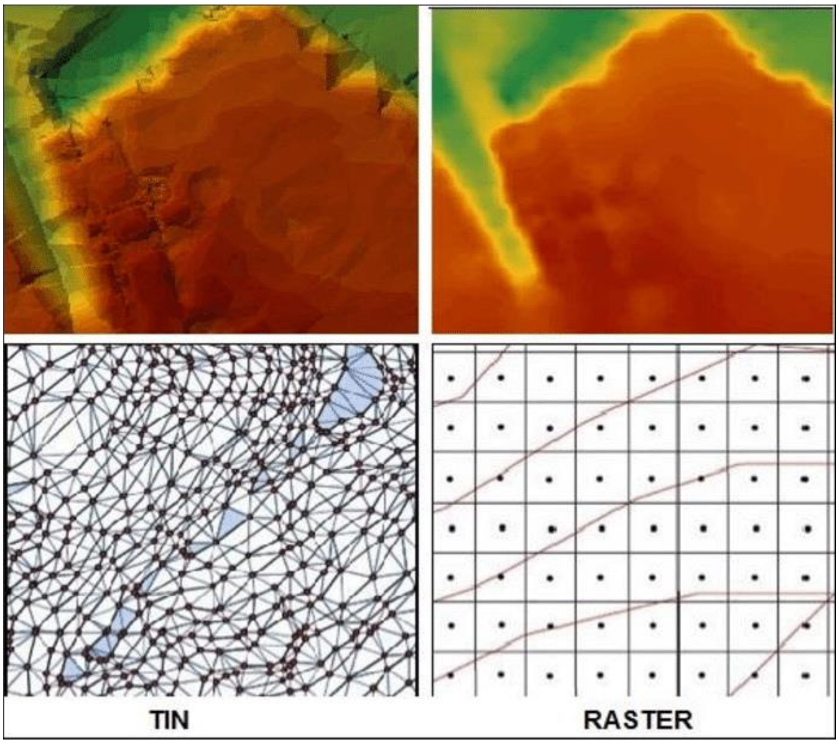


Figure 4.8: Illustrating the difference between TIN and raster-based elevation models. Illustration by Silva. Et al 2016.

One key advantage of raster DEMs is the simple structure, which makes them suitable for implementing algorithms and map displays. Rasters have a uniform spatial structure and almost all properties of gridded DEMs are defined by a single characteristic, the cell size (Hengl & Evans, 2009). This makes Raster models more suitable for computer models used in image processing, such as flow accumulation computations in ArcGIS.

The advantage of the simple structure far outweighs the drawbacks of raster DEMs compared to TIN models. Despite this there are some disadvantages that are worth mentioning, and these factors should be considered when evaluating the results. In a raster DEM, complex topography are often under-represented, while flat and smooth terrain are over-represented in relation. The difference in distance between the raster cell center in cardinal and diagonal directions can have negative impacts on the precision of hydrological models (Hengl & Evans, 2009).

4.6 Hydrological Modeling in GIS

In hydrological analyses, the watershed is an important unit and can be defined as the total area upstream from the outlet point. In a DEM, the watershed is the collection of cells that drain to a defined cell, the outlet point. Except for water that evaporates, all precipitation that falls in a watershed will eventually pass through the outlet point. The outer boundaries of watersheds/watersheds are known as drainage divides and marks the border between drainage basins. To generate a hydrologically correct DEM it's important to identify any errors in the DEM. This can be done with the Identify Sinks tool in ArcGIS. The Identify Sinks tool locates sinks or pits in the DEM that can cause discontinuities in the waterflow network. Sinks are often faults in the data source caused during the generation of the DEM (Esri, n.d.*h*). A sink in a raster is a single cell or a collection of cells in which it is not possible to assign a flow direction in accordance with the flow direction codes, either because the area is completely flat, or the cell has a NODATA value. To fill the sinks ArcGIS Pro has a built-in tool called "Fill" that assigns a flow direction to each sink cell.

Sinks exist for a number of reasons but are mainly a product of errors introduced in the generation of the DEM (i.e., the interpolation method) or from the collection of the original LiDAR data. Natural sinks also exist, but they must be verified by fieldwork and ruling them out completely is often the simplest solution. Sinks occur when the surrounding cells have higher elevation values than the center cell. (Esri, n.d.*l*) They may also occur due to two raster cells flowing into each other, causing an infinity loop where they are flowing back and forth between each other. Sinks can be identified and evaluated using the sink tool. (Esri, 2020)

Before you can run the flow direction tool it is important to ensure that the DEM is "hydrologically correct". This means that pits and indentations in the DEM, here referred to as sinks, must be removed to ensure continuous water flow. The fill tool can also be used to remove peaks, opposite of sinks in the DEM. The principle of Fill is visualized in figure 4.9 (Esri, n.d.*h*)

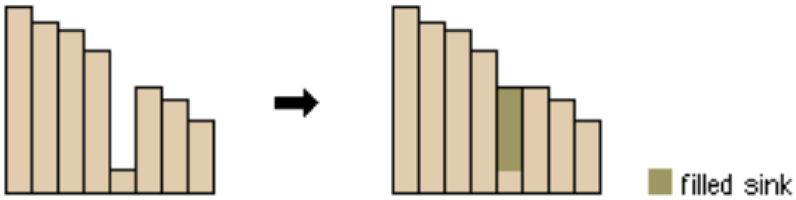


Figure 4.9: Illustrates the principles of filling sinks in a DEM (Esri n.d.).

4.6.1 Flow direction

Determining how water will flow across a DEM is crucial for hydrological modeling. Several flow algorithms have been developed to model the flow direction of water in a DEM. These can be sorted into two main categories: single flow and multiple flow.

One of the oldest and most widely used single flow algorithms is D8 (eight flow direction). The D8 algorithm determines which direction the water will drain from each cell based on a 3x3 movable window (O'Callaghan and Mark, 1984). D8 is based on the assumption that several cells drain into one cell (converging current), but it cannot model that a cell drains into several cells (diverging currents) (Esri, n.d.*j*). The principles of the D8 Algorithm are illustrated in figure 4.10.

To solve this problem a variety of multiple flow algorithms have been developed. Common to all of these is that water has the ability to drain from one cell to multiple cells (diverging current). The method can in some cases overestimate the spread of the water, and not limit the flow to actual stream channels. D-infinity is a multiple flow direction algorithm developed by Tarboton (1997). The method determines the direction of drainage to the cells in that the direction function is continuous and is specified as an angle between 0 and 2π. The output is illustrated in degrees (0°-360°) (Tarboton, 1997). Common to all the algorithms is that they all are based on water flowing to the area of steepest descent.

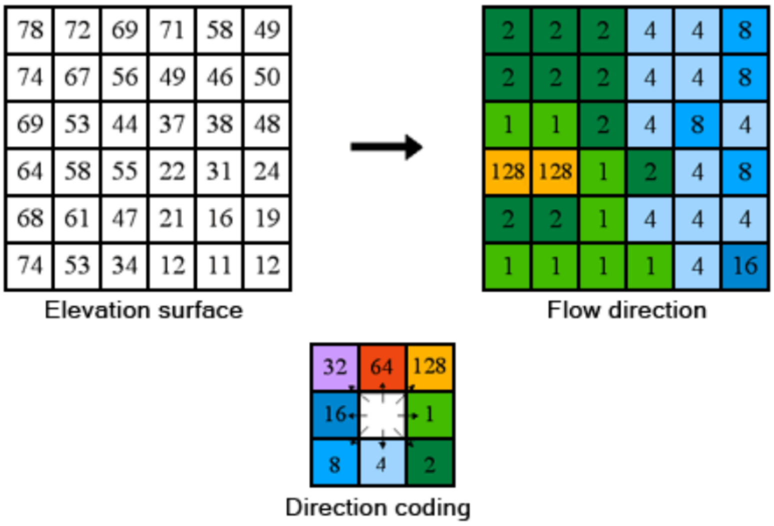


Figure 4.10: Illustrates how the D8 flow algorithm works. (Esri, n.d.j).

4.6.2 Flow accumulation

An important hydrological parameter is the accumulation of surface water. The flow accumulation tool in ArcGIS builds on the output raster of the flow direction tool and constructs a new raster showing where water will accumulate based on the assumption that water always will flow in the direction of steepest descent to its downslope neighbor. The output is an accumulation raster where each cell is assigned a value based on how many upslope cells that drain into the particular cell (Esri, n.d.*i*). In figure 4.11 The principles are illustrated. This can further be utilized to indicate accumulated area.

To exemplify, if a DEM has a resolution of 1 meter and 100 upstream cells drain into a single cell, they will carry the water from an accumulated area of 100 square meters. How the resolution of the raster is calculated is presented in Chapter 5.3. In this study a DEM with 0.5-meter resolution was used.

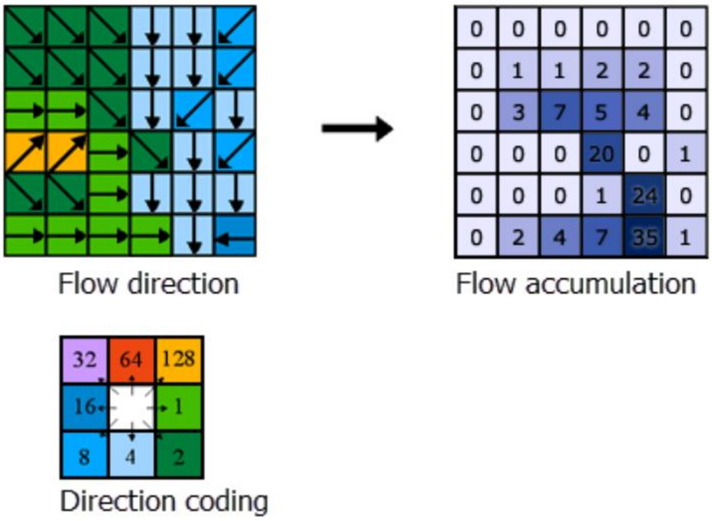


Figure 4.11: How Flow accumulation works (Esri, n.d.i).

4.6.3 Weighted overlay

In ArcGIS parameters such as precipitation amounts, infiltration to the ground and erosion can be included in the models in the form of weight rasters. This allows the user to control the influence that different parameters have on the model result. This allows for simulating the effect different surfaces, such as asphalt, grasslands or woodlands have on the water flow (Esri, n.d.*p*).

4.6.4 Con

The Con tool performs a conditional if/else evaluation of all the cells in an input raster. The con tool can be used to filter out unwanted values by assigning true or false values (Figure 4.12). In this thesis Con will be used to filter out smaller streams that do not have enough contributing area to cause any damage (Esri, n.d.*c*).

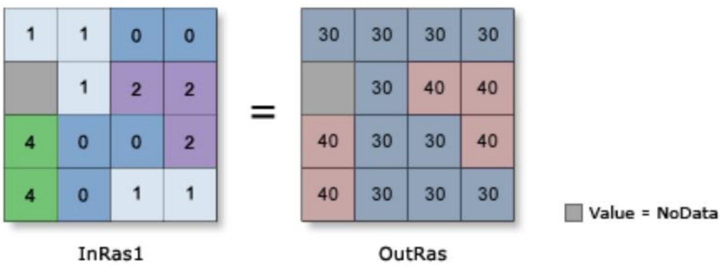


Figure 4.12: Illustrates the principles of the Con tool. (Esri, n.d.*c*)

4.7 ModelBuilder

ModelBuilder in ArcGIS Pro is a visual programming language that can be used for building geoprocessing models. Geoprocessing models automates the process of documenting the spatial analysis and managing the data. ModelBuilder uses a drag and drop interface, ensuring that it is easy to use and the schematic setup aids visualization, making it easy to find faults or inconsistencies in the model. ModelBuilder models are represented as diagrams that chain together sequences of processes using the output of one tool as input in another tool. Input data are symbolized with blue boxes, tools are symbolized in yellow, and output data in green (Figure 4.13). ModelBuilder is Python compatible, and ModelBuilder tools can easily be exported as Python script. (Esri, n.d.*o*) This makes it possible to edit the script rather than editing the model. Using ModelBuilder in this thesis ensures versatility and makes it so that people that are not familiar with ModelBuilder can edit the tools in Python. The results can also be edited and applied across multiple platforms.

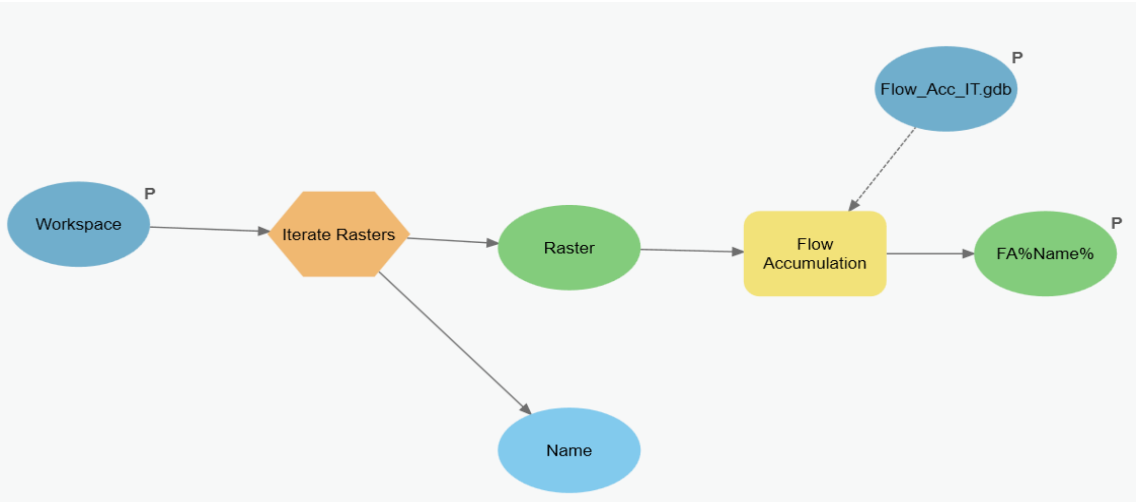


Figure 4.13: ModelBuilder data types.

4.7.1 Iterators

In ArcGIS there are a number of different iterators in ModelBuilder that you can use for iterating different values, datasets, datatypes and workspaces. Iteration is also known as looping or batch processing. Iteration involves repeating a process over and over to help automate repetitive tasks, thereby reducing time and effort required by the user to perform a task. (Esri, n.d.*f*) With iteration in ModelBuilder, a process can be executed over and over, using different settings or input data in each repetition. ModelBuilder helps provide flexibility, as an entire model or a single tool or process can be looped and executed repeatedly. In ModelBuilder, iterators are illustrated with orange hexagons as shown in Fig 4.13.

4.7.2 Submodels

ModelBuilder does not allow for several iterators within the same model. This is because when an iterator is added to the model, the iterator is set up to loop through all the tools in the model, not just the ones “downstream” from the iterator (ESRI n.d.a). To solve this problem, it is possible to add sub models. Adding sub models involves the creation of an independent model within ModelBuilder, which you then add to the main model (i.e., nesting).

In the Generate_Blockage_Tool and Hydro_Tool models nesting has been utilized to loop through all of the input data. To ensure that the model is running in the intended direction, the result of the previous tool has been set as a precondition for the next tool, this is to ensure the tools are running in the intended order. The precondition function makes it so that a tool will not run before the preconditioned output is generated. This is illustrated in figure 4.14 below, where the tool Iterate_FlowAcc will start running when the output from Iterate_FlowDir: “FD_%Name%” is finished. All the models developed in this thesis are running from left to right.

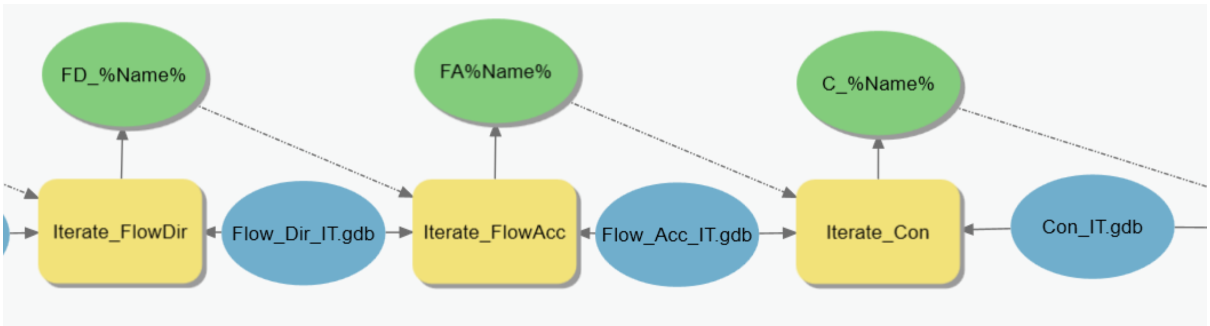


Figure 4.14: Illustrates how sub models can be used in ArcGIS. The flow “Iterate_FlowDir” tool as illustrated in Figure 4.13, is here nested inside another tool.

5. Methods

In this study a set of tools and methodologies has been made to automate the process of water delineation, and how modeling accumulated water can help indicate damage potential of a critical point if it gets blocked or obstructed. This Chapter will present shortly the function and workflow. The ModelBuilder flowcharts for the models can be found in Appendix B. The scripts for the tools can be found in Appendix E.

5.1 Data

The data used in this thesis were carefully chosen, since the primary aim is to develop tool sets that can be applied to any watershed in Norway. With this in mind it was important to ensure all of the data that are needed are accessible in all parts of Norway. The choice of input data therefore fell on the Felles Kartdatabase datasets (FKB Datasets). LiDAR data was sourced from Kartverket via Hoydedata.no. The data that were not available online were either collected by me or others on the Trygg Elv team. The data utilized in this thesis was downloaded and collected from various download portals, an overview of the data can be found in the data overview table in Appendix D.

Felles Kartdatabase is defined by FKB product specifications approved by the Geovekst-forum. FKB data is data that follows the FKB product specification. The datasets in FKB consist of, among other things, altitude, water, soil type (AR5), land use, buildings, roads, railways and airports.

All elevation data are available from hoydedata.no. The data used in this thesis is derived from the project "nasjonal detaljert høydemodell" (national detailed elevation model). The project started in 2016 and runs until 2022. The goal and result of the project is to produce a detailed digital elevation model for the whole country of Norway with a resolution of at least 1 meter. In addition to laser data collected through the national detailed elevation model project, other laser and raster data for other parties in "Norge Digital" (Norway digital) can also be visualized and downloaded through the hoydedata.no download portal. Elevation and depth data are suitable as a basis for detailed terrain models, terrain analysis and various forms of terrain visualization. It is possible to download readymade DEMs from Hoydedata.no, in various resolutions, as well as smaller local projects with higher resolution. The point clouds have varying density (point / m^2) to adapt to different objects. This thesis will utilize terrain data with a resolution of 0.5 meters (Kartverket, 2021).

5.1.1 Field data

During the summer of 2021, bathymetric data were collected for parts of the study area (from the outskirts of Ler to the Ler Village center; Illustrated with purple dots in Figure. 5.1. Flood zone maps were produced by Graf, 2021and Gomez, 2021. The flood zone maps will be compared to output from the models developed in this thesis.

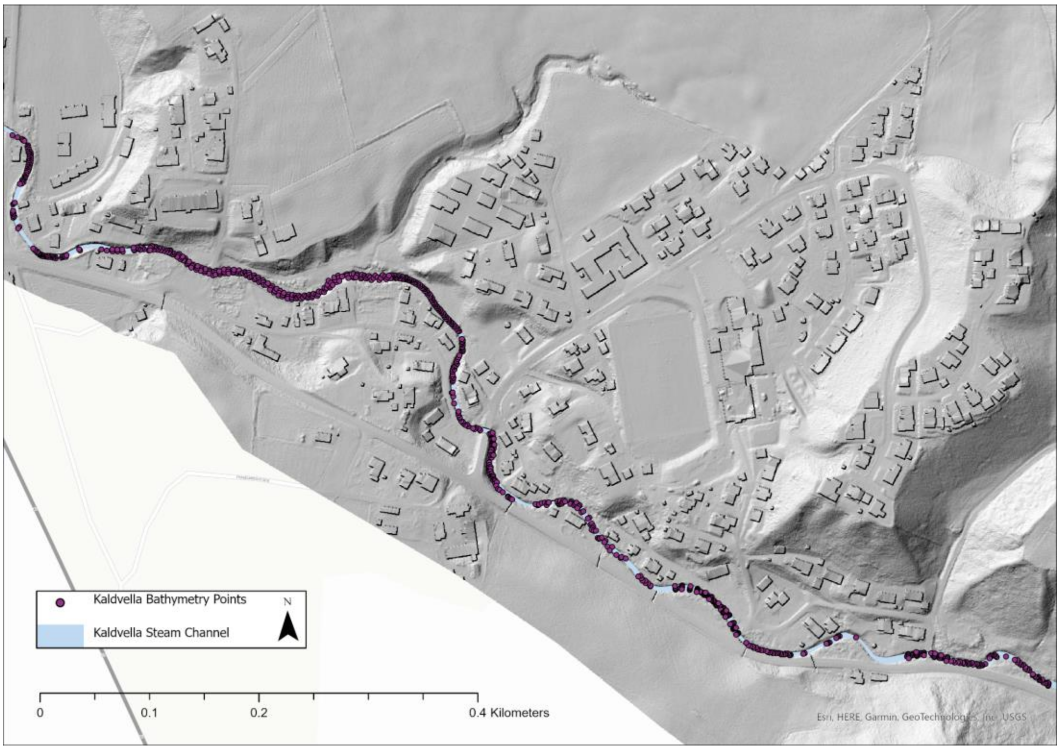


Figure 5.1: Showing the bathymetry points collected with GPS.

The bathymetry points were collected with an ALTUS APS 3x and a Leica VIVA GS12 GNSS GPS with RTK (Real Time Kinetic) connection to ensure sufficiently high precision. In the planning stage it was decided that we should ensure that all points were collected with a maximum of 2 centimeters of deviation. In the field this proved difficult, due to heavy vegetation in or above the stream, or because the stream ran too deep in the terrain. As a result, some individual points and areas have an error of up to 5 cm, which was considered an acceptable compromise. In some areas the stream channel is running relatively deep in the terrain, while at the same time being overgrown by vegetation, making it problematic to get satisfactory point accuracy any time of the year. The areas with no points collected were interpolated from adjacent points. No further attempts were made to collect bathymetry for this thesis. This was decided because accurate river bathymetry is not critical to the results, since it is not a hydraulic model, and the output waterline is only based on the line of steepest descent, from the upstream pixel, as described in Chapter 4.6. On 16.11.2021, additional GPS point data were collected from areas where the floodplain in the study area had been modified and from artificial levees that had been added since the LiDAR data was published. The points were collected to the same specifications as the bathymetric points with the aim to capture points with an accuracy of +/-2 cm. The collected data was utilized to manually correct terrain features such as raising the floodplain and adding the levees. This is discussed in further detail in Chapter 5.5.

5.1.2 Batch clip

Data are available in different standard sizes for downloading national, county and municipality-scale. To automate the clipping process the batch clip tool in ArcGIS Pro was selected. The batch clip tool clips all the input data in the input folder to fit the clip feature. In this study the clipping feature is the Kaldvella watershed. The watershed polygon used in this study is derived from NVE. This step can be best described as cutting all the data with a cookie cutter, so that all the data have the same extent. Clipping the data and minimizing the file size is important for shortening the data processing time. There is an option to set the output coordinate system with the batch clip tool, which is important, since working with data with different spatial references might introduce errors. ModelBuilder flowchart can be found in Appendix B, and script can be found in Appendix E

5.2 LiDAR data filtering

LiDAR data were downloaded for the whole of Melhus municipality and clipped to fit the Kaldvella watershed using the Kaldvella watershed polygon from NVE.
The catchment area of Kaldvella consists of 91 LAS files with a total number of 260,087,430 LiDAR points, of which 143,896,483 are ground points. This corresponds to ~ 59% of the points. The average point spacing for all of the Las files is 0.406912088 meters. The LiDAR data is provided with a spatial reference. It is important that all the files use the same coordinate system. The horizontal reference is UTM zone 32N with ETRS 1989 as the datum. The vertical reference is NN00 (Norway Normal Null 2000), an updated vertical coordinate system made for Norway.

In the LiDAR data filtering in the “Make LAS dataset layer” tool it is possible to select which point types to include in the DEM, making it possible to filter for example ground points from none ground points. In The world of GIS, the terms Digital Terrain Model (DTM) and Digital Elevation Model (DEM) are often used synonymously for the bare earth points. When selecting for all points you get a Digital surface model. The difference is illustrated in Figure 5.2.

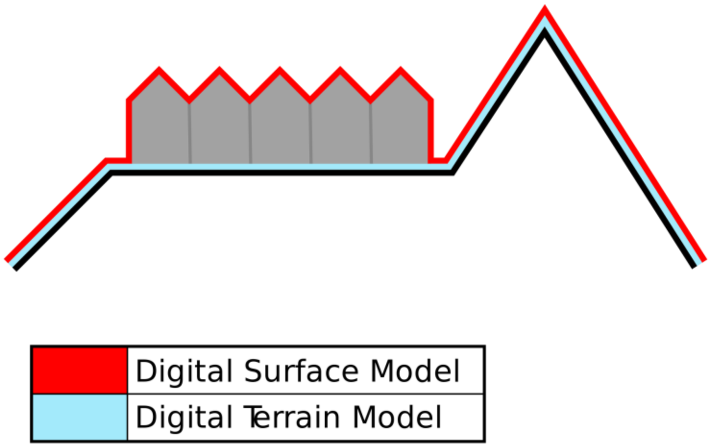


Figure 5.2: Showing the difference between DSM and DTM (Martinn, 2011)

5.3 DEM generation

The Las_To_Raster_Tool has been made to automate the process of generating DEMs from LAS data. The workflow for the tool is illustrated in Figure 5.4. The input format is zLAS which is a compressed LAS format. The convert LAS tool in ArcGIS decompresses the LAS data, so that all the information in the LAS file is available. FKB_Vanngrense (water boundary) was used as surface constraints. With the tool "Make LAS Dataset Layer"' it is possible to select and filter the LiDAR data by class codes, making it possible to filter out points such as trees and vegetation. In this study Class codes 2 (Ground), 9 (Water), 10 (Rail), 11 (Road Surface) and 17 (Bridge Deck) was used (Esri, n.d.*m*).

5.3.1 Choice of cell size

The resolution of a raster DEM will be limited by the number of laser points in the terrain. McCullagh (1988) suggests that the number of cells in the raster should correspond to the point density within a covered area. McCullagh (1988) has designed an equation (Equation 2) to estimate this where S is the size of the cell, n is the number of (ground) points in the terrain and A is the covered area (Liu, 2008)

$$\text{Equation 2: } S = \sqrt{\frac{A}{n}}$$

In this case, the size of the catchment area of Kaldvella is 29700000 m² based on figures from Nevina (Figure 2.2). The number of ground points registered within this area is 143 896 483. This constitutes ~ 55% of the total 260087430 points in the LAS dataset. Based on this, the equation recommends cell size of 0.455. I have therefore rounded up and will use a cell size of 0.5 in my DEM, to make both map algebra and the data processing easier.

$$0.45431 = \sqrt{\ }\frac{29700000\ m^2}{143\ 896\ 483\ points}$$

In the project, a point density of 5 points per square meter for ground (Red LiDAR) and 4 points per square meter in water (Green LiDAR). The green LiDAR data was collected by Terratec in 2016. In areas where the green LiDAR has not reached the riverbed, the data is supplemented with multi-beam echo sound (Høydal, 2016). Due to the size of the stream channel of Kaldvella some areas have very poor coverage, and there are long distances between the points actually reaching the streambed. Because of this the green LiDAR does not represent the actual bathymetry, but interpolated values based on the few LiDAR points to hit the stream bottom. Other problems are changes in the terrain made after the LiDAR Data was collected. This makes it necessary to manually correct the resulting DEM. In other study areas with better green LiDAR data coverage, it might prove more useful than here. For this reason, green LiDAR will be included in the thesis. The cell size of a raster is defined by the size of each square pixel and makes it possible to control the properties of the DEM. Although there are several benefits to a raster-based DEM, the representation has some disadvantages. The grid tends to underrepresent complex topography and overrepresent a smooth and flat topography. This is also partly due to the point density in the LiDAR data on which the DEM is based. In hydrological modeling the grid can have a negative impact on the result of the analysis. The distance to the center of the cell, where the elevation value is stored, is different between the cells that are diagonal and adjacent of and just above (Hengl & Evans, 2009).

The tools this thesis aims to produce will work with all kinds of raster surfaces, both integer and float, and all resolutions. To ensure maximum accuracy the resolution of the DEM plays a major factor, and a higher resolution will also produce a more accurate result, but as Hengl & Evans (2009) points out, higher resolution also results in longer computation. When working with small rivers and streams where the inclusion or exclusion of small features can have huge impacts, on the modeled flood water path. The inclusion of micro features can therefore be critical to the accuracy of the model output (Muhadi et.al 2020). Furthermore, making an educated consideration of the most appropriate grid size is also important since processing time increases exponentially with decreasing grid size (Barr & Mansager, 1996, p.39). Thus, a model with too fine a grid will result in a slow model that requires a lot of disk space. At the same time as ensuring all these things it is also important to note that the final product, the tools should be easy to use and ensure maximum accuracy while retaining a user-friendly workflow.

5.3.1 Choice of interpolation method

For this application the drainage line analysis is based on the assumption that all precipitation will act as surface runoff. For this thesis it is more important to ensure an uninterrupted flow of water rather than the elevation being correct. Meaning that ensuring water is flowing as it would in the natural terrain is more important, than elevation values being correct to centimeter level. At the same time sustaining small features is important. Since this study looks at small rivers where inclusion of small features or the lack thereof, both can have substantial impact on the model output. It is therefore important to find a solution that will display small features to a satisfactory degree. Due to the fact that the model should be able to be applicable to all types of small watersheds, poses some issues. The model will have to be able to handle a variety of topographies, with the least amount of modification. In this thesis there has not been done any comparative analysis of different interpolation methods. These choices are purely based on results of existing work and recommendations from the literature such as Arun (2013) and Guarneri & Weih (2012).

The tool sets this thesis aims to produce will have to work with a variety of different topographies and terrains. To make the tools as easy to use as possible the study will take a “one size fits all” approach. While Kriging might be the superior interpolation method in terms of accuracy, this is highly dependent on the input, which also requires deeper knowledge of the data. Since the tools will have to work with a range of terrains, while sustaining a high degree of accuracy, and inclusion of small features. This rules out Natural neighbor interpolation, since this interpolation method will smooth the terrain, and small features will be removed. IDW is deterministic while Kriging is based in geostatistics. IDW assesses the predicted value by taking an average of all values within the specified search radius and applying "weights" to the points based on distance, where adjacent points are prioritized. Kriging is also more reliant on the user having better knowledge of the data, compared to IDW, since it requires more input values (Arun, 2013). In addition to this Kriging interpolation is not an option in the LAS dataset to raster tool. So, to use Kriging interpolation you have to use other tools. IDW interpolation is the best compromise, both in terms of accuracy and complexity. Since search distance and barriers can be set, this makes IDW interpolation the best allrounder, considering the various types of watershed topographies the tool has to handle. Surfaces created with IDW interpolation will not exceed the known values of the sample points, meaning that IDW is a good interpolator for estimating phenomena where distribution and distance are strongly correlated. For ensuring maximum versatility in the number of topographies the tool is able to process effectively. The interpolation method used in this study therefore fell on IDW, with binning and void fill set to “linear”. ModelBuilder flowchart can be found in Appendix B; 11.2. The script can be found in Appendix E; 14.2.

5.5 DEM modifications

To correctly model the flow of water, the DEM has to be modified. The LiDAR data captures the ground from above, and underlying or underground features will not be included in the DEM. The solution to this is to add these features to the DEM afterwards. In this thesis this has been achieved with a combination of manual processes, others have been automated in ModelBuilder. Things that are unique to Kaldvella has not been automated into tools. So, processes such as raising the floodplain and adding buildings from FKB polygons has been done manually. Adding levees to the DEM has been made into a tool to make the process easier, despite being a problem local to Kaldvella. Things that are expected to be an issue also in other watersheds have been automated, such as adding culverts and removing bridges. The modification of the DEM is only to prepare it for the hydrological model.

As mentioned in Chapter 3.6 the results from the model will be compared and evaluated up against results from HEC-RAS. The program requires a lot of knowledge to operate, both within hydrology and hydraulics, in addition navigating the user interface itself also requires prior knowledge. In early stage planning other tools might be more effective and more user friendly than HEC-RAS for identifying problematic areas. On the other hand, the program is capable of very sophisticated analysis, much more so than both SCALGO Live and ArcGIS Pro. This is therefore a very good tool for comparing and verifying the output from other tools, such as SCALGO Live and ArcGIS Pro. This thesis will use HEC-RAS output to compare results, while SCALGO Live will be used to compare functionality, not output.

In the bottom of the Kaldvella watershed in the middle of Ler, there has been a house development project over the last couple of years. This area is nice and flat and relatively easy to build on, being a natural floodplain. This invites some problems, because the DEM predates the newly built houses. This means that the ALS data from 2016 does not include the new buildings or the modified terrain. The terrain in this area has been raised to prevent flooding, related to this there has also been installed storm drains. In context to this, Me (Torgeir Skurdal) and Prof. Knut Alfredsen went to Ler to collect GPS point data that was later compared to the DEM. The comparison showed that the whole area has been raised by approximately 1 meter. Since there is no time to collect new LiDAR data or register all of the terrain as points to make a new DEM, a choice was made to raise the whole area by 1 meter, to ensure that the terrain is more accurate than the current LiDAR data. This is a compromise and could be a potential source of error in the validation of the model. There are also drainage features being installed in this area in parallel with this thesis being written. Rather than following the continuous development as it takes place, and having to update the DEM, the features added after December 2021 are not included.

5.5.1 Buildings

The buildings have been raised to simulate the real-world situation. This was done by extracting the building's footprint with the tool extract by mask. This results in buildings raster with correct elevation. Using a raster calculator five meters were added causing the buildings to extrude five meters from the ground. The optimal solution would have been to use the real height of the buildings but the age of the LiDAR data, makes it so that new buildings have been built inside the Kaldvella watershed, as well as many buildings that have been demolished. As a compromise all buildings were all raised by the same height value. This causes the rain falling on roofs to not follow the actual flow direction, which would follow the sloping of the roof. Figure 6.4 in the results illustrate the DEM with and without buildings raised.

5.5.2 Floodplain

Down on the floodplain where they have recently built new houses the terrain has been changed and the terrain has been manipulated to a large degree. To better reflect the current situation, there were collected GPS points on the floodplain (Figure 5.3); the collected values have been used to make a new interpolated floodplain raster based on the elevation value of the GPS points. This results in a raised floodplain that is more similar to the current terrain. This is a compromise and does not represent the current situation perfectly, but it is an improvement compared to the old LiDAR data.

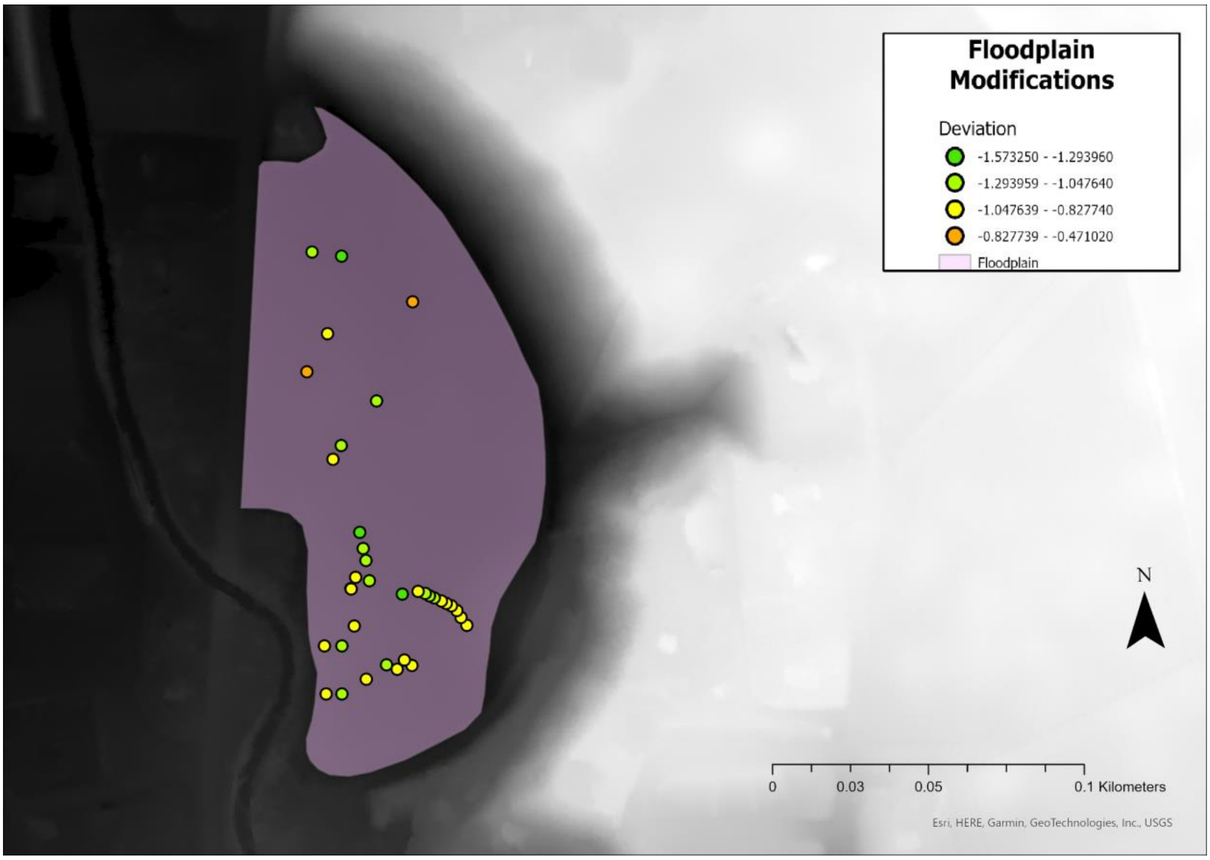


Figure 5.3: Showing the collected elevation points on the floodplain.

5.5.3 Stream Channel

In his thesis Graf, 2021 mentioned that his model could be improved by mapping and adding the river bathymetry to the model. In relation to Terratec's survey of Gaula in 2016, on behalf of NVE, there is existing depth data for the lower part of the study area. In the project a Optech Titan (green laser) was utilized to collect bathymetric data in the river Gaula. In the project description from NVE, it is specified that the project should cover 15 meters above normal water level. (Høydal, 2016) As a result, the lower part of Kaldvella and large parts of Ler village center have been mapped with high resolution Green LiDAR. Despite this the penetration rate in the Kaldvella stream channel is overall very poor. Likely due to the size of the stream channel being too narrow or overgrown by vegetation at the time of the green LiDAR survey. Due to poor coverage these data cannot be used for Kaldvella.

An attempt was made to supplement the green LiDAR, with manual GPS points in the summer of 2021, and large parts of the stream were mapped. Despite a lot being mapped, at the time of the GPS point collection in June 2021parts of the stream channel were overgrown by trees or other vegetation, and the results were not as good as anticipated in these areas. In the areas with low or no coverage, the bathymetry was interpolated. Here there was made the choice to drop the whole stream channel by the average of all the collected points (0.287 meters). The result is a stream channel that more correctly represents reality. It is important to underline that correct bathymetry is not critical to this thesis, since it is not a hydraulic model, such as the work by Graf, 2021. From Figure 5.4 it is possible to see that the DEM stream channel on average is ~0.28729 meters higher than the actual stream bottom.

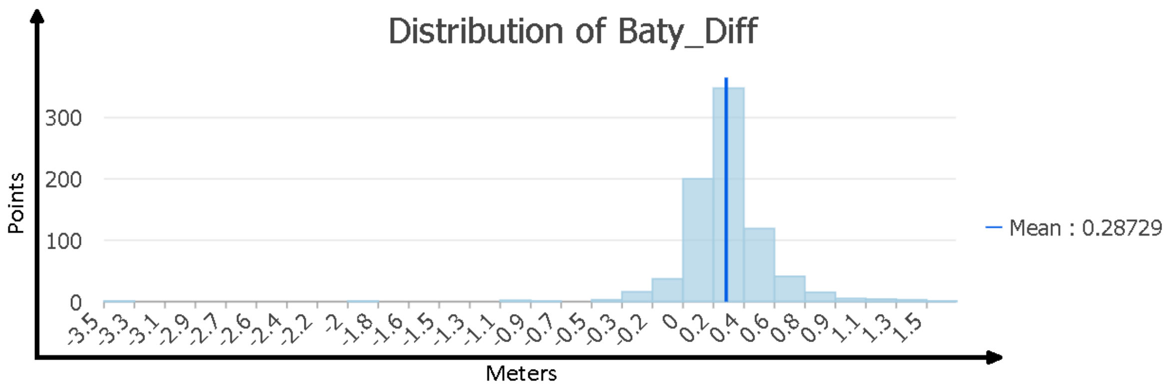


Figure 5.4: Showing the difference between the bathymetry point and the LiDAR data.

Within the Kaldvella watershed, there are several bridges that are not registered in NVDB, the national databases for roads, owned and managed by Vegvesenet (the Norwegian road authorities). These are most likely private bridges, or old enough to predate the NVDB database. In and near the center of Ler these bridges have been measured with GPS and made into a feature class for this project. The other unregistered bridges are approximated by intersecting the water features (FKB vannområde) and the road features (FKB vegområde), then taking samples to visually inspect if the result is accurate. Some of the suggested bridges coincide with registered culverts; these locations have been removed from the bridges feature class using the erase tool. When generating the DEM from LiDAR data the model uses ground points. Due to the angle of LiDAR scan it is not able to capture points directly under the bridges. This causes remanence of the bridges to be left in the DEM. This has an impact on the accuracy of the drainage line and has to be addressed. To solve this issue the “Remove_Bridges_Tool” was made. The tool collects the elevation from upstream and downstream from the bridge locations, then interpolates new values using IDW interpolation. This ensures the flow of water follows the stream channel, such as in the natural environment. The ModelBuilder flowchart for the Remove_Bridges tool can be found in Appendix B; 11.3. The script can be found in Appendix E; 14.3.

5.5.5 Culverts

Culverts and drainage pipes were also downloaded from the NVDB. The dataset includes both the length and diameter of the drainage features making it relatively easy to include them in the DEM to ensure hydrological correctness. The data is quite extensive, and there are a lot of drainage pipes in the study area. Despite this there probably are existing pipes, either old or privately owned that are not included in the database. The “add culverts to DEM” tool adds culverts to the DEM to better simulate the real-world situation. Since ArcGIS Pro is not able to model drainage pipes in a simple way and including these are vital to simulate real world situations. This has been solved by dropping the DEM in the culvert locations. This is done by extracting the rasterized location of the culverts, and then using the tool “raster calculator” to recalculate their elevation value. The new elevation value is set using the attribute KOORDH, which is the coordinate height of the culvert. The values have been validated by sampling some random culverts and validating the KOORDH with the average of start and end point of the culvert. The rasterized culverts with the new corrected value are then added into the main DEM. If the KOORDH is not registered the tool runs a conditional if/else evaluation to find if KOORDH exists, by checking if it has a value above 0. If this is true, the tool will use the KOORDH value. If the evaluation comes out as false the tool will take the alternative path, and estimate the culvert value based on elevation values in the DEM. This is done by extending the culvert line by 25cm in each direction. The start, mid and end points of the extended line are then extracted as points using the “XY table to point” tool. Using the tool “extract value to point” it is possible to extract the DEM elevation value in these locations. Using the “calculate field” tool the midpoint is recalculated using the average value of the start and end points.

When closely investigating the Hillshade DEM it is possible to see that there are locations that clearly have culverts. This thesis has not developed any methodologies for automatic extraction or indication of culvert locations. To solve this problem the FKB Vann område (water area) and FKB vann grense (water boundary) datasets has been combined into one dataset “FKB_All_Water” the water boundary was expanded with a 25cm buffer to give it a more realistic surface extent, before being merged with the Water area data. The FKB_All_Water was then used to drop the DEM by the difference of the DEM and the bathymetry point data. This ensures that water will follow its natural flow path.

These areas will possibly show up as damage points in the tool output. An alternative solution to this is to remove the damage points that intersect FKB_All_Water.
ModelBuilder flowchart can be found in Appendix B; 11.4. The script can be found in Appendix E;14.4.

5.5.6 Levees

The Laser data predates many of the flood protection measures that have been added in the year after the LiDAR data was collected. To ensure that the model is representing real world values the levees have been added. The levees have been constructed from GPS point data collected in the field. To automate this process the tool, add_levee has been created. The Input GPS points are first converted to line features. These are then buffered with 0.5 meters in all directions to give them a more accurate footprint. The buffer polygon is then used as a masking feature to interpolate the points to raster data, that is then added to the main DEM. ModelBuilder flowchart and Appendix B; 11.5. Script can be found in Appendix E;14.5.

5.5.7 Block critical points

The Block critical points tool (Generate_Blockage_Tool) creates obstacles in the stream channel. The critical points used in this thesis are known problem points in Kaldvella where the stream channel is shallow or narrow. Other locations are bridge locations. The input for this tool is point locations. The points are buffered and clipped using the FKB vannområder polygon; this produces an oblong polygon that only covers the stream channel. This polygon is then buffered, and the tool spatial statistics are run for the buffer zone, the highest value in the buffer zone is then assigned to the polygon. This fills the critical points completely. By using this logic, it is possible to find where water will flow when the stream channel is blocked with sediment, or other obstacles such as vegetation. The polygons are then rasterized and added to the modified DEM. The workflow for this tool can be found in Appendix B; 11.6. The script can be found in Appendix E; 14.6.

5. 6 Iterate Hydro

The drainage line model developed in this study automates the water line delineation process by creating iterator tools for all the necessary steps. These steps are presented in Chapter 4.6 above. This is done by nesting the individual models in the main model, as illustrated in Figure 4.14. The tool iterates through all of the rasters, with a different point filled. The

ModelBuilder flowcharts can be found in Appendix B; 11.7- 11.9. The scripts can be found in 15.7 to 15.7.10.

5.6.1 Flow direction

The flow direction was calculated using the D8 method. ArcGIS Pro has three built-in methods, D8, MFD and DINF. The main difference between the three is that the D8 will only drain to one downstream cell, while the MDF and DINF has the ability to drain to several downstream cells. All methods were tried and evaluated. The different outputs were compared to the FKB water area polygons, and all of them matched up well, when sampling random locations. The raster has been constructed with surface constraints and the stream channel has been lowered, so results were to a large degree the same, and therefore the choice fell on the simplest of the methods (D8) since this is also the lightest process to run. The result of the flow direction tool is a raster where all cells are assigned a flow direction, these ranging from 1-128 as illustrated in Figure 5.5 below. ModelBuilder chart can be found in Appendix B; 11.7.4.

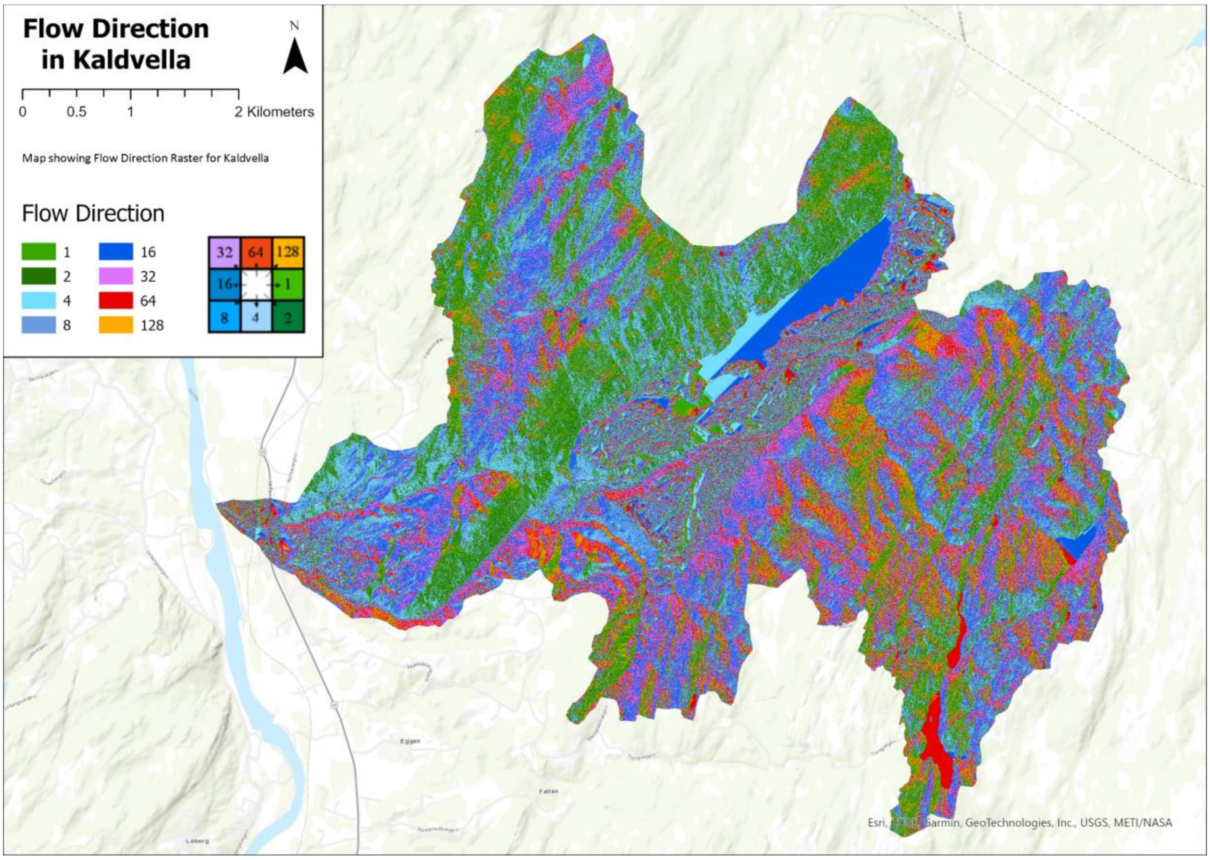


Figure 5.5: Flow direction map of Kaldvella.

5.6.2 Flow accumulation

The flow direction tool makes no sense to run by itself, since the output is chaotic and hard to read. The Flow accumulation tool illustrates how water would accumulate in the terrain, and the direction of flow the water would have on a surface. The cell value of the output flow accumulation raster is equivalent to the number of upstream cells draining through the selected cell. This value can be multiplied with the cell size of the raster to find the accumulated area draining to the individual cells. The output is a drainage line in raster format where the cell value is equivalent to the number of upstream cells that will drain through the individual pixels (Figure 5.6). Using a precipitation weight-raster, it is possible to calculate the volume of the accumulated surface water. This makes it possible to model local precipitation patterns. Without a precipitation raster, the tool will assume that equal amounts of precipitation will fall in the entire watershed. Kaldvella has no measurement stations, so in this study weight-rasters will not be used to model water volume. ModelBuilder flowchart can be found in Appendix B; 11.7.5. Weight rasters can also be used to model the impeding effect different surfaces have on the water flow. A short script for producing a weigh raster from areal use data, inspired by values used by Kielland (2020) can be found in Appendix B; 11.10.

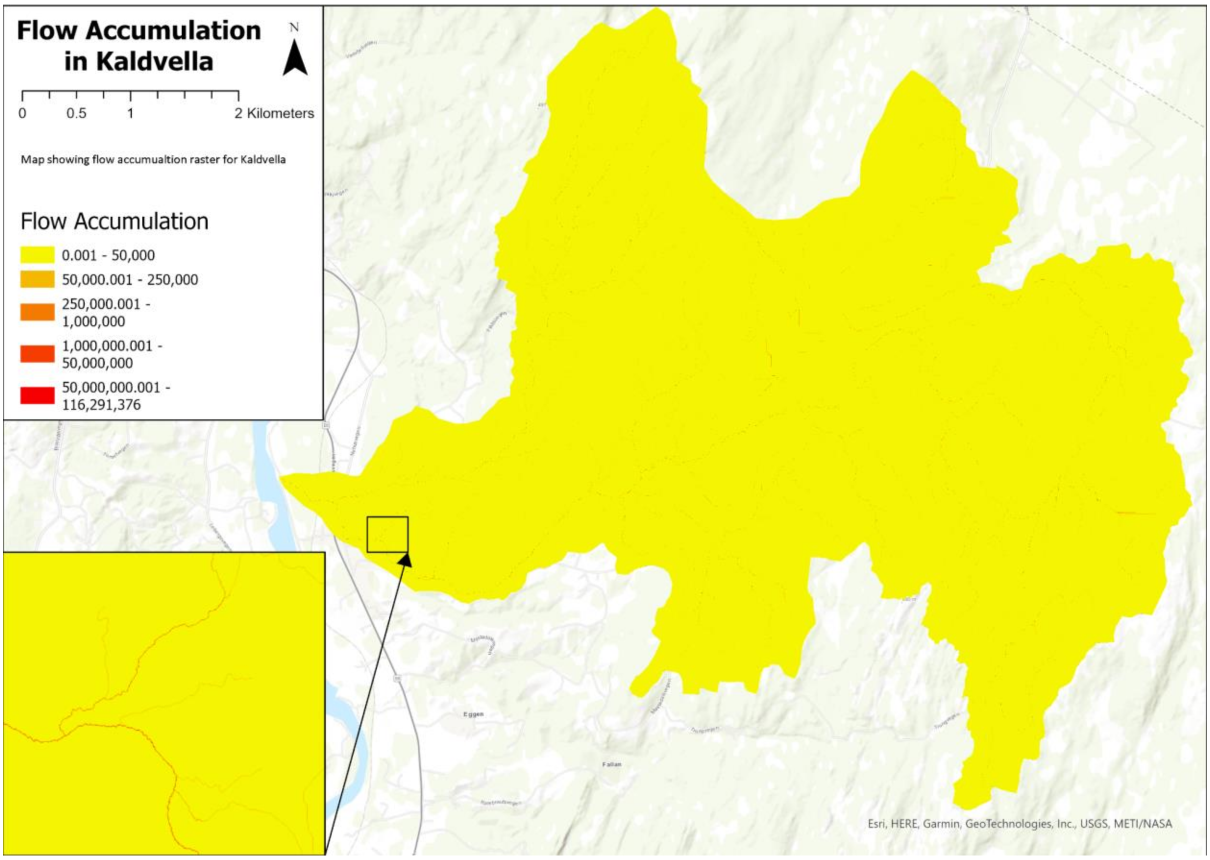


Figure 5.6: showing Flow Accumulation in Kaldvella.

5.6.3 Damage Points

Using the accumulated area and intersecting the water path with infrastructure such as buildings and houses makes it possible to make assumptions about the damage potential at a given location. This process has been developed into a tool. The ModelBuilder chart can be found in Appendix B; 11.8. The script for the tool can be found in Appendix E; 14.8.

5.6.4 Threshold values

To be able to identify and classify flood paths a threshold has to be defined, to separate flood prone river stretches, from safe stretches. To define the threshold value for flooding, Meiforth (2013) received assistance from hydrology experts (Sellæg, 2016). The analysis was based on previous flood events and known flood damage in Trondheim. The conclusion from this analysis was that 50,000 m² of accumulated area was the critical limit in urban areas, while non-urban areas such as parks and forests had a threshold value of 25,000 m² (Sellæg, 2016; Meiforth 2013). The Kaldvella watershed is dominated by non-urban areas, and therefore a lower limit of 25.000 was selected. The classes were divided into manual intervals inspired by the values used by Meiforth (2013) and Sellæg (2016):

Damage class 1. 25 000m² -50 000 m²	Damage class 2. 50 000 m² - 250 000 m²
Damage class 3. 250 000 m² - 1 000 000 m²	Damage class 4. 1 000 000 m² - 5 000 000 m²
Damage class 5. 5 000 000 m²-25 000 000 m²	Damage class 6. Over 25 000 000 m²

Table 5.1: showing the damage classes, inspired by (Sellæg, 2016; Meiforth 2013)

The purpose of the class division is to separate the flood paths of differentiating sizes, in relation to the accumulated area. The damage potential of a flood path is expected to correlate with the accumulated area. It is expected that a flood path with a larger contributing accumulation area will have a greater damage potential than a smaller one. To exemplify, one can assume that 5,000,000 m² of accumulated area will have a much greater damage potential than a flood path with 50,000 m² accumulated area. Using a weight raster, it is possible to find the amount of accumulated rainwater that will drain through a cell. Weight rasters will not be used in the thesis for finding water volume. Using weight-raster to simulate the impeding effect different surfaces will have on the water was considered, but it was decided to not include this, since this will not make much sense when not combined with a known amount of precipitation.

The damage classes (1-6) have also been used to set the symbology size. 1 being the lowest and 6 being the highest damage class. One (1) being the lowest amount of accumulated water capable of causing damage, and six (6) reflecting the total size of the watershed. The size of the Kaldvella watershed is 29.7 km² (297 00 000 m²). When running a water delineation analysis, the output from Kaldvella into Gaula is 290 72 844 m². Assigning damage classification has been developed into a tool. The ModelBuilder flowchart can be found in Appendix B; 11.9. The script for the tool can be found in Appendix E; 14.9.

5.6.5 Depressions

Here the study has utilized "Bluespot mapping", a method developed by the municipality of Copenhagen in Denmark to identify areas sensitive to flooding. Bluespots are depressions in the DEM that have an inlet of water, but no outlet. In a situation of intense precipitation or flood water taking alternative pathways these depressions might fill with flood water, causing damage to buildings by water intrusion (Balstrøm, 2022). The Bluespot_Mapping is run once, and the floodpaths are linked to the bluespots by selecting by location, using intersect. The model used in this study is inspired by Balstøm and a GIS exercise by prof. Rød at NTNU, department of Geography. The ModelBuilder flowchart for the Bluespot:Mapping model and Iterate_Bluespot_ID can be found in Appendix B; 11.11 & 11.12. The scripts for the models can be found in Appendix E; 14.10 & 14.11.

6. Results

The results presented in this Chapter include the testing of the models by running them on the Kaldvella watershed. The results from the model will be presented and compared to HEC-RAS flood maps made by Graf (2021) and Gomez (2021). The output produced by the models and the discussion around the results are meant to answer the research questions presented in Chapter 1.3.

The main objective of this thesis is to develop methodology that will indicate downstream damage potential from clogged critical points. To investigate this issue, a set of tools have been developed (which must be deployed in the intended order) to produce valid results. The steps are numbered to be as intuitive as possible.

The results Chapter is divided into three parts. First, the results from the terrain tools will be presented, showing the LiDAR derived raster, as well as the modifications that have been made to the terrain raster. Secondly the input and the output from the models will be presented. Lastly the output from the model will be compared to flood maps by Graf (2021) and Gomez (2021) to evaluate if the model output presents realistic scenarios.

6.1 LiDAR to raster

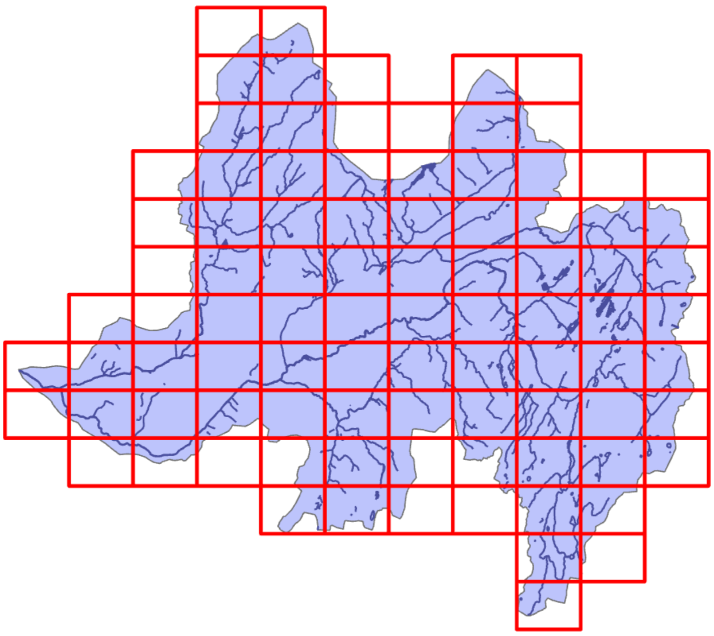


Figure 6.1: Kaldvella LAS files

The Kaldvella watershed consists of a total of 91 LAS files (Figure 6.1). These files are combined in a LAS dataset.

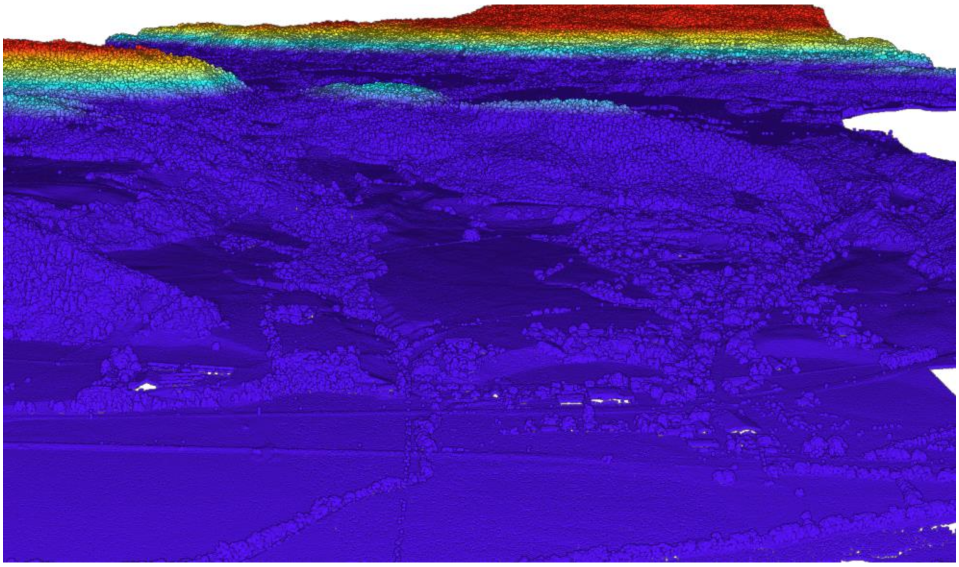


Figure 6.2: Showing the LAS data before non-ground points are filtered out.

The raw LiDAR include everything captured by the laser beams, resulting in a DSM (Figure 6.2). The DEM used for analysis in this study is based on the ground points. The above-ground points are filtered out to leave the bare earth points, as seen in figure 6.3. In areas where coverage is poor, interpolation is very important, since different interpolation methods produce different results.

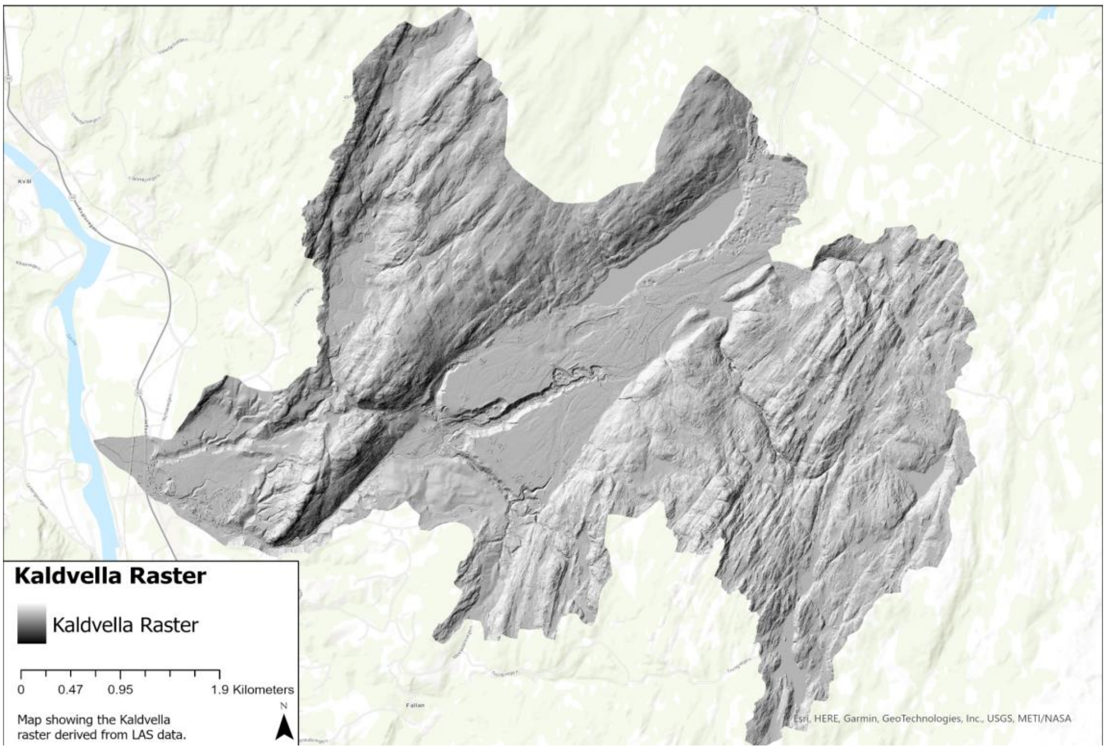


Figure 6.3: Showing the Kaldvella raster derived from LiDAR. A hillshade filter has been applied to help aid visualizing the topography.

The resulting DEM (Figure 6.3) is a raster with a resolution of 0.5 meters, as discussed in Chapter 5. The resolution was chosen based on the point density of the LiDAR data.

6.2 Raster modifications

Buildings

Figure 6.4: Comparing buildings included in the FKB building data from 2021 (left) and LiDAR data from 2015 (right).

As seen in Figure 6.4, the number of buildings in the LiDAR data do not match those in the FKB building data. A choice was therefore made to rasterize the FKB buildings and add them to the DEM.

Floodplain

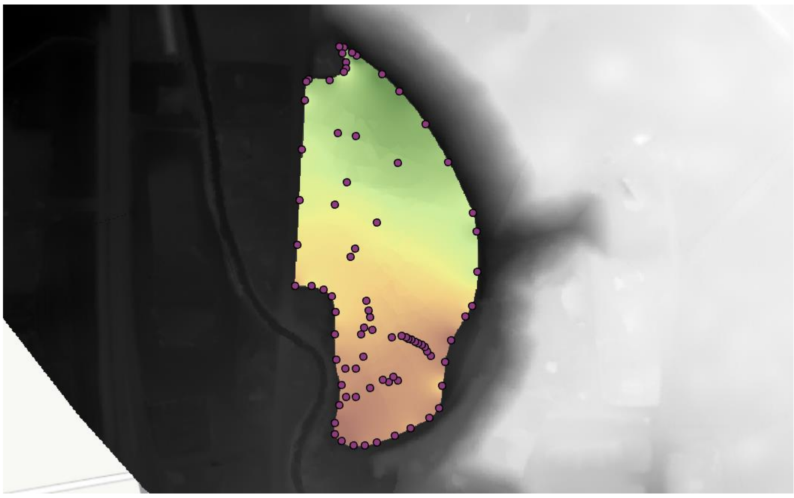


Figure 6.5: illustrating the interpolated raster surface.

The floodplain has also been raised by ~1 meter, based on GPS points collected during fieldwork; this is illustrated in Figure 6.5, where brown corresponds to high elevation and green corresponds to lower elevation. This modification causes the water to flow differently compared to the original LiDAR data, which better reflects the current situation. The exact changes in meters can be seen in Figure 5.3. in Chapter 5.

Stream channel

In flood mapping applications it is important to be able to produce realistic results. Figure 6.6 illustrates the flow accumulation (illustrated in red) compared to the FKB water area (blue). When comparing them the water accumulation output matches up well with the FKB data. To get to this point, a number of changes has been made to the DEM that affect the output of the water accumulation raster. These included lowering bridges, adding levees, and adding culverts to the DEM. The results from these modifications are presented below (see Chapter 6.1.2 - 6.1.4).

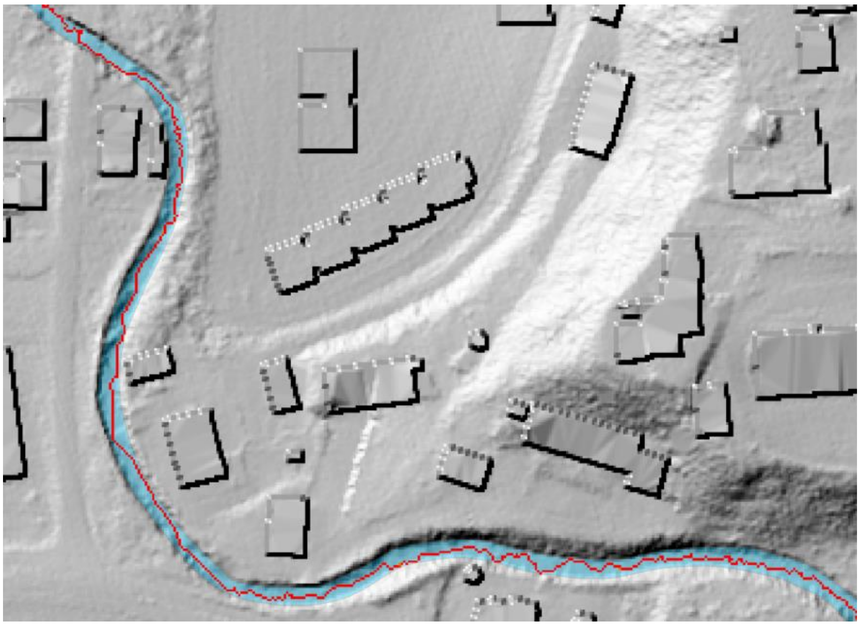


Figure 6.6: Illustration of the FKB water area (blue) compared to the Water accumulation raster (red).

Bridges

Within the Kaldvella watershed there are a total of 30 registered road bridges. These are illustrated in red in Figure 6.7. To ensure that the water flow is not blocked by the bridges, these have been removed.

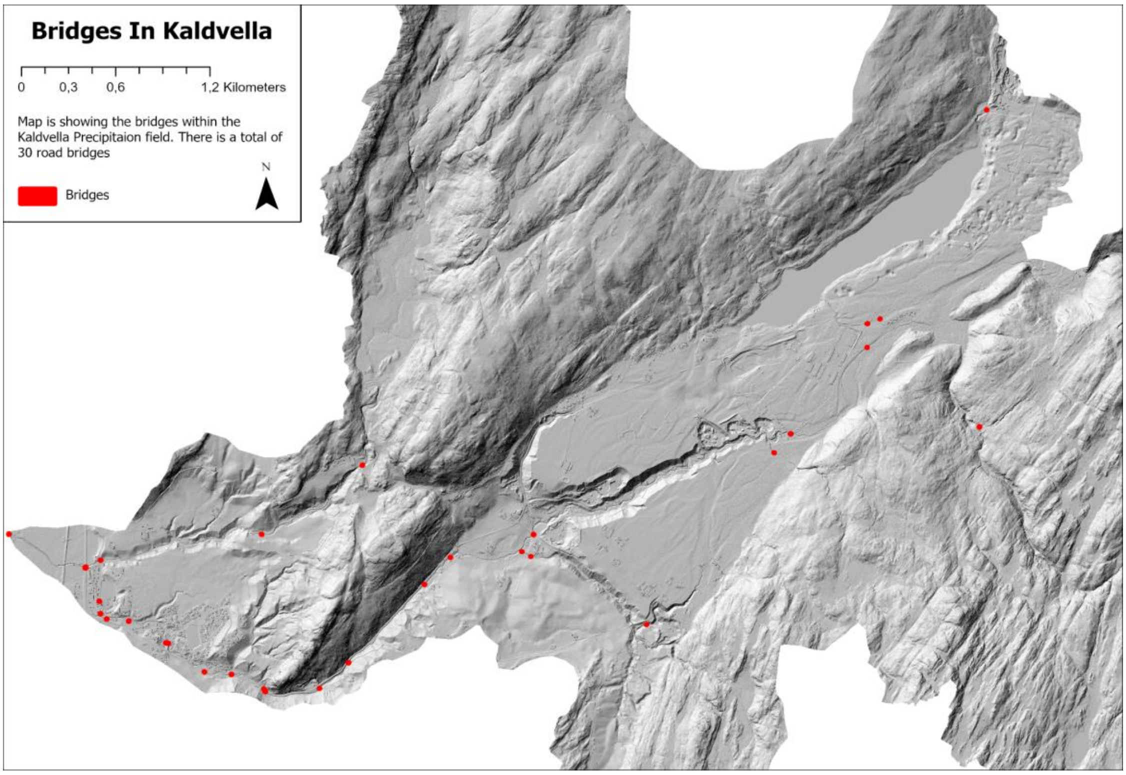


Figure 6.7: Road bridges in Kaldvella.

A selection of bridges is illustrated in Figure 6.8 to show how the DEM looks before and after the removal of bridges. Any remnants of the bridges are completely gone, and the water flow is not in any way disturbed or obstructed by the bridges.

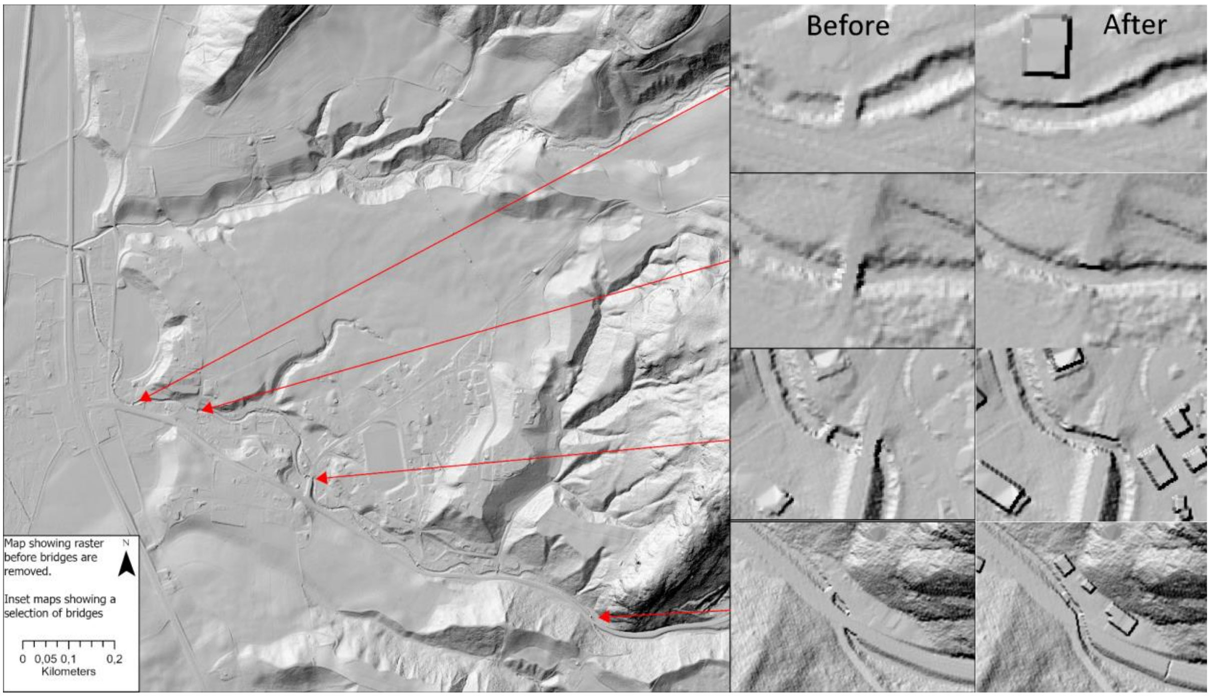


Figure 6.8: Showing raster before and after removing bridges from DEM.

Culverts

As with bridges the raster cells that coincide with culvert locations have been dropped using the values from the variable KOORDH. These are the elevation values of the culvert. As illustrated in Figure 6.9 a small area can have many culverts, this highlights how important it is to include the culverts in the analysis. A sample of culverts is illustrated in Figure 6.10 to help illustrate how the DEM looks after the modifications were made.

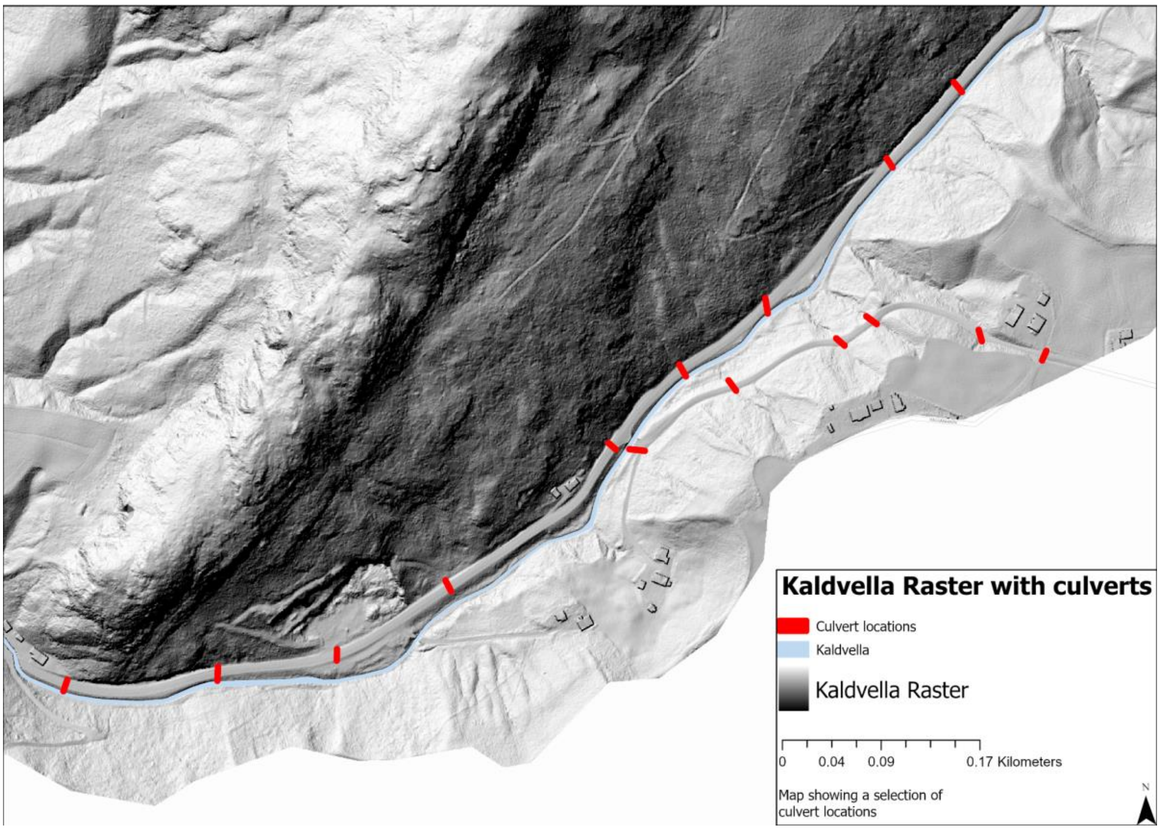


Figure 6.9: Showing a selection of culverts in the Kaldvella watershed.

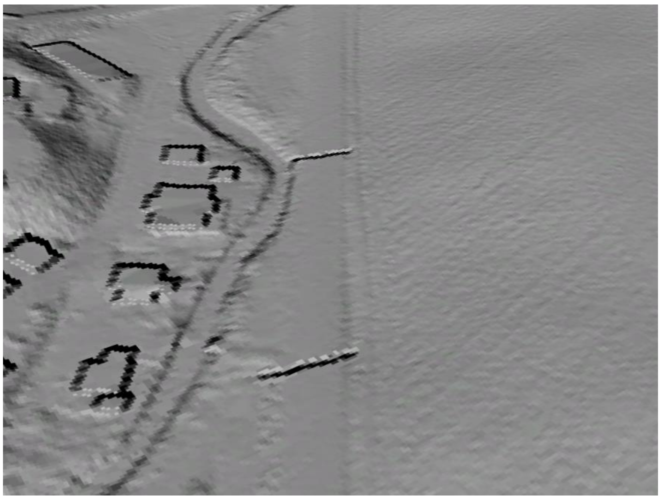


Figure 6.10: Illustrates the output from the workflow for adding culvert to the DEM. The raster cells coinciding with the culvert locations have been lowered, using the values from the attribute KOORDH (coordinate height).

Levees

In recent times levees have been added to help prevent the Kaldvella from flowing onto the floodplain. Since these are more recent than the LiDAR data, the levees were added to the DEM from GPS points to better represent the actual terrain. The levees are illustrated in Figure 6.11.

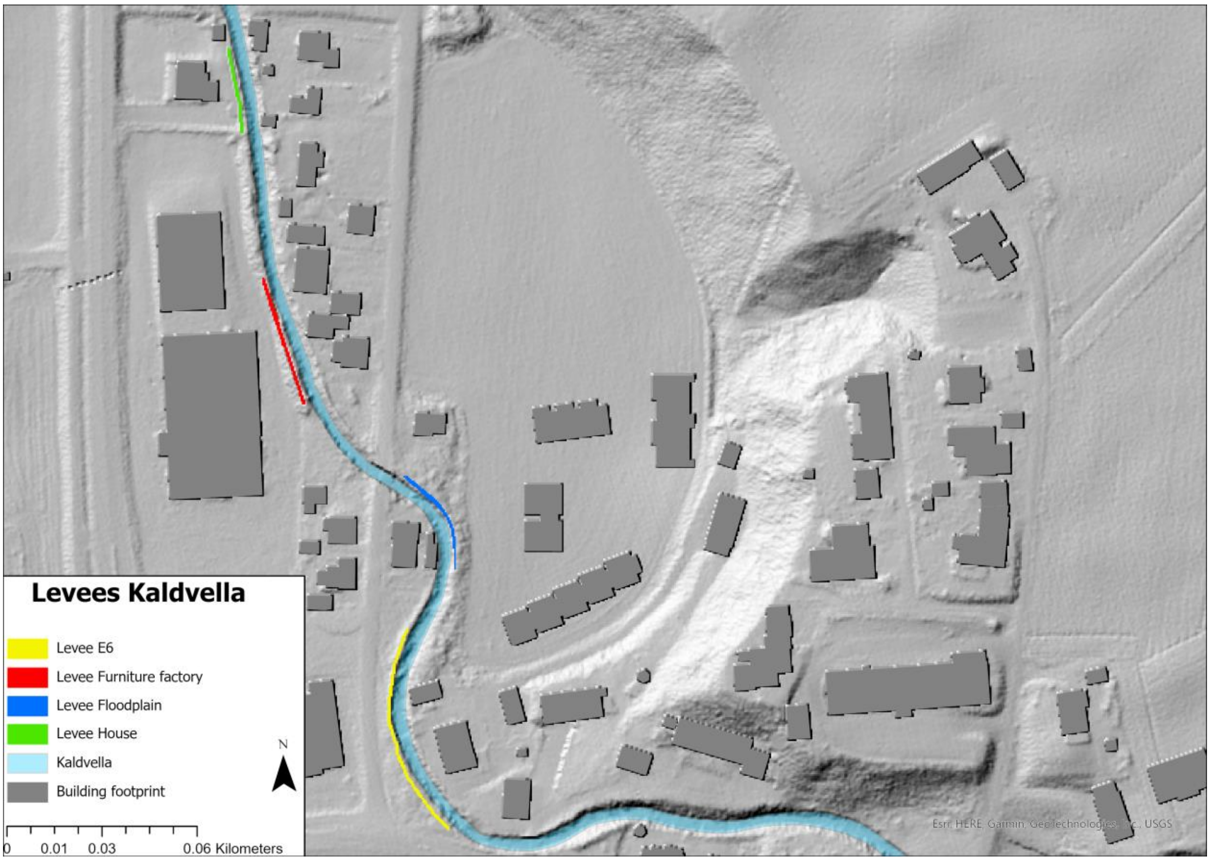


Figure 6.11: Showing levees that have been added, based on collected GPS points.

6.3 GIS Hydrological Modeling results

The input data were carefully chosen with flexibility in mind, since the tools are supposed to work with a range of different watersheds. On this basis the FKB datasets were chosen, as they are available all over Norway. The LAS data with the finest resolution was selected to reach the highest possible level of accuracy. Culverts and bridges were downloaded from NVDB, the national database for road data. The bridge data was insufficient, and many bridges within the watershed were not included in the data. This has been supplemented by intersecting water and road data, and then visually inspecting the results by comparing them to aerial photos, to ensure that they are correct. The culvert data are more difficult to verify on the map, since they are not visible in aerial photos.

The model is set up to run two scenarios, with and without critical points filled. This is both to compare the two outputs, as well as map water paths that have the potential to cause damage. The output is damage caused to buildings, roads, and railway. In this study the railway is not in danger of being flooded by Kaldvella.

In this study the main emphasis is on critical points, and the damage critical points can have. The output from a running model with no critical points filled, might be useful for pointing out critical points. The results from running the model with open flood paths will be presented briefly, but not discussed in detail, since the main emphasis of this thesis is damage caused by critical points. A general conclusion will be drawn from the result of running the tools with open paths, but individual points will not be discussed. The table showing the indicated damage can be found in Appendix C.

6.3.1 No critical point filled

In this chapter the result from running the tools without critical points filled is presented. The output indicates that the water has great damage potential, despite following its natural channels. It is still important to note that the output does not take local variations in precipitation into account, neither the different impeding effects that different surfaces, such as roads, farmland or forest may have. As discussed previously, precipitation data was excluded, since the resolution of the data is not high enough. As a result, the model assumes that all precipitation will accumulate as surface water, which is to some degree, a special situation, but not unlikely. This could happen if, for example, an extreme precipitation event were to occur at a time when the storage capacity of the watershed is filled, e.g., after a long period of rain, during cold periods when the ground is frozen, or during the spring season when the magazines in the watershed are filled by melting snow. In these situations, most of the precipitation is likely to accumulate as surface water. Using a weight raster to simulate the impedance of various surfaces was considered and tested, but this was chosen to be excluded as well. A short script for producing a weight raster from areal use data, inspired by values used by Kielland (2020) can be found in Appendix B; 11.9.

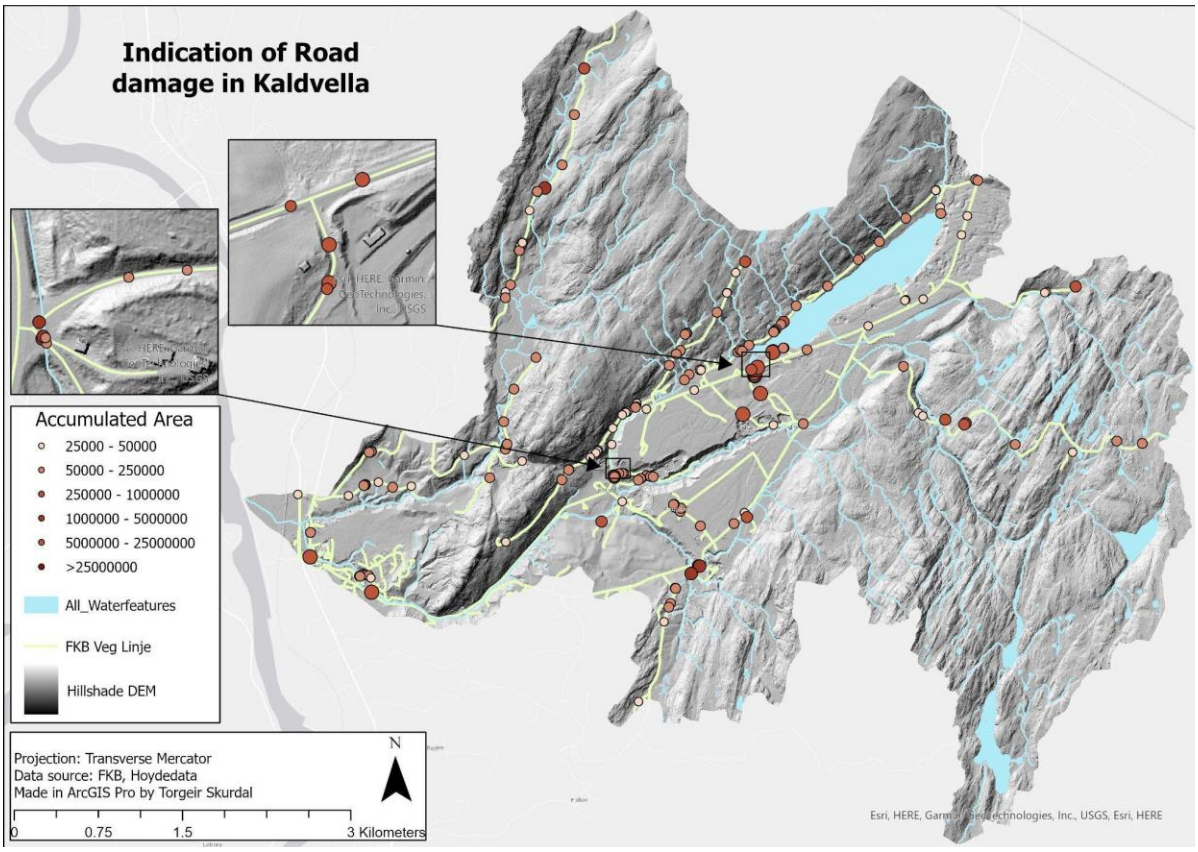


Figure 6.12: Road points that might be sensitive to water damage, with no critical points filled. These are locations with contributing areas larger than 25 000 square meters.

Indicated road-damage results from running the model with open flood paths will not be discussed in detail, since the thesis will focus on the damage potential of the critical points. The table with results from Figure 6.12 can be found in Appendix C; 12.2.

In Figure 6.13 many of the points are highlighted, these are locations that intersect with the FKB Water datasets, these locations might be culvert locations.

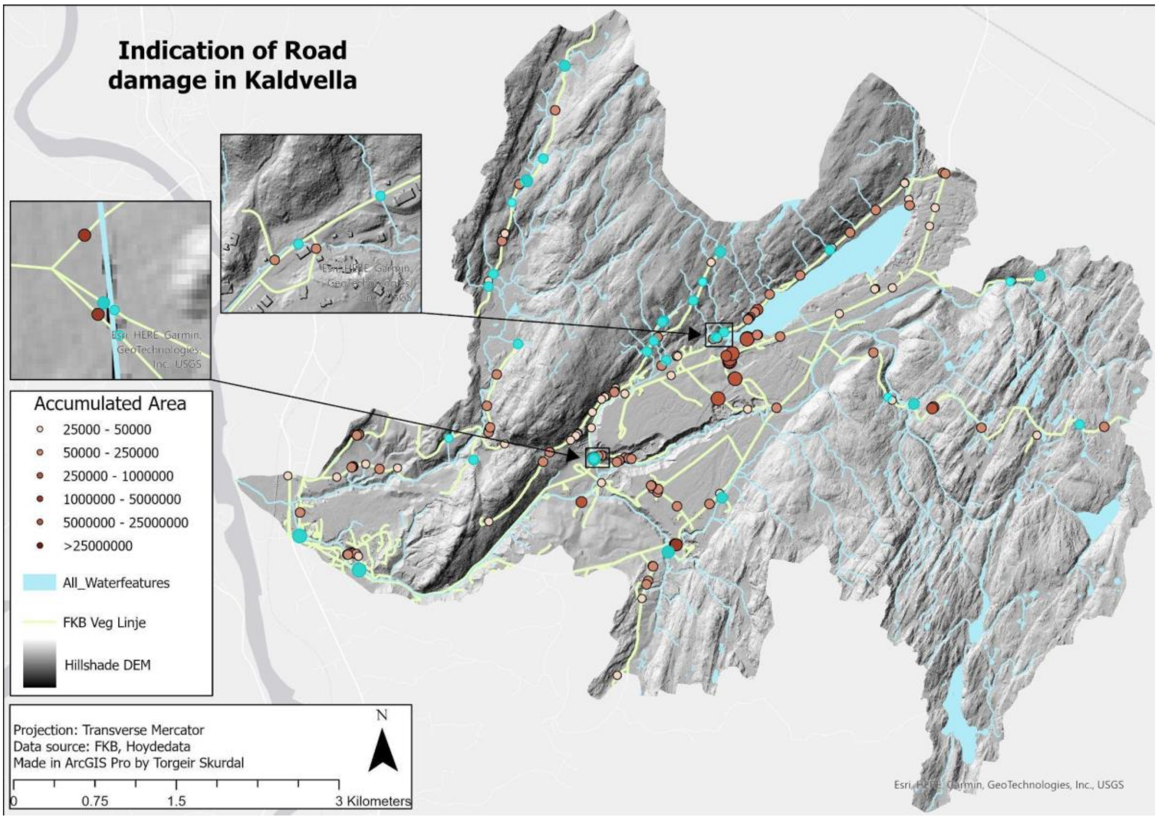


Figure 6.13: Illustrating points in roads that might be sensitive to water damage, with no critical points filled. The points intersecting FKB vann are highlighted. These points are associated with uncertainties, since these might be culvert locations that are not included in the culvert dataset.

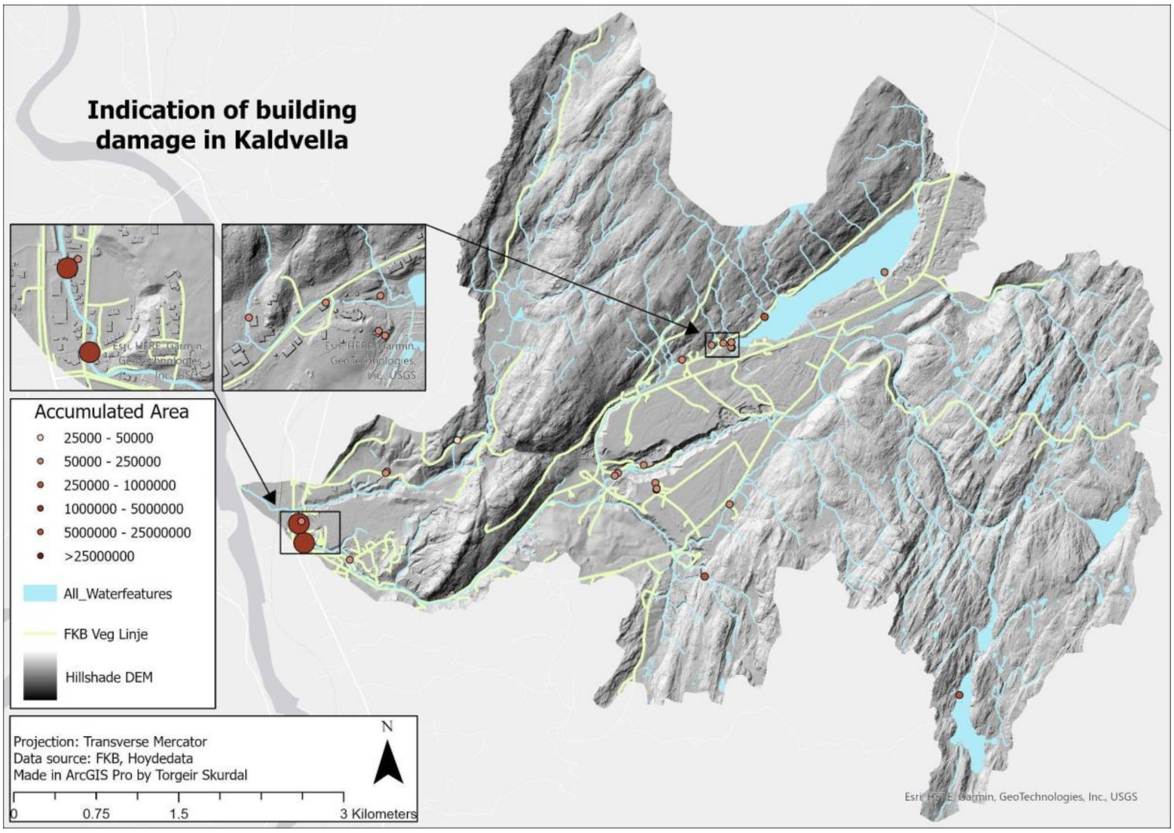


Figure 6.14: Illustrating buildings that might be sensitive to water damage, with no critical points filled.

Indicated building-damage results from running the model with open flood paths will not be discussed in detail, since the thesis will focus on the damage potential of the critical points. The table with results from Figure 6.14 can be found in Appendix C; 12.1

6.3.2 Critical points blocked

The process of modeling water delineation with critical points filled consists of three separate tools, as presented in Chapter 5. The Generate_blockage_Tool, Iterate_Hydro_Tool, Iterate_Damage_Points and Iterate_Damageclass have to be run in the intended order to produce valid results. The critical points locations are illustrated in Figure 6.15. The critical points are a combination of known problem locations where the river is shallow and running relatively high in the terrain. Other points are bridge locations that might act as bottlenecks in a flood and cause the water to find alternative paths. It is important to note that the damage potential of all the “critical points” chosen in this study is related to some uncertainty, and that the points are partly chosen to test the performance of the models.

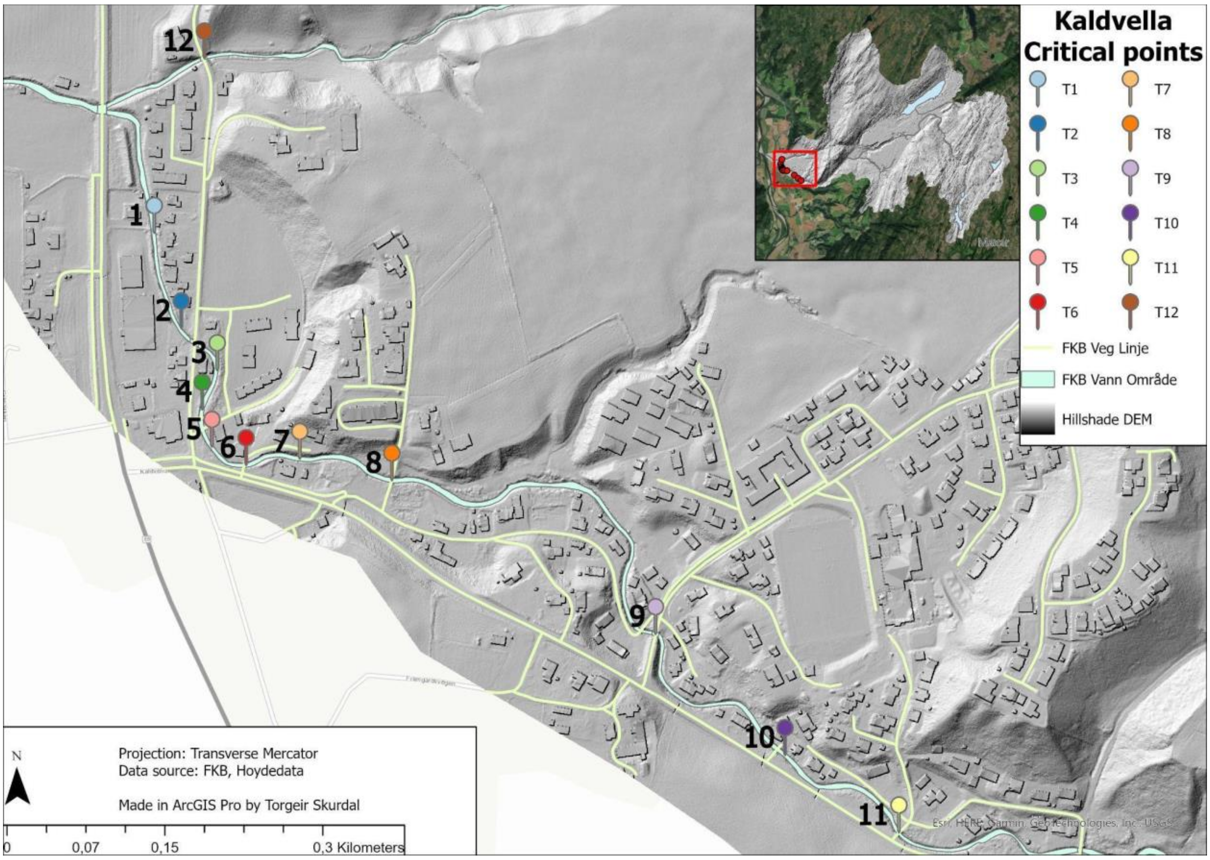


Figure 6.15: Showing an overview of the critical points used in this study.

Below are several maps illustrating what can happen if the critical point (illustrated in Figure 6.15) are blocked. The model loops through all the rasters and produces drainage lines, and calculations for accumulated area for each of the fill rasters. The model output will be compared to Flood maps by Gomez (2021) and Graf (2021). In maps comparing both building-and road-damage in the same map, buildings are illustrated with dots, while road damage are illustrated with triangles. For critical point locations where the indicated damage is extensive, the map would be overcrowded if everything was included so the building points and road points are illustrated in different maps (Figure 6.31 – 6.36). In this case all damage points are illustrated with dots. The size of the points is correlated to the size of the contributing upstream area.

Critical point 1

Figure 6.16: Showing the indicated damage caused to buildings by critical point 1, from running the model.

ORIG_FID	objtype	bygningsnu	LayerName	Acc_Area	Damage_Class
1	Bygning	184537494	T1	23205100	5
2	Bygning	184537508	T1	23206900	5
3	Bygning	184537443	T1	23208340	5
4	Bygning	184537486	T1	23204850	5
5	Bygning	184537532	T1	23204420	5

Table 6.1: Showing identifying output attributes for critical point 1 building damage.

The output from the model indicates that if critical point 1 is filled it has potential to cause great damage to several private properties, which is highlighted in Figure 6.16. The output attributes describing the damage class and the building numbers of the damage buildings are illustrated in Table 6.1. Looking at the bluespot in figure 6.16 It is also likely that water will accumulate in this area, causing water intrusion. According to the model output, to prevent damage, flood prevention measures in this location should be considered.

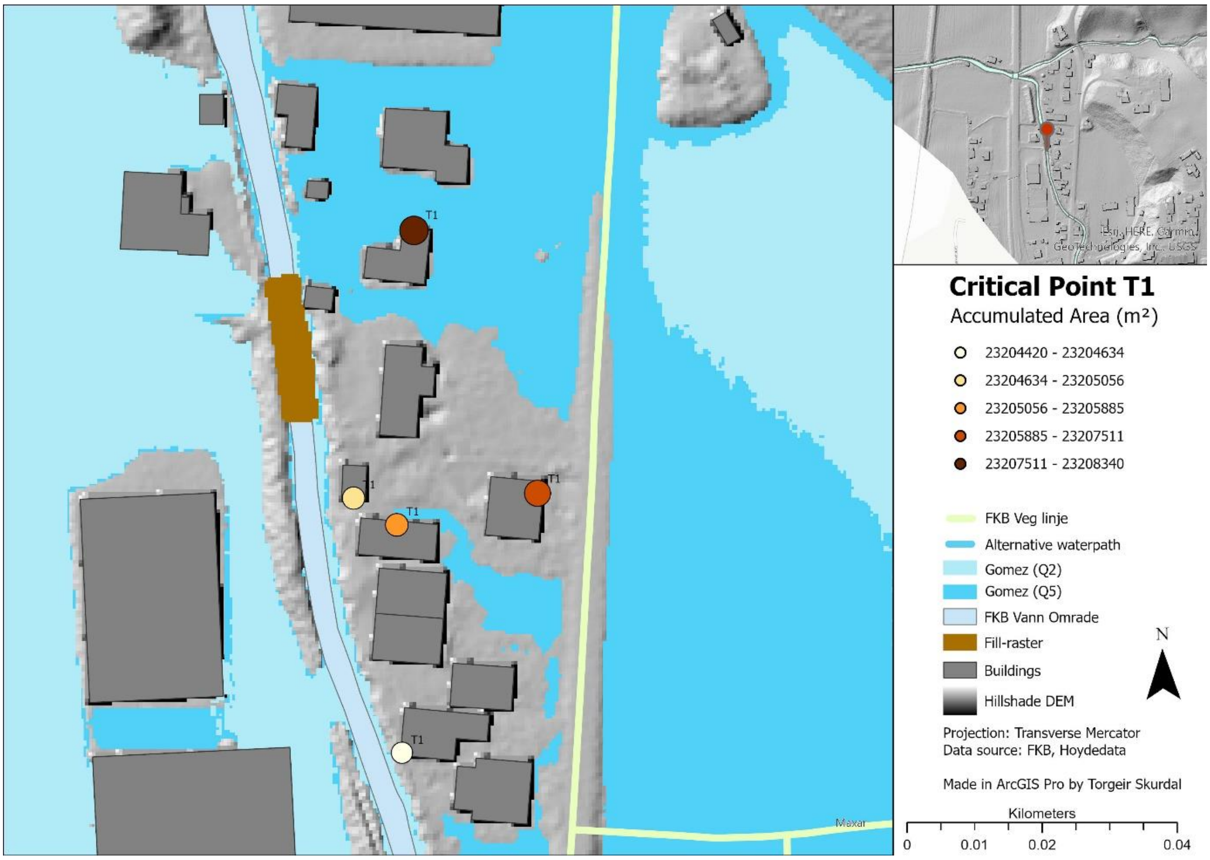


Figure 6.17: Showing the modeled damage caused to buildings by critical point 1, compared to Flood map by Gomez (2021)

As seen in Figure 6.17 showing the damage points compared to flood maps by Gomez, most of the damage points match up with the extent. The extent of the modeled bluespots, have a slightly lesser extent than the flood maps. Looking at the flood maps by Gomez (2021) the model output corresponds with a ~5-year flood.

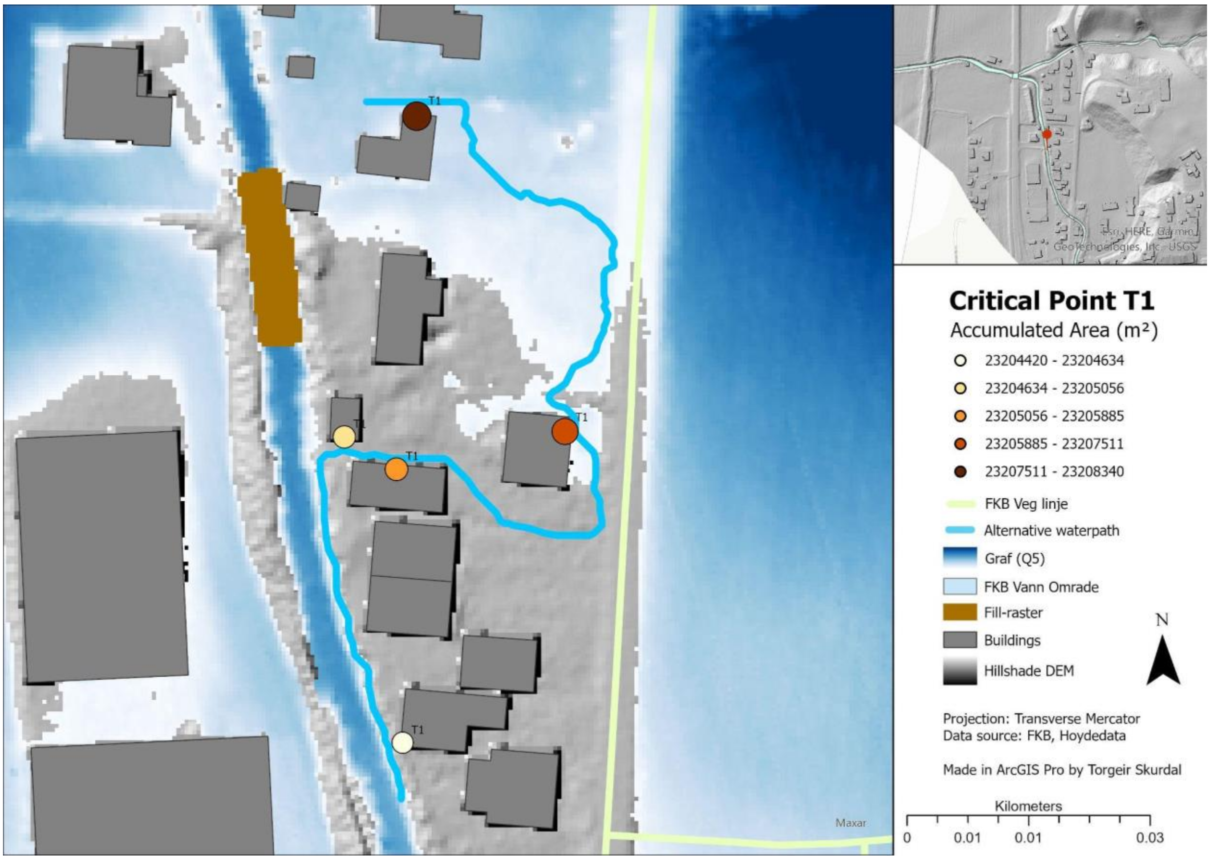


Figure 6.18: Showing damage caused to buildings by critical point 1, compared to Flood map by Graf (2021).

As seen in Figure 6.18 showing the indicated damage points compared to flood maps by Graf, about half of the damage points match up with damages in the flood map. The extent of the modeled bluespots, have a much lesser extent than the flood maps. Looking at the flood maps by Graf the model output, corresponds with a ~5-year flood.

Critical point 2

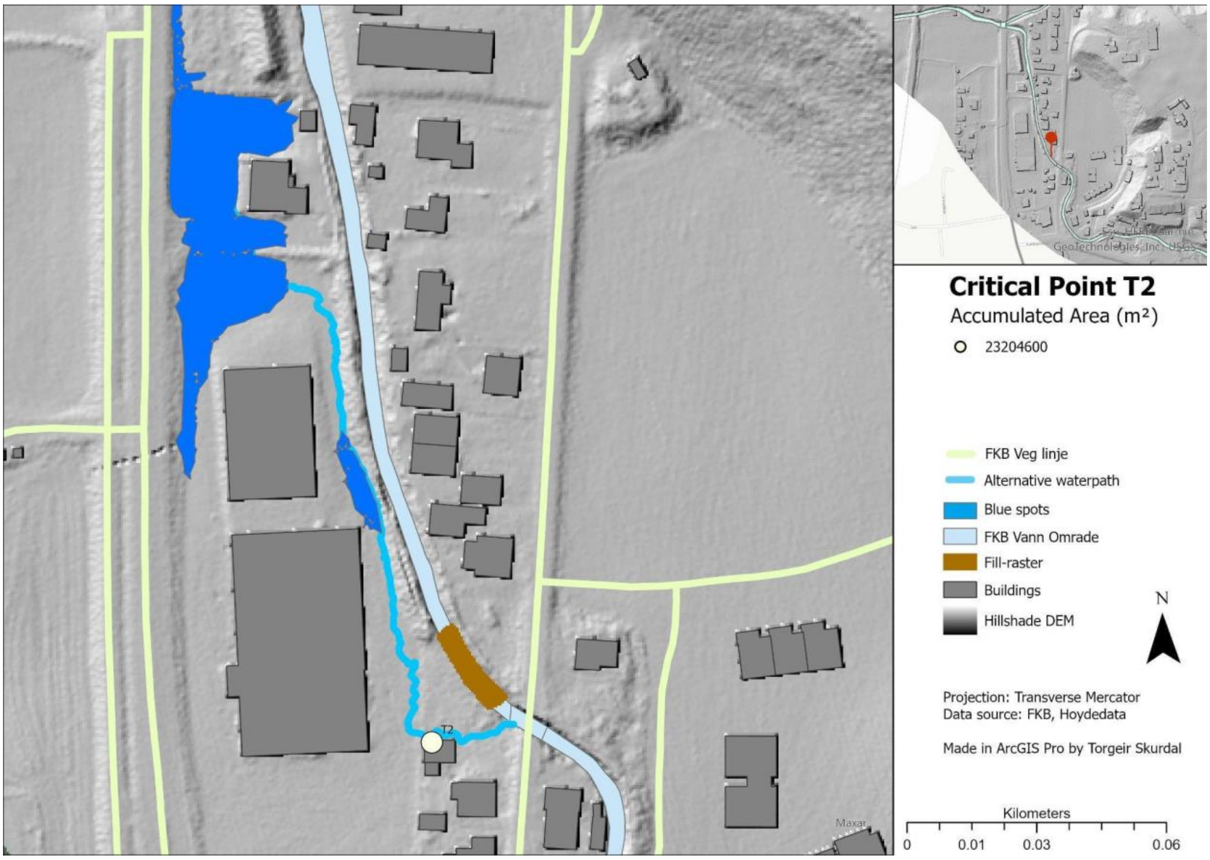


Figure 6.19: Showing the indicated damage caused to buildings by critical point 2, from running the model.

ORIG_FID	objtype	bygningsnu	LayerName	Acc_Area	Damage_Class
1	Bygning	184537605	T2	23204600	5

Table 6.2: Showing identifying output attributes for critical point 2 building damage.

If critical point 2 is blocked it has the potential to cause damage to a building (highlighted in figure 6.19). The output attributes describing the damage class and the building number of the damaged building can be seen in Table 6.2. To prevent damage, flood prevention measures in this location should be considered. The bluespot matches up with a 2-year flood but underestimates the extent.

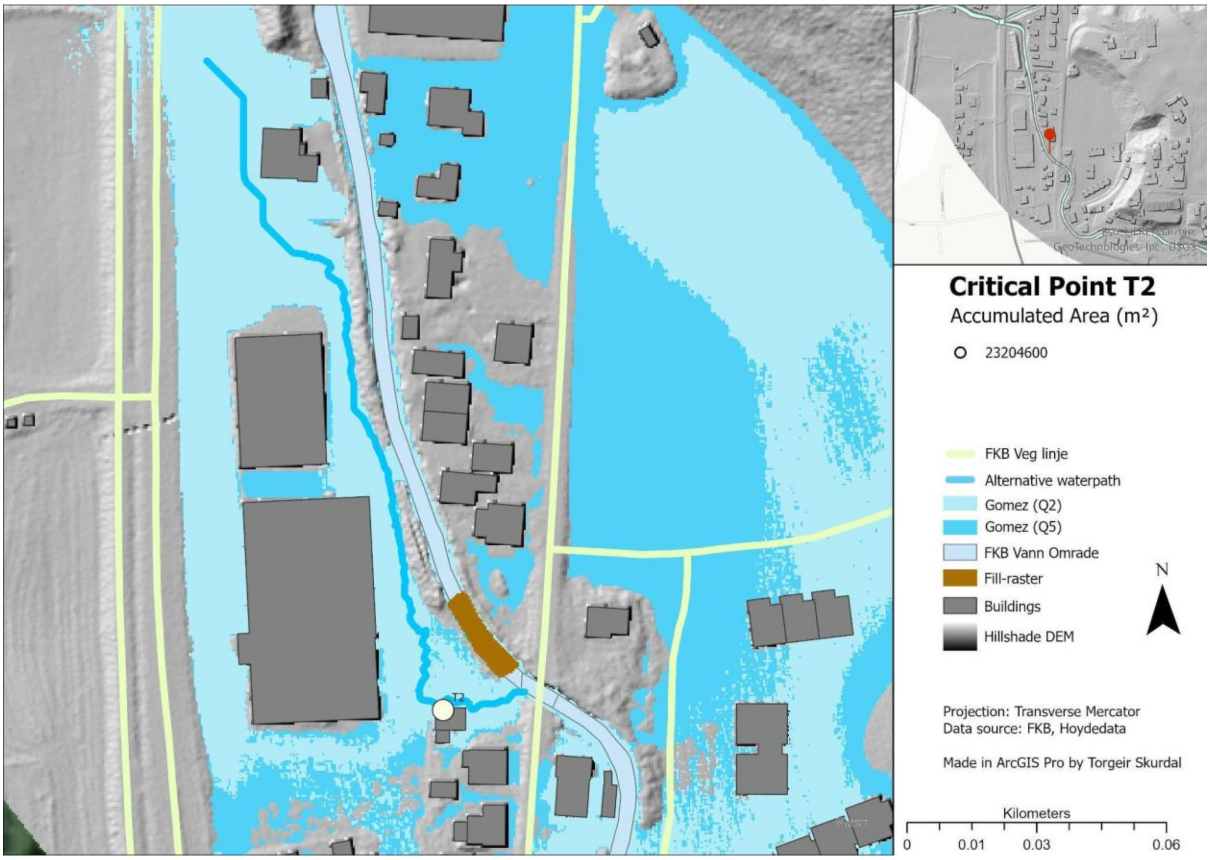


Figure 6.20: Showing the modeled damage caused to buildings by critical point 2, compared to Flood map by Gomez (2021).

Comparing the results from the model with the flood maps by Gomez (2021) (Figure 6.20), it is apparent that the model underestimates the extent of the flow path if critical point 2 is blocked. The output from the model corresponds to less than a 2-year flood. This underlines the sensitivity of this area.

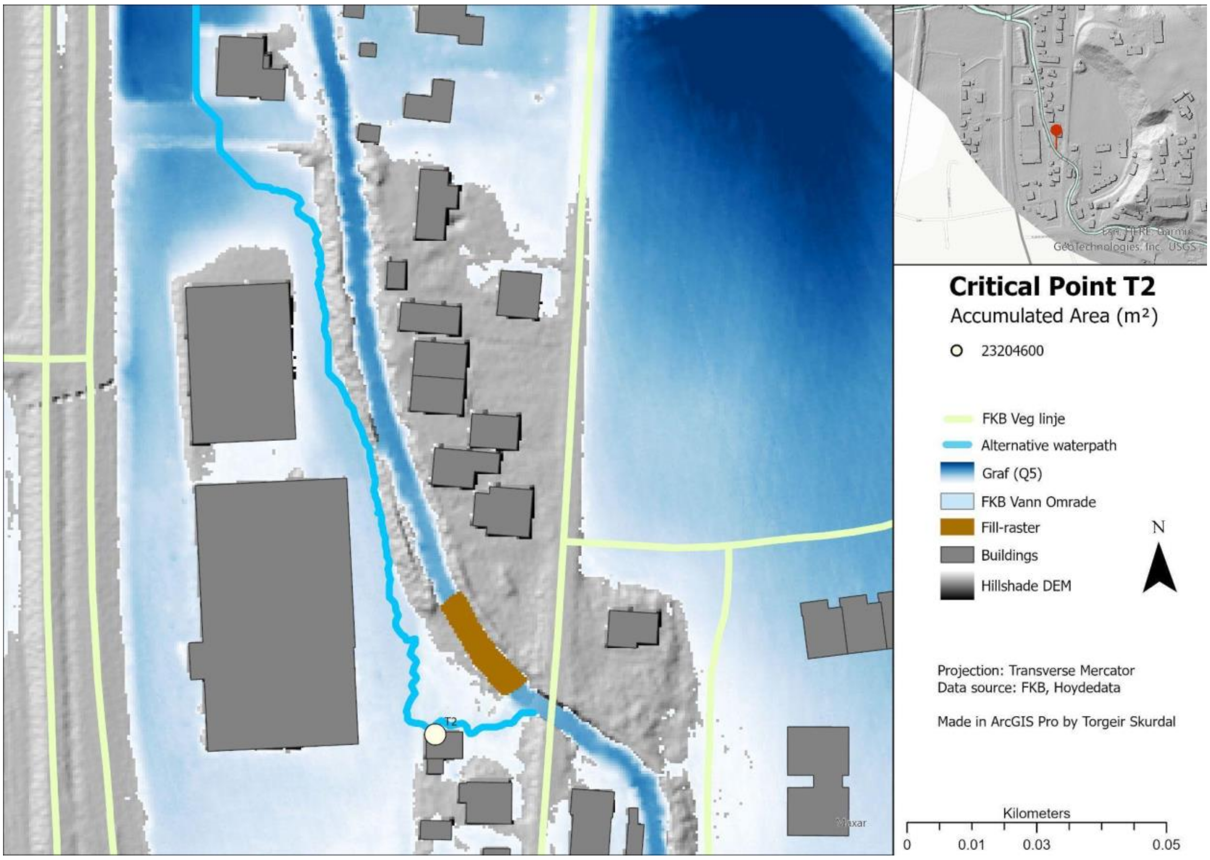


Figure 6.21: Showing the modeled damage caused to buildings by critical point 2, compared to Flood map by Graf (2021).

Comparing the results from the model with the flood maps by Graf (2021), illustrated in Figure 6.21 it is apparent that the model underestimates the extent of the flow path if critical point 2 is blocked. In the flood map by Graf a much larger area is flooded compared to the model output. Compared to map by Gomez (Figure 6.20) the extent of the flooded are very similar.

Critical point 3

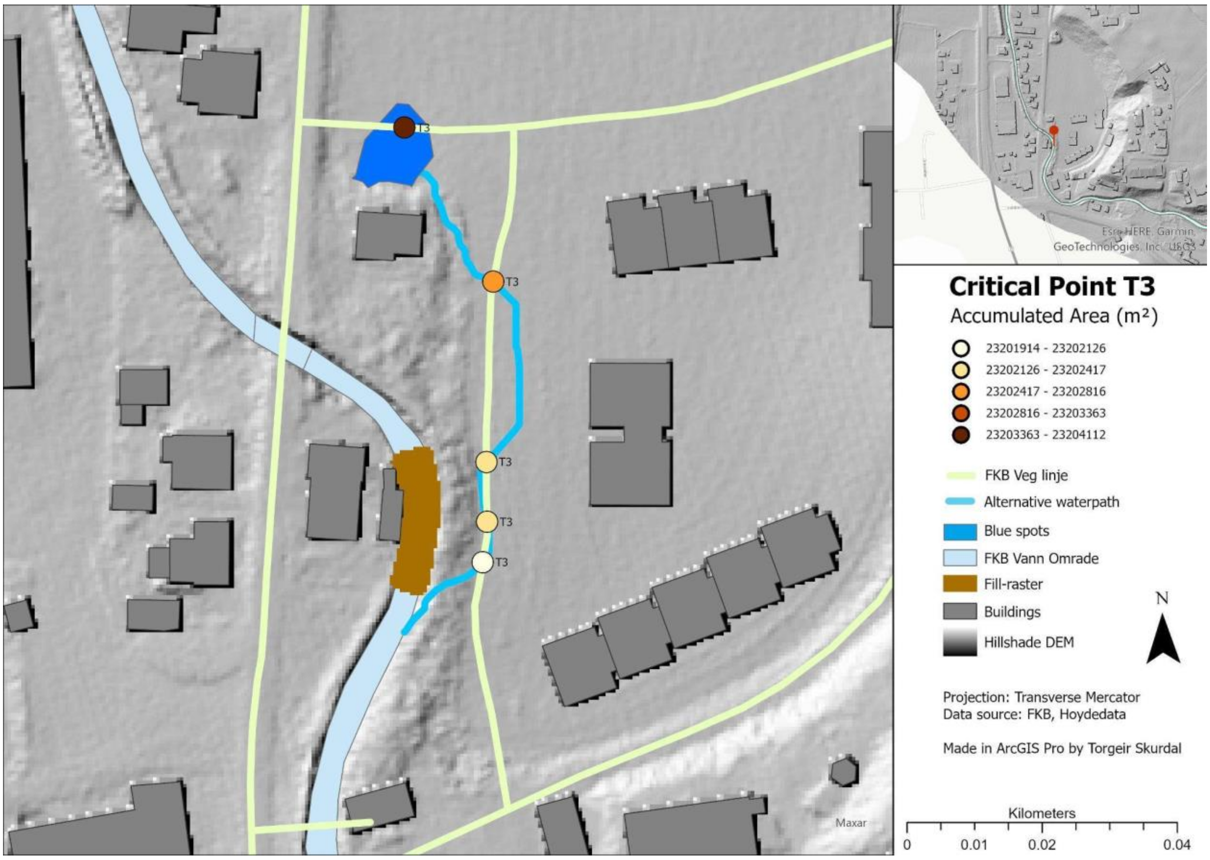


Figure 6.22: Showing the indicated damage caused to roads by critical point 3, from running the model.

FID_T3	GATENAVN	LayerName	Acc_Area	Damage_Class
2	Litjskjeet	T3	23204110	5
32	Litjskjeet	T3	23202430	5
44	Litjskjeet	T3	23202400	5
51	Litjskjeet	T3	23202220	5
57	Litjskjeet	T3	23201910	5

Table 6.3: Showing identifying output attributes for road damage caused by critical point 3.

If critical point 3 is blocked it has potential to cause damage to roads (highlighted in Figure 6.22), as water will flood over roads in this situation. The output attributes describing the damage class and the name of the road that may be damaged are illustrated in Table 6.3. To prevent damage, flood prevention measures in this location should be considered.

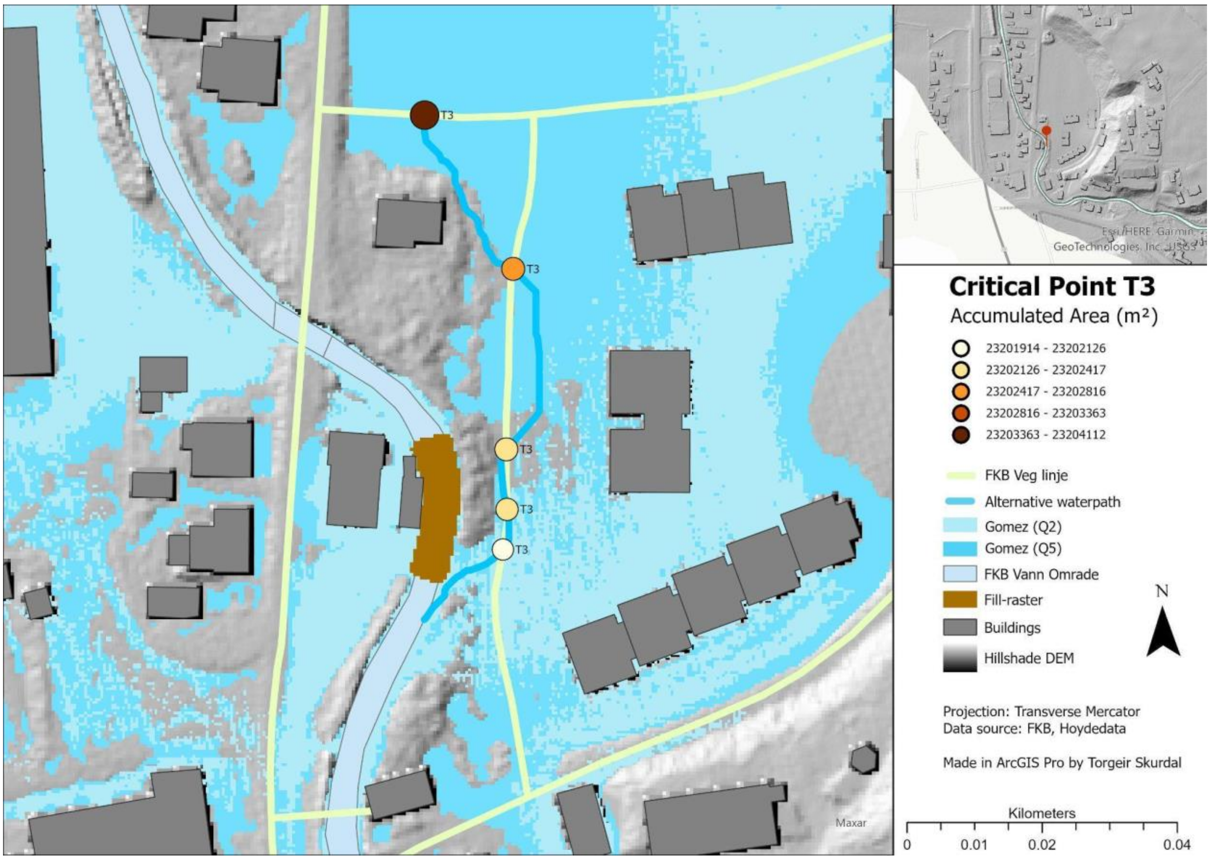


Figure 6.23: Showing the modeled damage caused to roads by critical point 3, compared to Flood map by Gomez (2021)

Comparing the Model output to the flood maps by Gomez (Figure 6.23), the damage corresponds to a 5-year flood. In the Q2 flood map by Gomez the water flows down towards the buildings, while the model estimates that the water will flow along the road, which might be a result of raising the terrain of the floodplain.

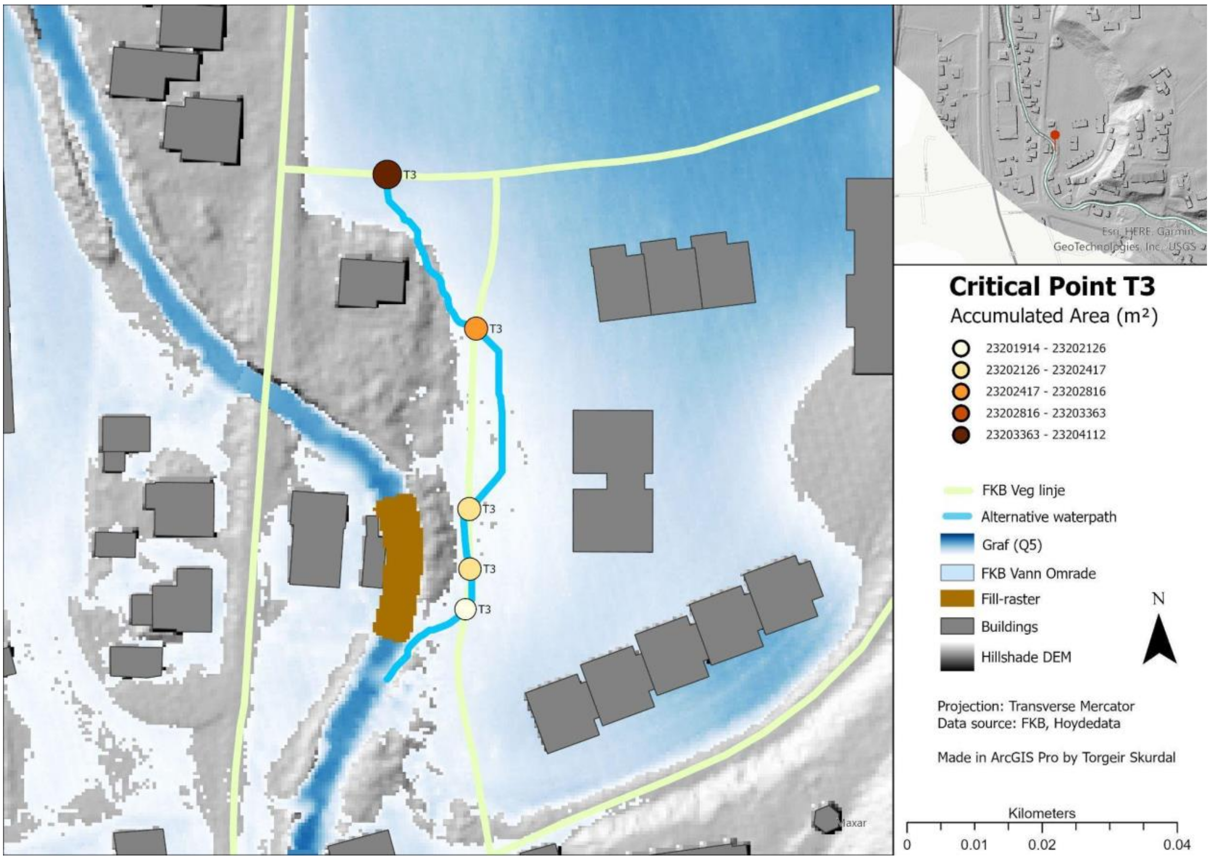


Figure 6.24: Showing the modeled damage caused to roads by critical point 3, compared to Flood map by Graf (2021)

Comparing the model output with the flood map by Graf (Figure 6.24) it is apparent that raising the flood plain influences the flood path. When comparing the damage this corresponds to the damage of a 5-year flood. Graf and Gomez (Figure 6.23) produce very similar extents in this area.

Critical point 4

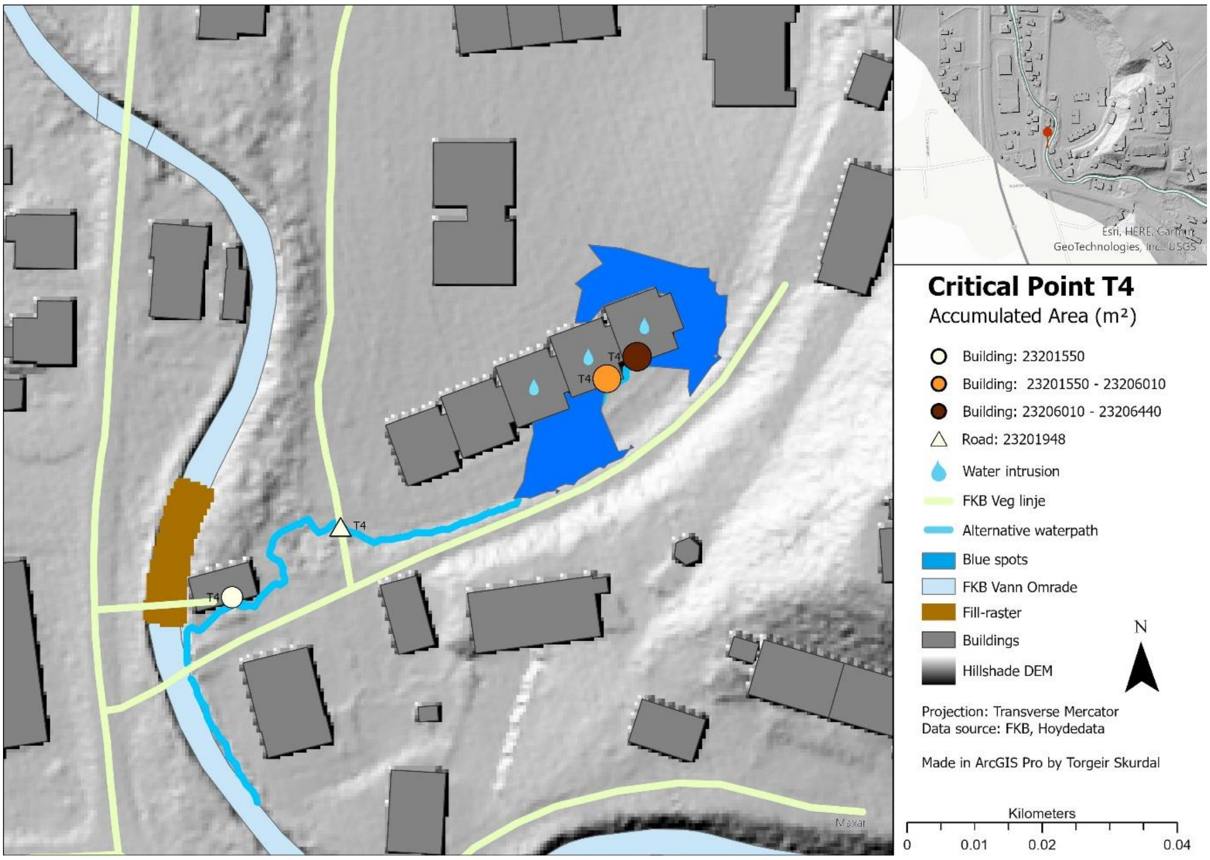


Figure 6.25: Showing the indicated damage caused to buildings and roads by critical point 4, from running the model.

ORIG_FID	objtype	bygningsnu	LayerName	Acc_Area	Damage_Class
1	Bygning	184537575	T4	23201550	5
2	Bygning	300790260	T4	23206440	5
3	Bygning	300790261	T4	23206010	5

Table 6.4: Showing identifying output attributes for critical point 4, building damage.

FID_T4	GATENAVN	LayerName	Acc_Area	Damage_Class
65	Litjskjeet	T4	23201950	5

Table 6.5: Showing identifying output attributes for critical point 4 road damage

If critical point 4 is blocked it has potential to cause damage to buildings and roads (Figure 6.25), as water will flood over roads in the case that this point is blocked. The output attributes describing the damage class and the names of the roads that may be damaged are illustrated in Tables 6.4. and 6.5. To prevent damage, flood prevention measures in this location should be considered.

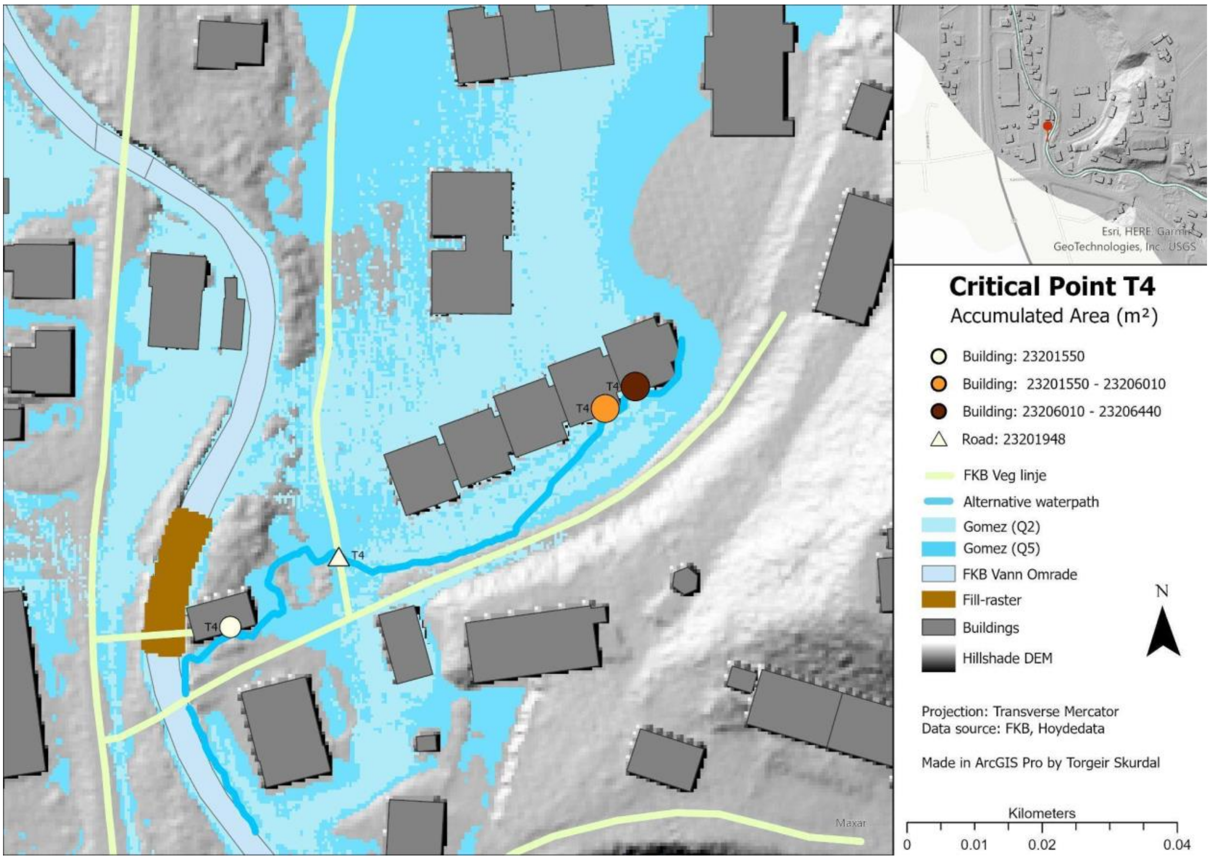


Figure 6.26: Showing the modeled damage caused to buildings and roads by critical point 4, compared to Flood map by Gomez (2021)

Comparing the model output with the flood maps by Gomez (Figure 6.26), it is pointing towards that the fill raster is placed wrong. If this location is filled, water will flow out towards the left of the map. In the model levees have been added along the riverbank to simulate the current terrain, so on the other hand, the levee can be the cause of the damages to the house. Still, this highlights how important the accuracy of input is for this model.

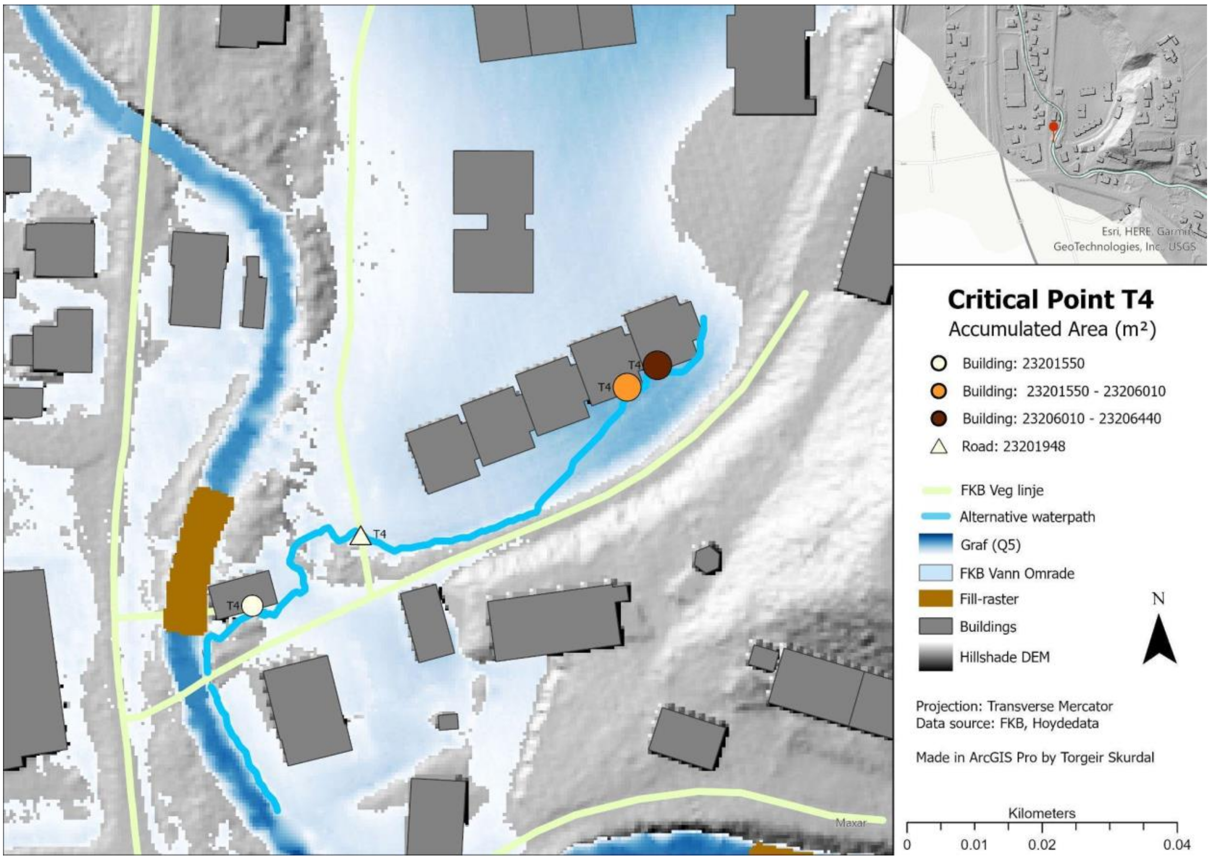


Figure 6.27: Showing the modeled damage caused to buildings and roads by critical point 4, compared to Flood map by Graf (2021).

Comparing the model output with the flood map by Graf (Figure 6.27), the flood path matches up with a 5-year flood, there are still uncertainties about if the water will flow out towards to the left or right first. In this bend there has constructed flood levees, and this might be the cause of the damage to the building closest to the stream channel.

Critical point 5

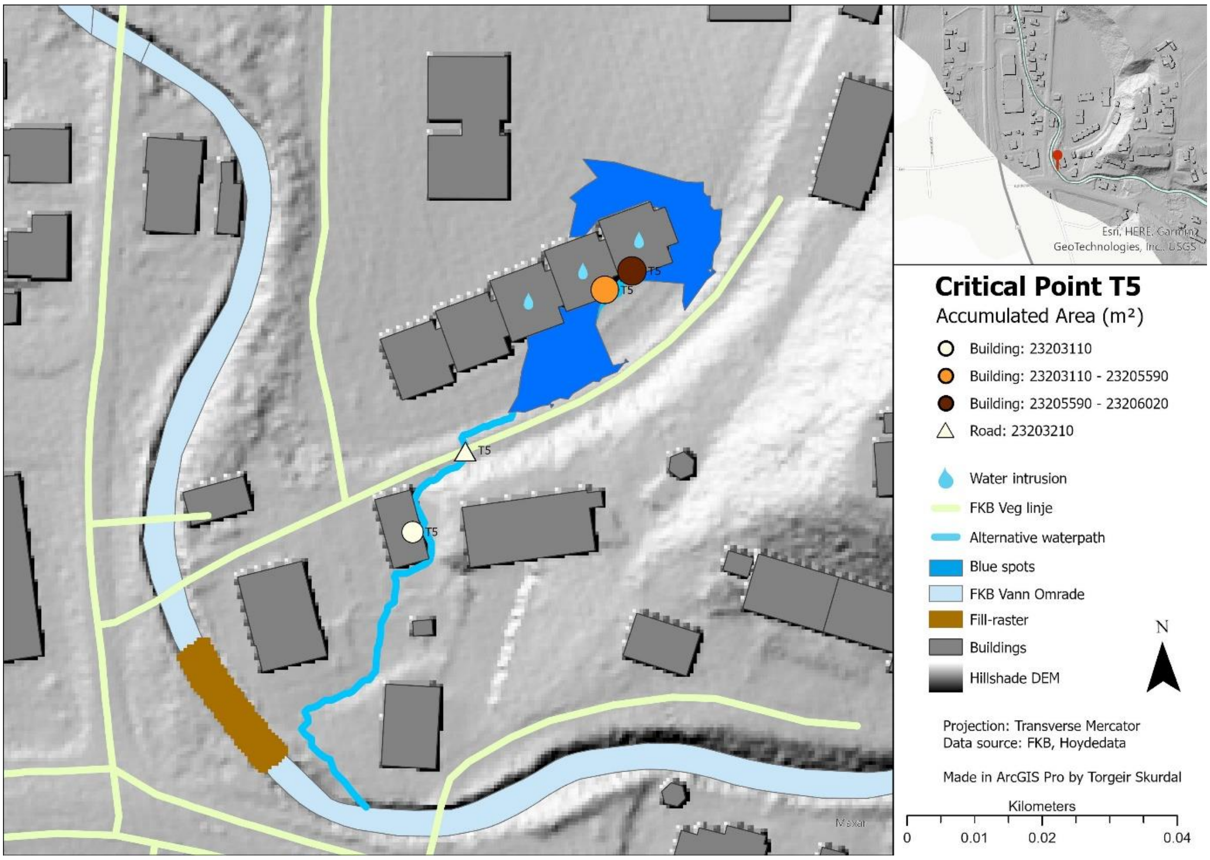


Figure 6.28: Showing the indicated damage caused to buildings and roads by critical point 5, from running the model.

ORIG_FID	objtype	bygningsnu	LayerName	Acc_Area	Damage_Class
1	Bygning	184537591	T5	23203110	5
2	Bygning	300790260	T5	23206020	5
3	Bygning	300790261	T5	23205590	5

Table 6.6: Showing identifying output attributes for critical point 5 building damage.

FID_T5	GATENAVN	LayerName	Acc_Area	Damage_Class
54	Nyhusvegen	T5	23203210	5

Table 6.7: Showing identifying output attributes for critical point 5, road damage.

If critical point 5 is blocked it has potential to cause damage to roads, this is highlighted in Figure 6.28, as water will flood over roads in the case that this point is blocked. The output attributes describing the damage class and the name of the road that may be damaged are illustrated in Tables 6.6. and 6.7 To prevent damage, flood prevention measures in this location should be considered.

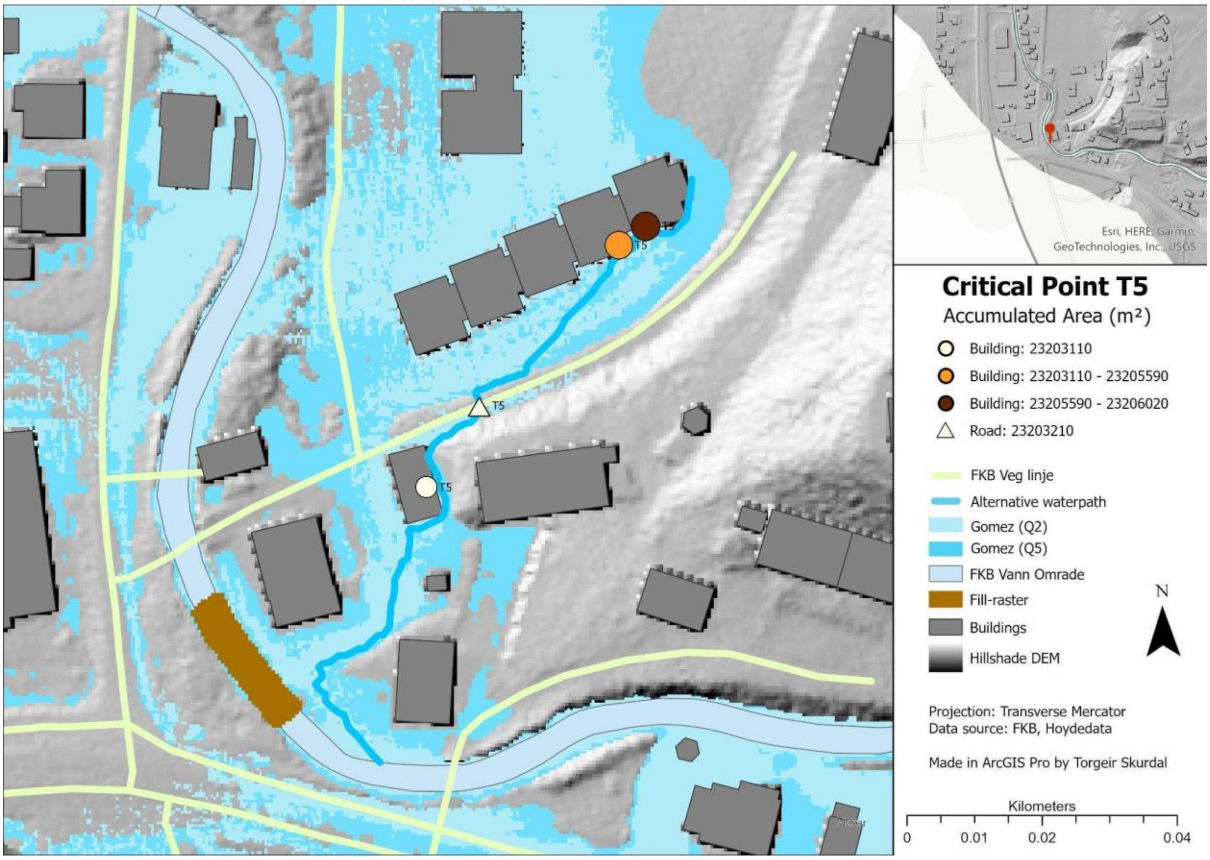


Figure 6.29: Showing the modeled damage caused to buildings and roads by critical point 5, compared to Flood map by Gomez (2021)

Comparing the model output with the flood maps by Gomez (Figure 6.29), the modeled water line matches up very well with the flow path of a 2-year flood, where it will flow out to the right causing damage to road and several buildings.

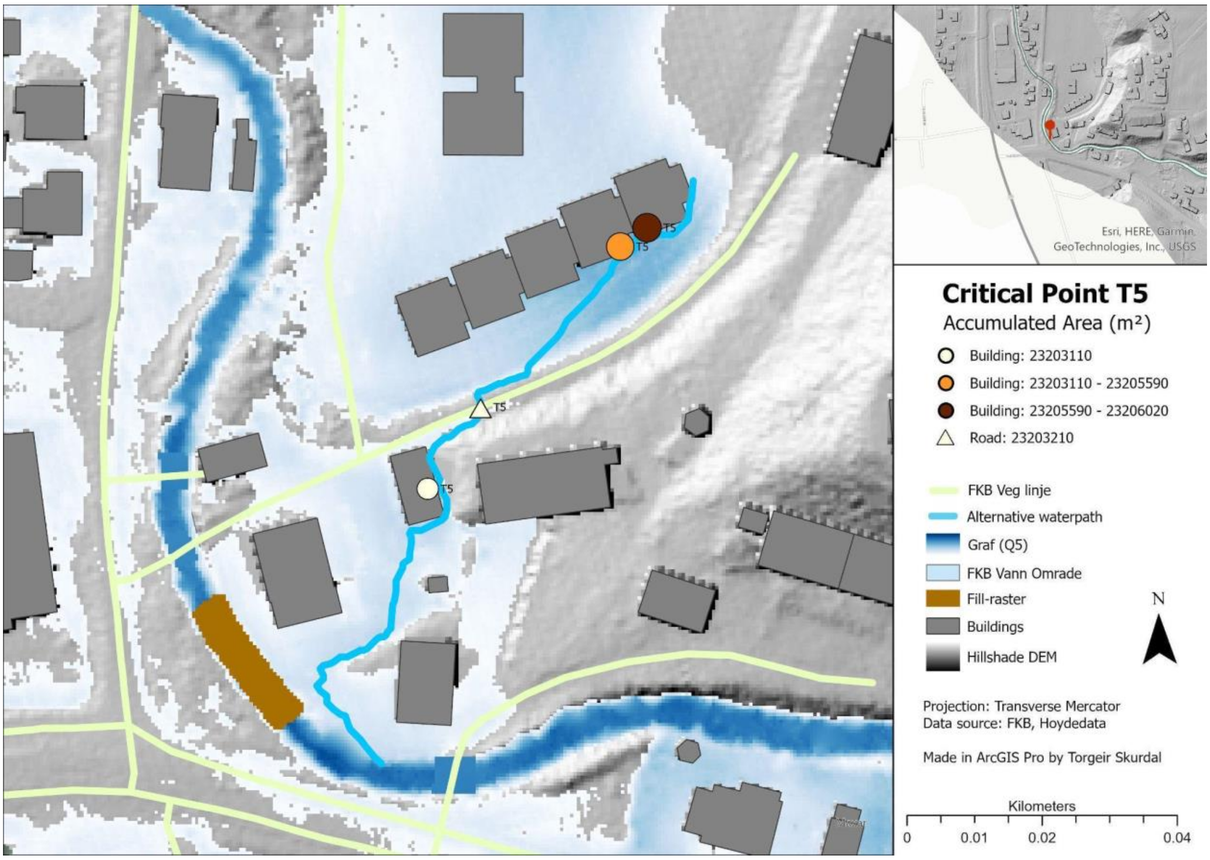


Figure 6.30: Showing the modeled damage caused to buildings and roads by critical point 5, compared to flood map by Graf (2021)

Comparing the model output with the flood maps by Graf (Figure 6.30), the modeled water line matches up with the map, but the extent is much greater and according to Graf this area will cause greater damage than the ones indicated by the model.

Critical point 6

Buildings

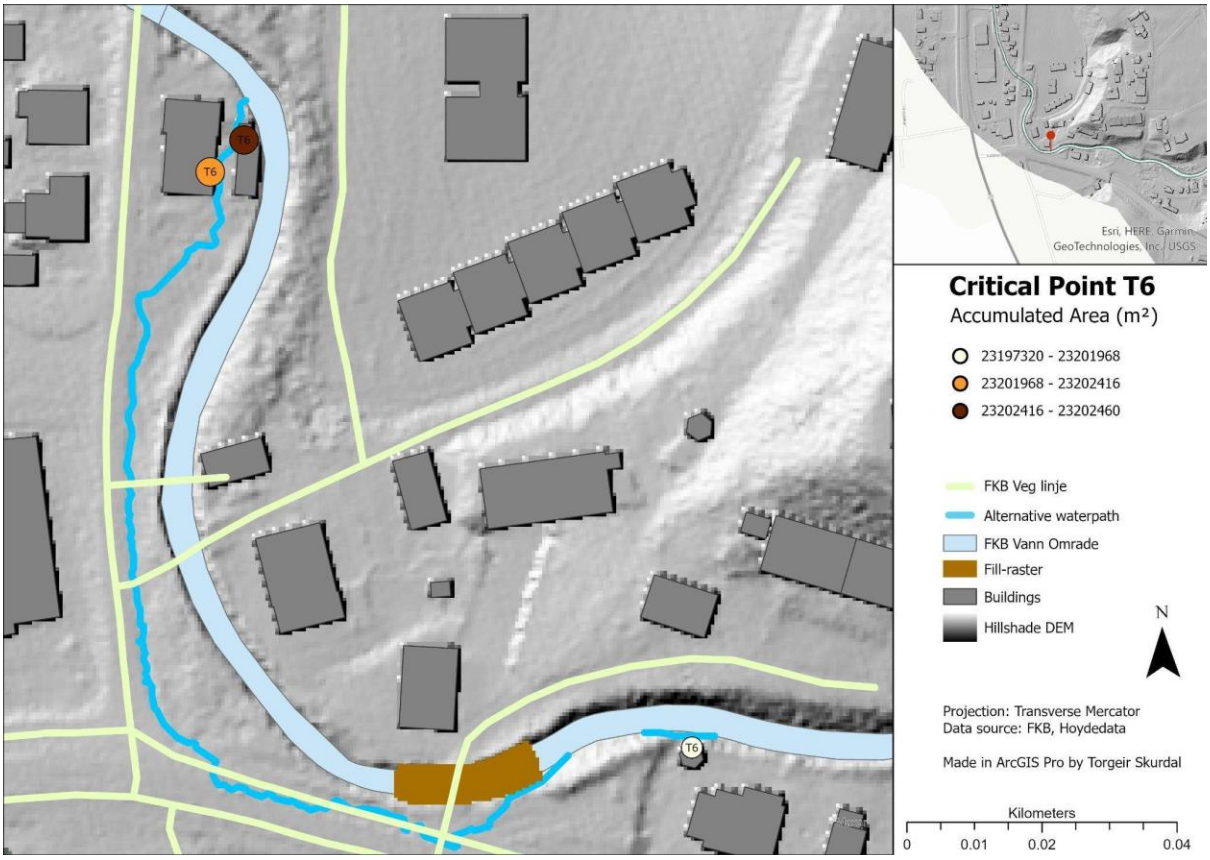


Figure 6.31: Showing the indicated damage caused to buildings by critical point 6, from running the model.

ORIG_FID	objtype	bygningsnu	LayerName	Acc_Area	Damage_Class
1	Bygning	21995444	T6	23202460	5
2	Bygning	10739004	T6	23202410	5
3	Takoverbygg	0	T6	23197320	5

Table 6.8: Showing identifying output attributes for critical point 6 building damage.

Roads

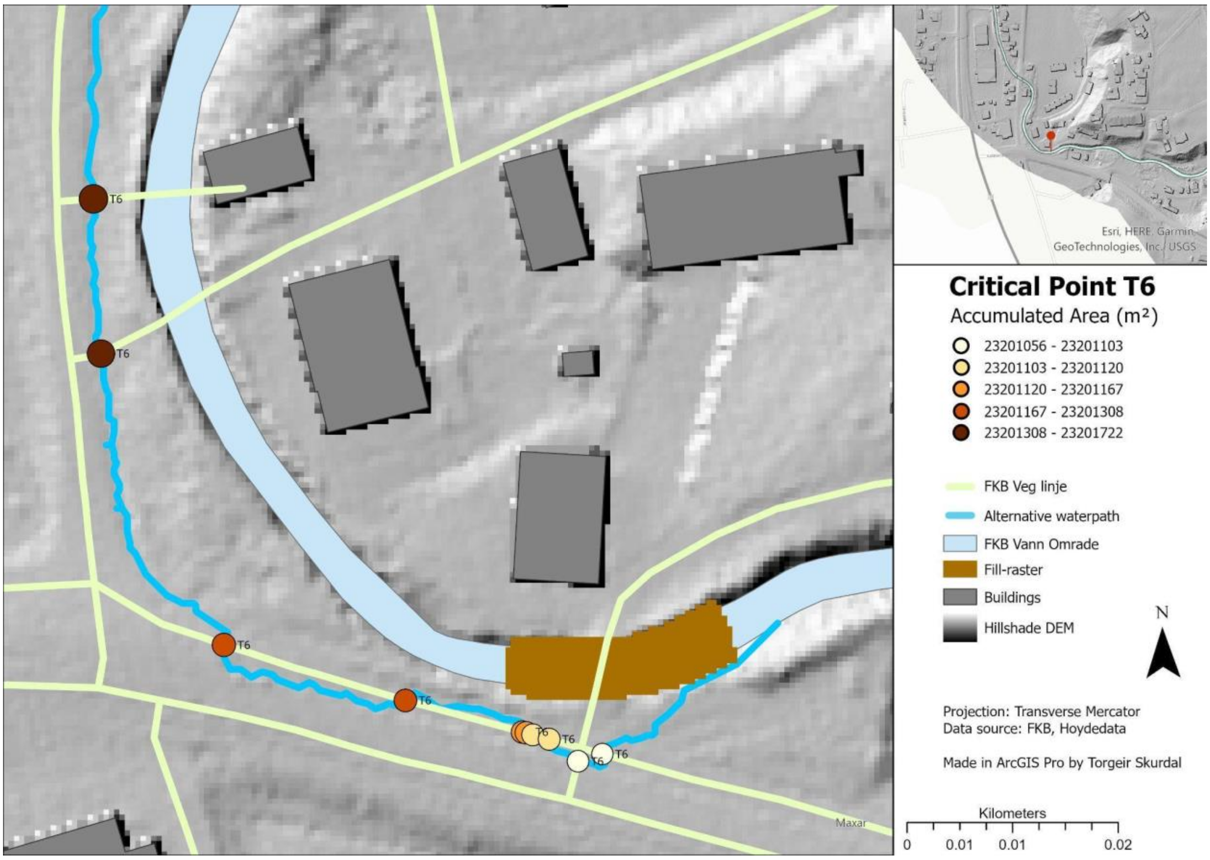


Figure 6.32: Showing the indicated damage caused to roads by critical point 6, from running the model.

FID_T6	GATENAVN	LayerName	Acc_Area	Damage_Class
100	Nyhusvegen	T6	23201620	5
222	Fremovegen	T6	23201080	5
221		T6	23201060	5
186		T6	23201170	5
206		T6	23201130	5
207		T6	23201130	5
208		T6	23201120	5
217		T6	23201110	5
76		T6	23201720	5
159		T6	23201250	5

Table 6.9: Showing identifying output attributes for critical point 6, road damage.

The output from the model indicates that if critical point 6 is filled it has potential to cause great damage to roads and buildings, this is highlighted in Figures 6.31 and 6.32. The output attributes describing the damage class and the building numbers of the damaged buildings are illustrated in Tables 6.8 and 6.9. The critical point 6 attribute table for road damage (Table 6.9) is lacking some street names (GATENAVN). These are sidewalks and therefore unnamed. Looking at the model output there are no bluespots in this area. According to the model output, to prevent damage, flood prevention measures in this location should be considered.

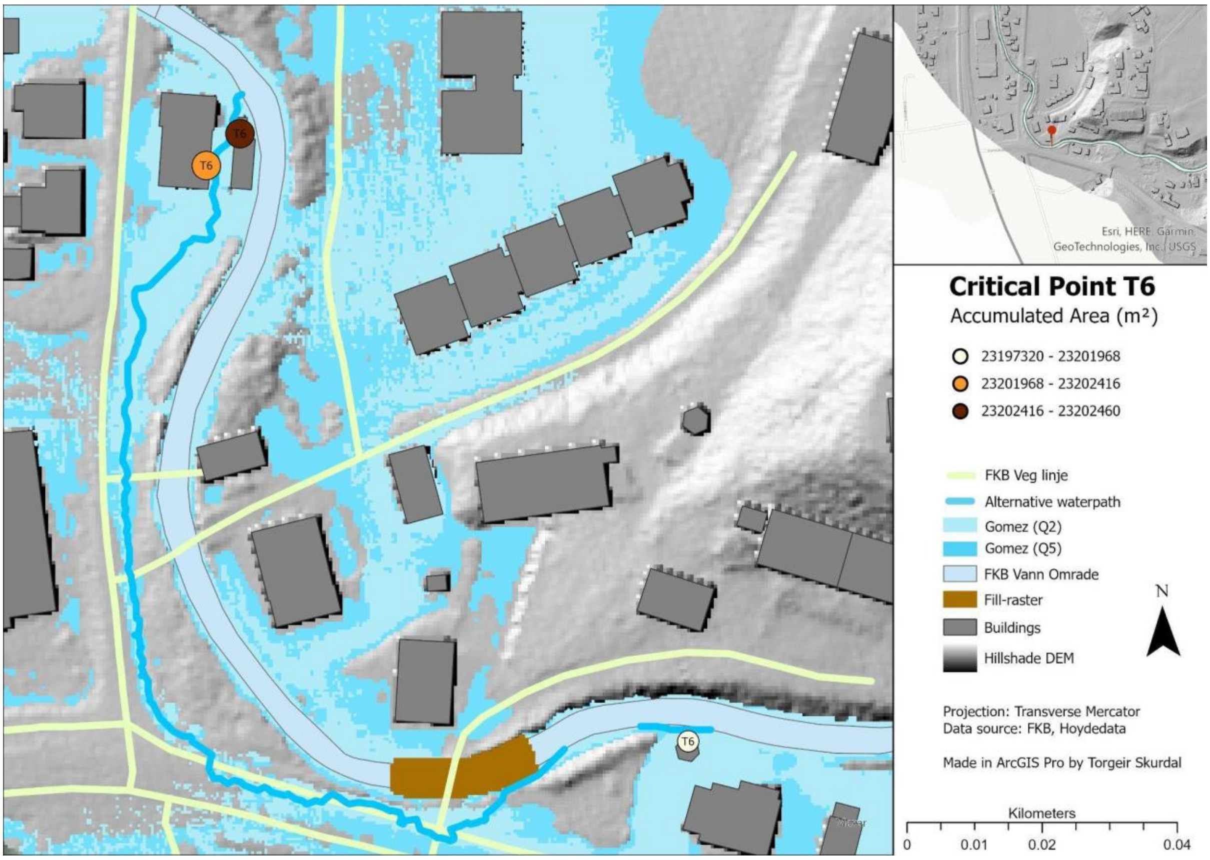


Figure 6.33: Showing the modeled damage caused to buildings by critical point 6, compared to Flood map by Gomez (2021)

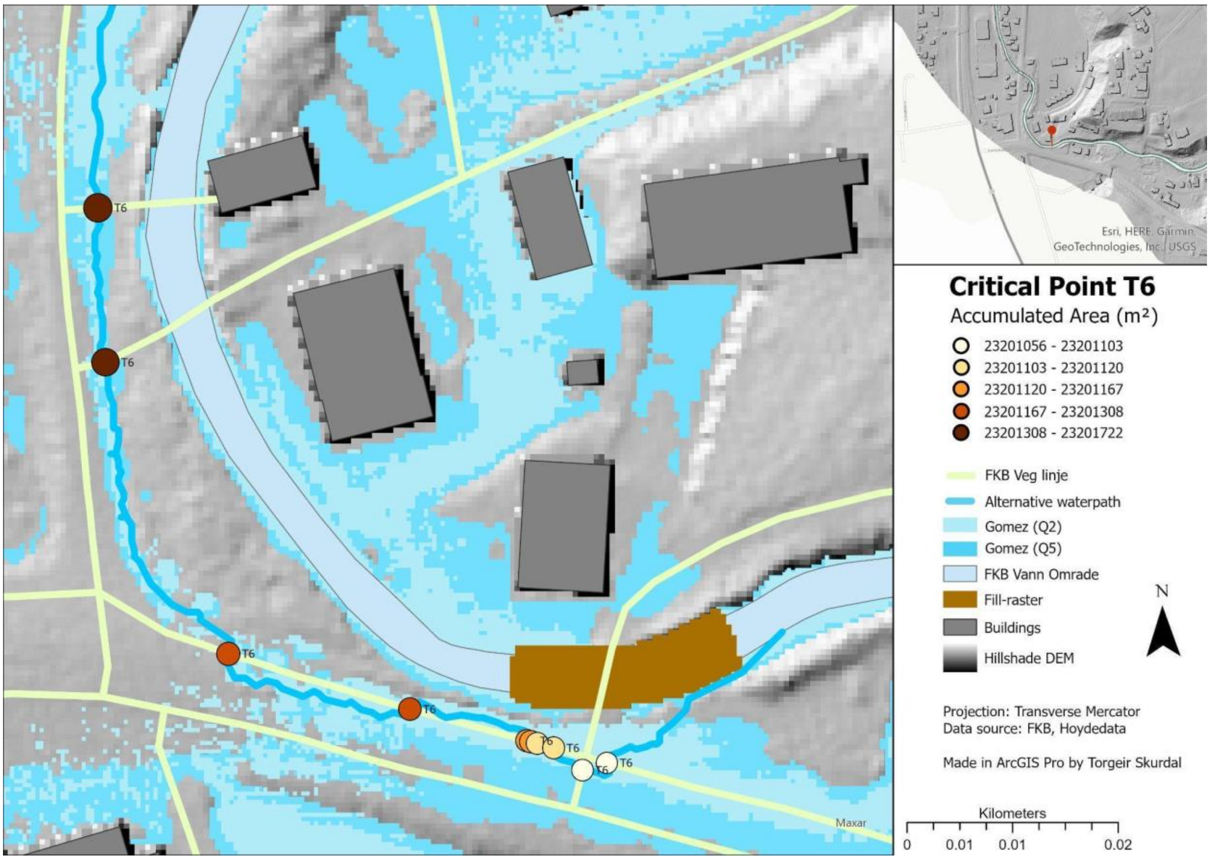


Figure 6.34: Showing the modeled damage caused to roads by critical point 6, compared to Flood map by Gomez (2021)

Comparing the model output to the flood maps by Gomez (Figure 6.33 and 6.34) the model output matches up with the flow path of a 2-year flood, where the water will flow the walking path where it meets the road and turn right to the houses. Both the Q2 and Q5 flood flows about the same path here.

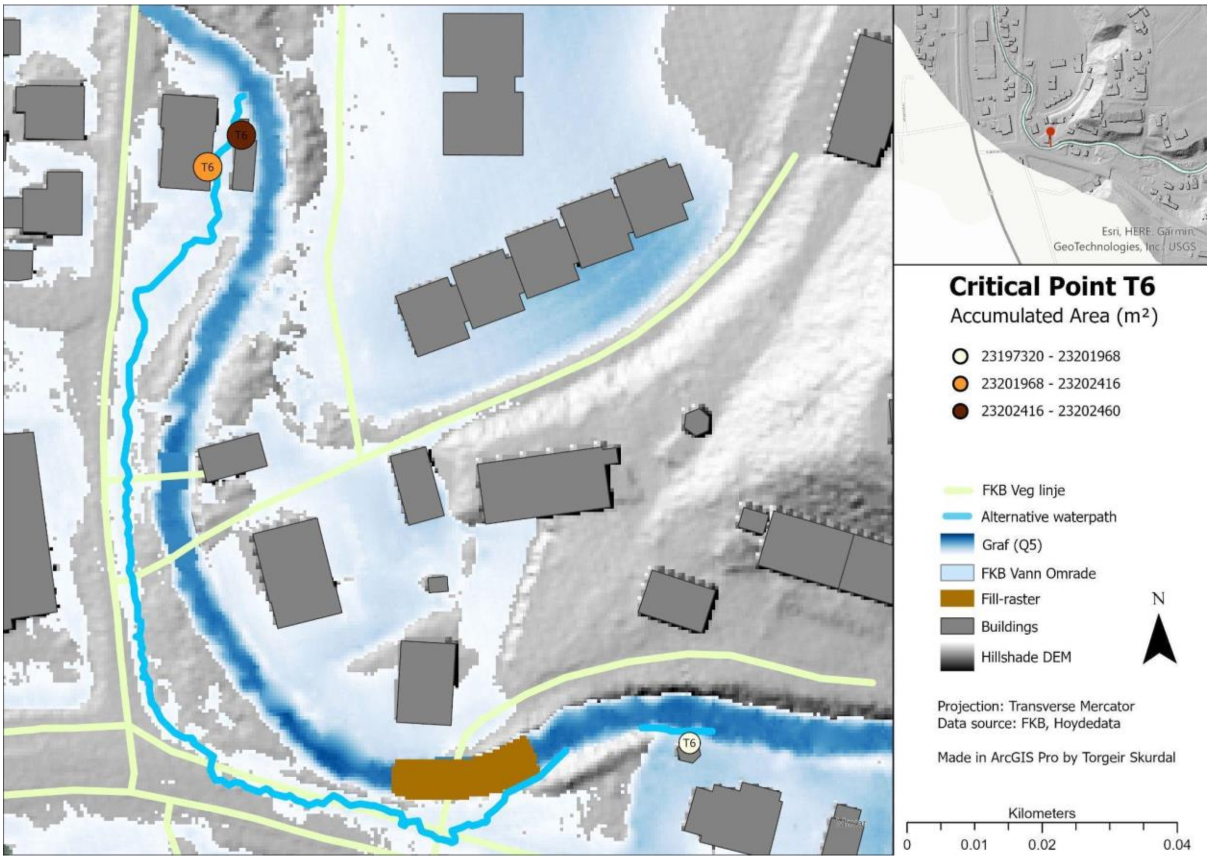


Figure 6.35: Showing the modeled damage caused to buildings by critical point 6, compared to Flood map by Graf (2021)

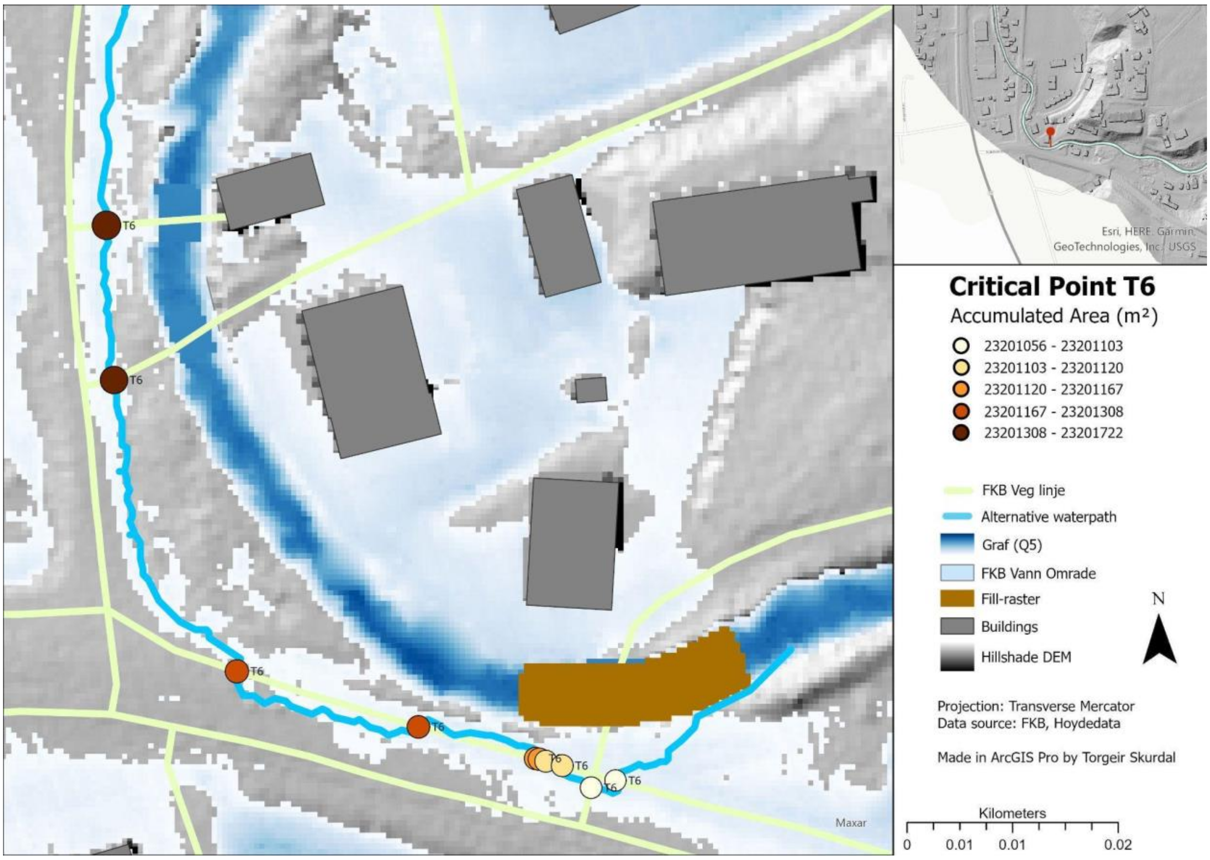


Figure 6.36: Showing the modeled damage caused to roads by critical point 6, compared to Flood map by Graf (2021)

Comparing the output with the flood maps by Graf (Figures 6.35 and 6.36) the output of the model is further validated, and the maps by Graf and Gomez shows very similar results if critical point 6 is filled or the channel overflows.

Critical point 7

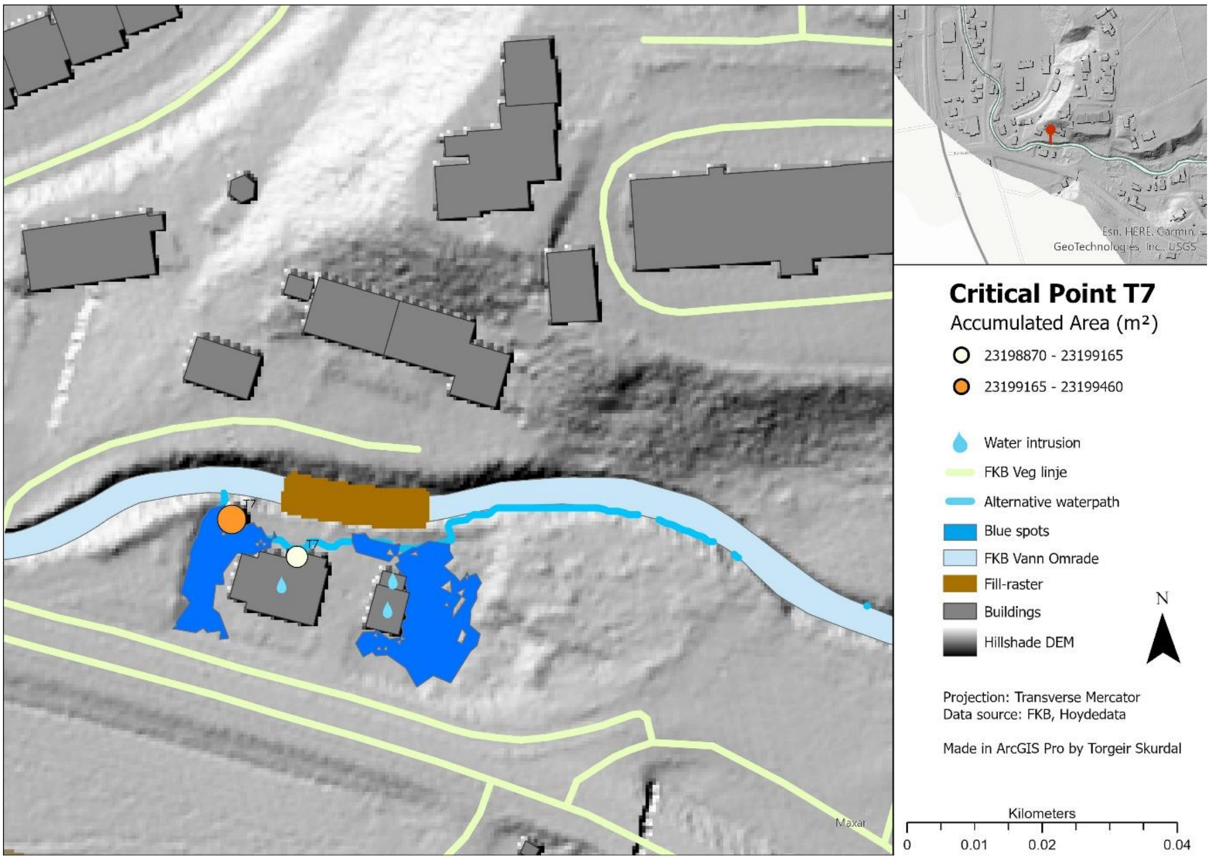


Figure 6.37: Showing the indicated damage caused to buildings by critical point 7, from running the model.

ORIG_FID	objtype	bygningsnu	LayerName	Acc_Area	Damage_Class
1	Bygning	184538016	T7	23198870	5
2	Takoverbygg	0	T7	23199460	5

Table 6.10: Showing identifying output attributes for critical point 7, building damage.

The output from the model indicates that if critical point 7 is filled it has potential to cause great damage to several private property, this is highlighted in Figure 6.37. The output attributes describing the damage class and the building numbers of the damage buildings are illustrated in Table 6.10. Looking at the bluespot It is also likely that water will accumulate in this area, causing water intrusion. According to the model output, to prevent damage, flood prevention measures in this location should be considered.

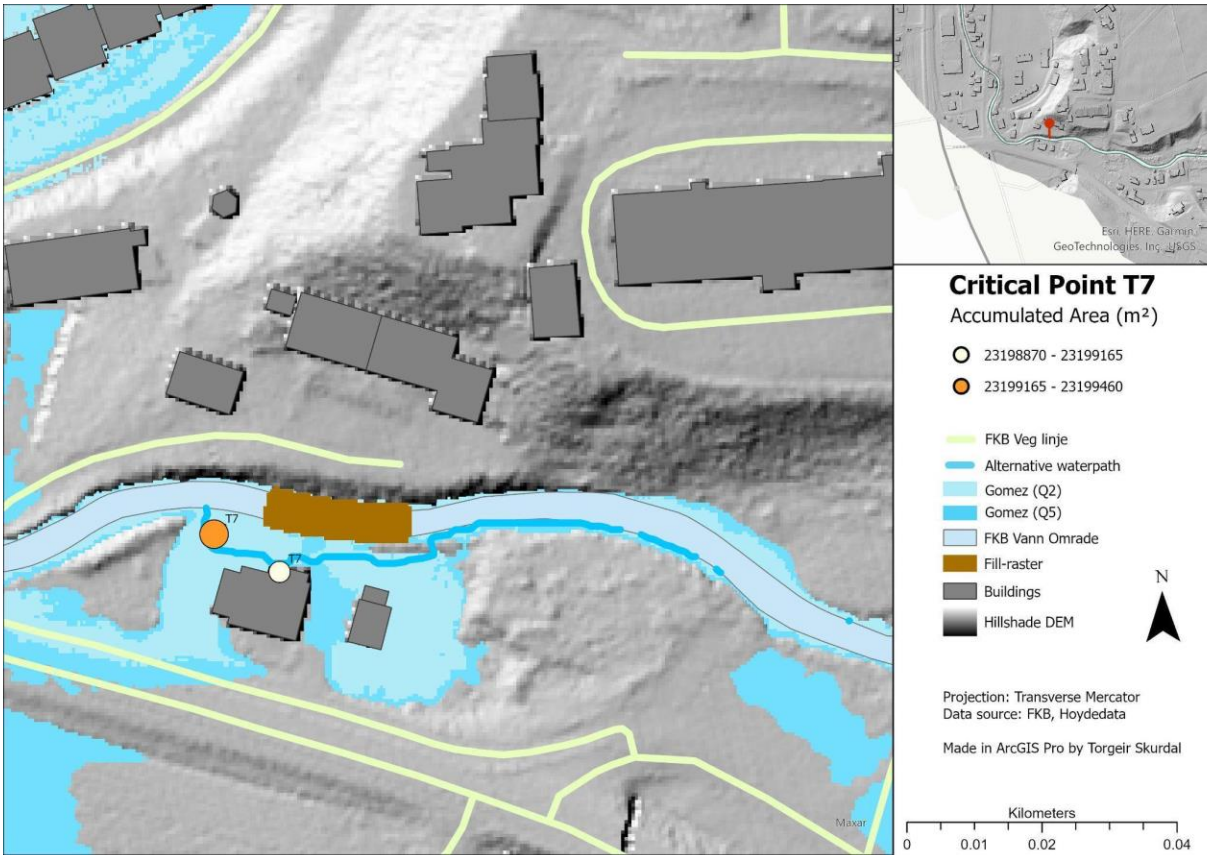


Figure 6.38: Showing the modeled damage caused to buildings by critical point 7, compared to Flood map by Gomez (2021)

Comparing the model output with the flood maps by Gomez (Figure 6.38) the flow path matches up with the 2-year flood map, considering both the drainage line and the bluespots, they both match up well with a 2-year flood. In the flood maps by Gomez the water will take alternative paths in this area, causing damage to roads, and likely buildings.

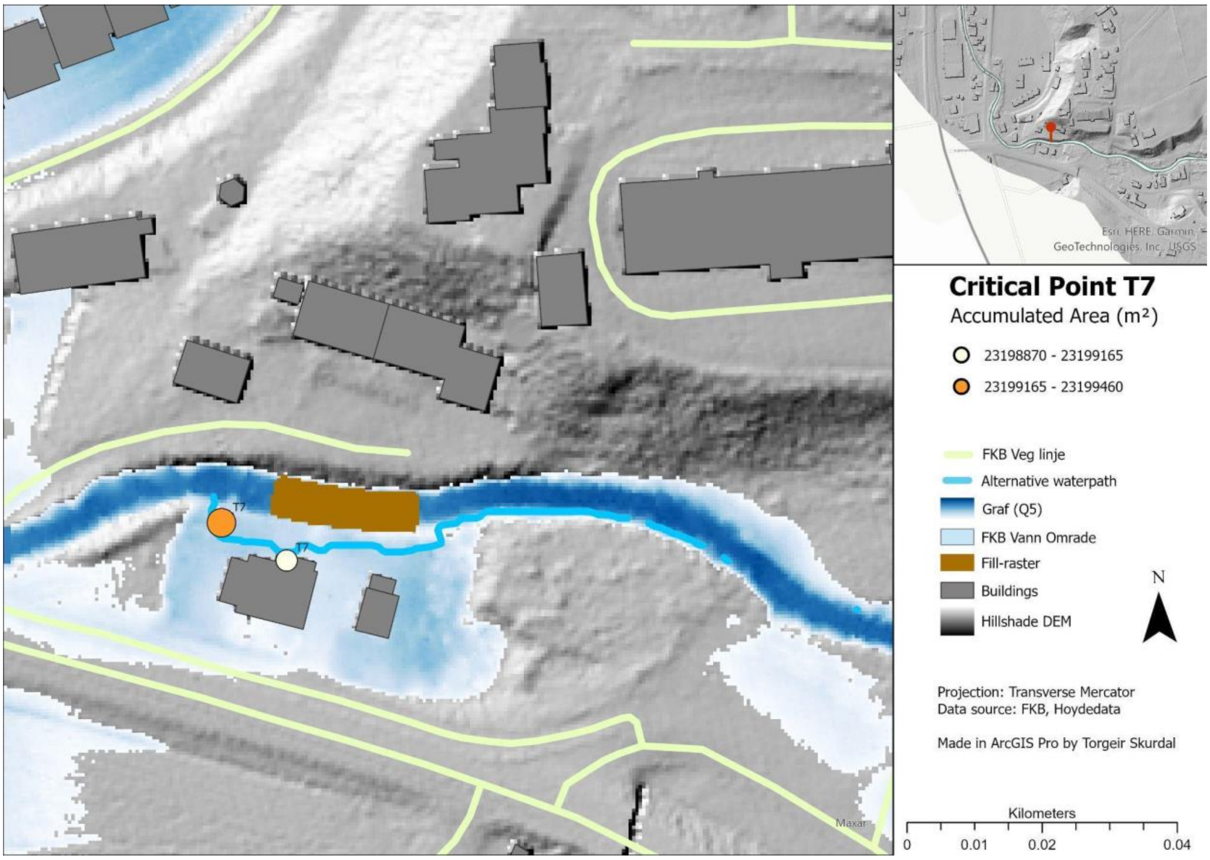


Figure 6.39: Showing the modeled damage caused to buildings by critical point 7, compared to Flood map by Graf (2021)

Comparing the model output with the flood maps by Graf the flow path matches up with the flood map, considering both the drainage line and the bluespots. In the flood maps by Graf the water will take alternative paths in this area, causing damage to roads, and likely buildings. Extent of the flooded area are also very similar in flood maps compared to Gomez.

Critical point 8

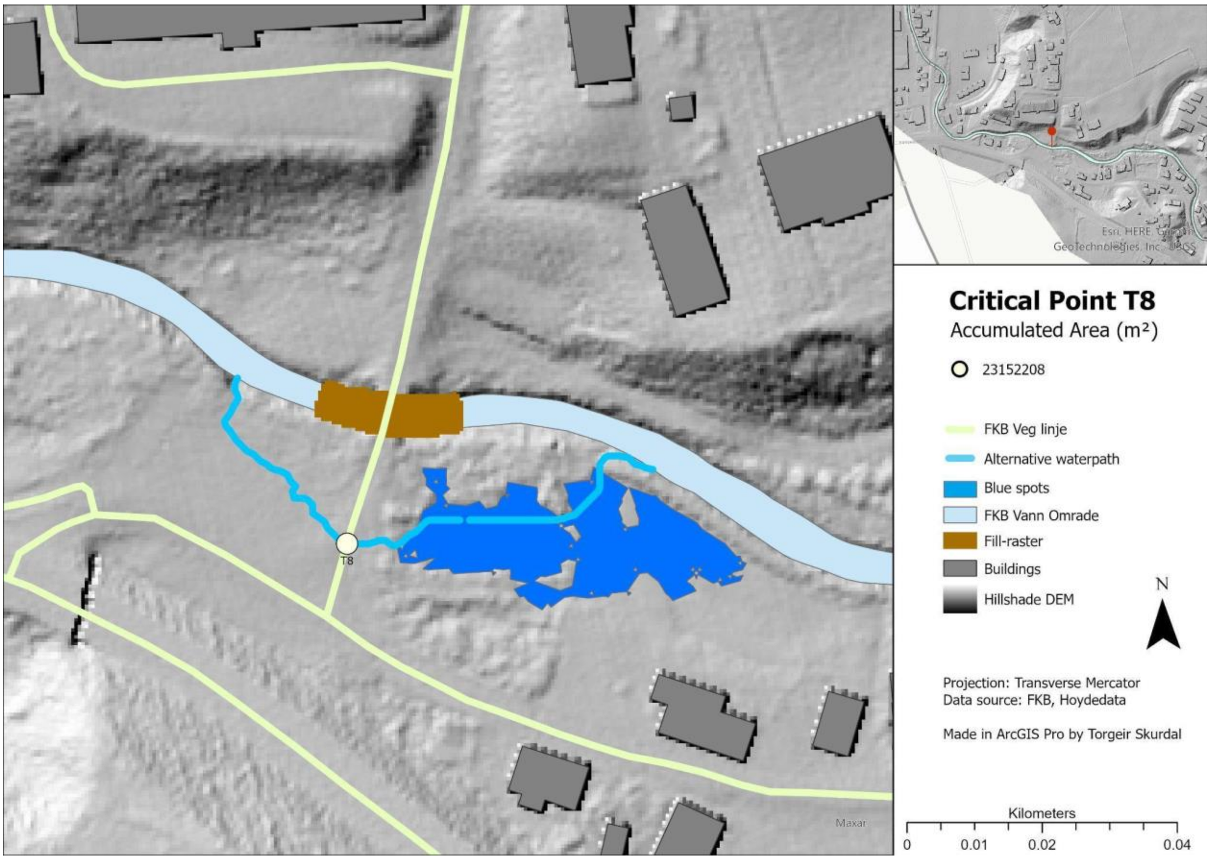


Figure 6.40: Showing the indicated damage caused to roads by critical point 8, from running the model.

FID_T8	GATENAVN	LayerName	Acc_Area	Damage_Class
74	Sagbakken	T8	23152210	5

Table 6.11: Showing identifying output attributes for critical point 8 road damage.

If critical point 8 is blocked it has potential to cause damage to roads, this is highlighted in figure 6.16. The output attributes describing the damage class and the name of the road that may be damaged are illustrated in Table 6.3. Looking at the bluespot, parts of someone's garden will be flooded. To prevent damage, flood prevention measures in this location should be considered.

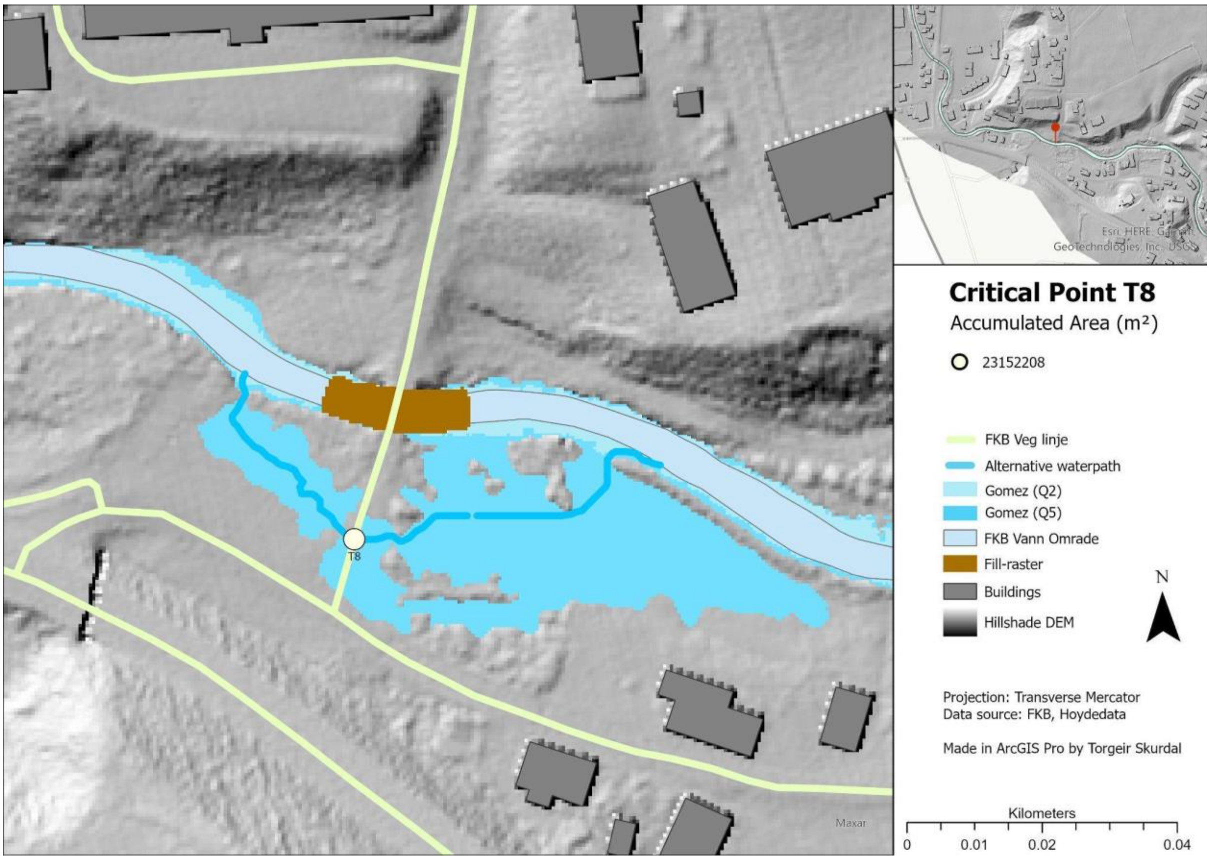


Figure 6.41 Showing the modeled damage caused to roads by critical point 8, compared to Flood map by Gomez (2021)

Comparing the model output to the flood maps by Gomez (Figure 6.41) the model output matches up with the flow path of a 5-year flood, where the water will flow and accumulate in the garden of the house, and on a parking lot to the lower left of the critical point. The area will overflow but the river will not change path.

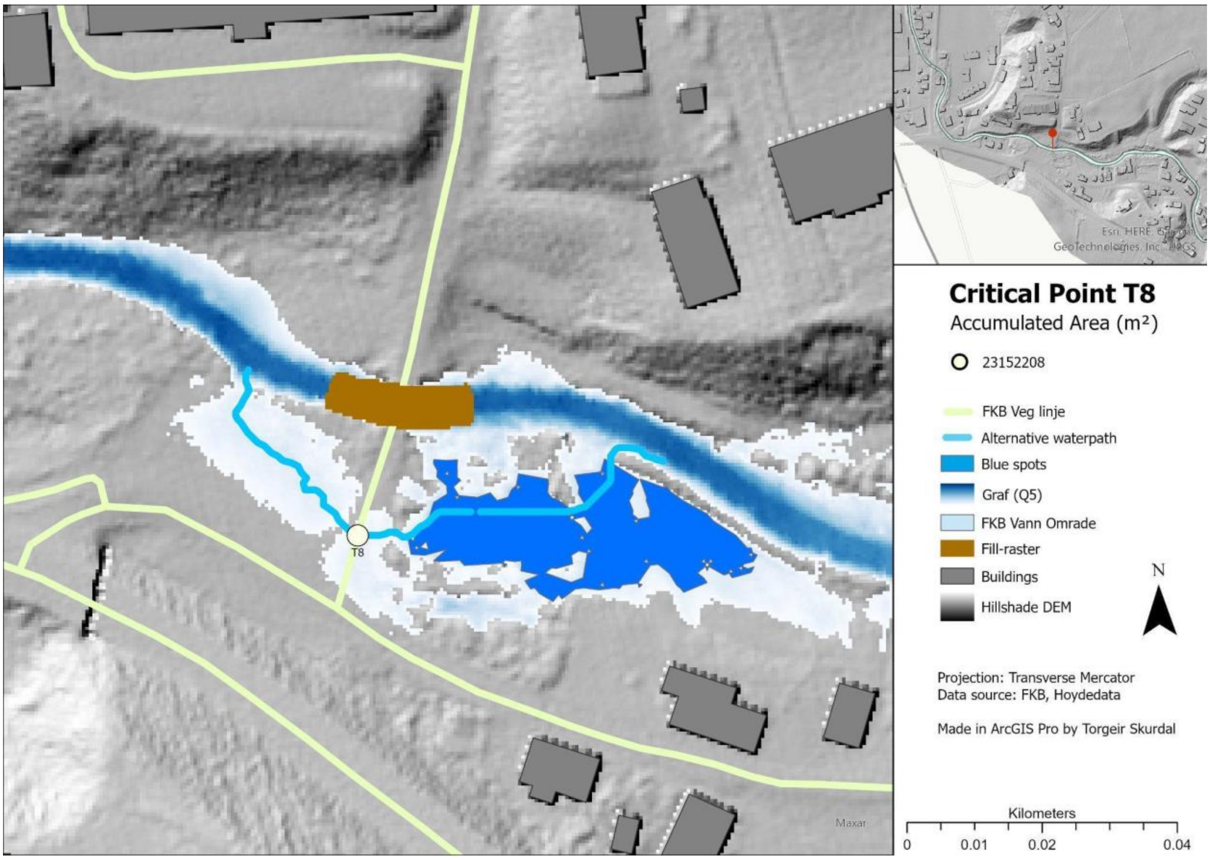


Figure 6.42: Showing the modeled damage caused to roads by critical point 8, compared to Flood map by Graf (2021).

Comparing the model output to the flood maps by Graf (Figure 6.42) the model output matches up map, and therefore matching up to a 5-year flood, where the water will flow and accumulate in the garden of the house, and on a parking lot to the lower left of the critical point. Compared to Gomez the flooded area is slightly larger. The area will overflow but the river will not change path

Critical point 9

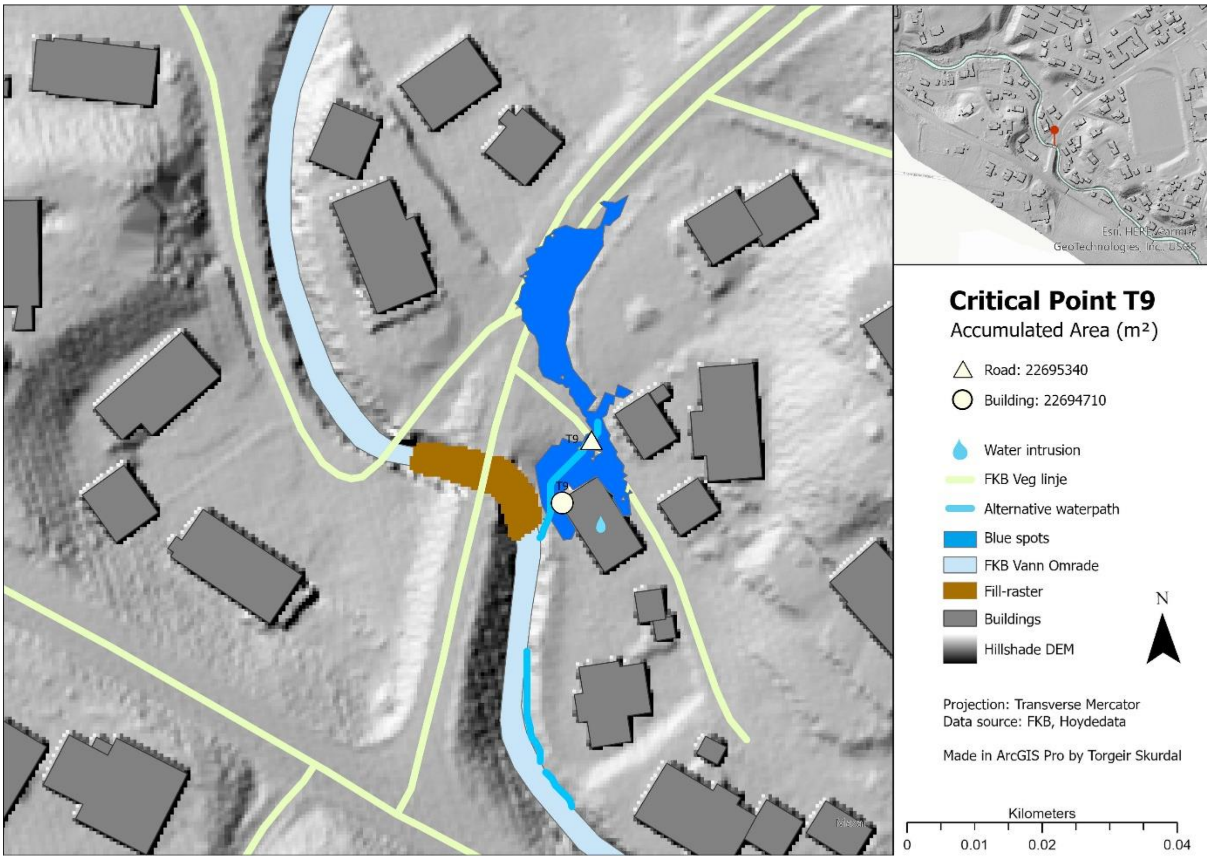


Figure 6.43: Showing the indicated damage caused to buildings by critical point 9, from running the model.

FID_disT9	objtype	bygningsnu	LayerName	Acc_Area	Damage_Class
1	Bygning	184539519	T9	22694710	5

Table 6.12: Showing identifying output attributes for critical point 9, building damage.

FID_T9	GATENAVN	LayerName	Acc_Area	Damage_Class
3	Lykkjvegen	T9	22695340	5

Table 6.13: Showing identifying output attributes for critical point 9, road damage.

According to the model output, if critical point 9 is blocked it has potential to cause damage to a building and roads, this is highlighted in Figure 6.43. The output attributes describing the damage class and the name of the road that may be damaged are illustrated in Tables 6.12 and 6.13. Looking at the bluespot, a house and the roads flooded. Both the building and parts of the roads reside within a bluespot that will get filled with water if this critical point is filled. To prevent damage, flood prevention measures in this location should be considered.

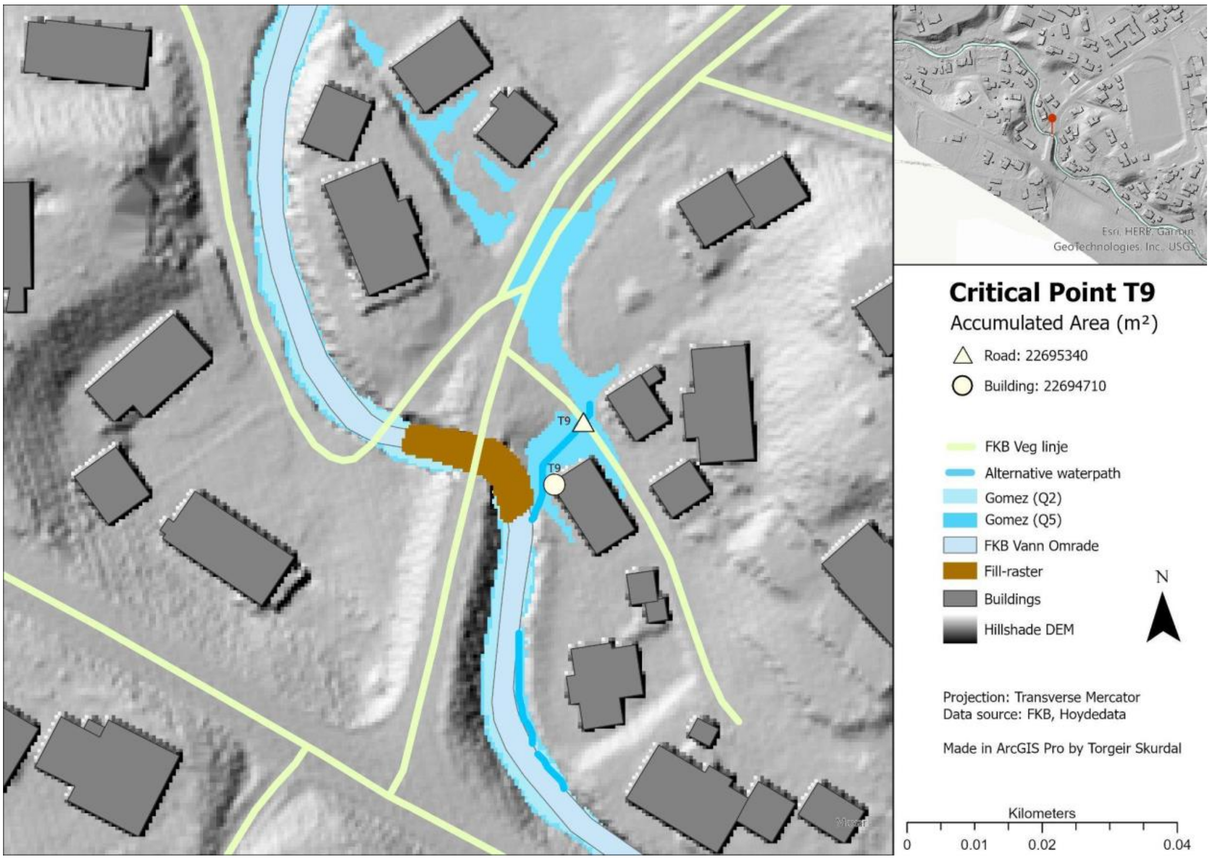


Figure 6.44: Showing the modeled damage caused to buildings and roads by critical point 9, compared to flood map by Gomez (2021).

Comparing the model output with the flood maps by Gomez (Figure 6.44) the flow path matches up with the 5-year flood map by Gomez, considering both the drainage line and the bluespot. River has potential to find alternative path.

Figure 6.45: Showing the modeled damage caused to buildings and roads by critical point 9, compared to Flood map by Graf (2021).

Comparing the model output with the flood map by Graf (Figure 6.45) the flood path corresponds to a 5-year flood. Considering both the drainage line and the bluespot, the bluespot has a slightly lesser extent compared to the flooded area in the map by Graf. River has potential to find alternative path.

Critical point 10

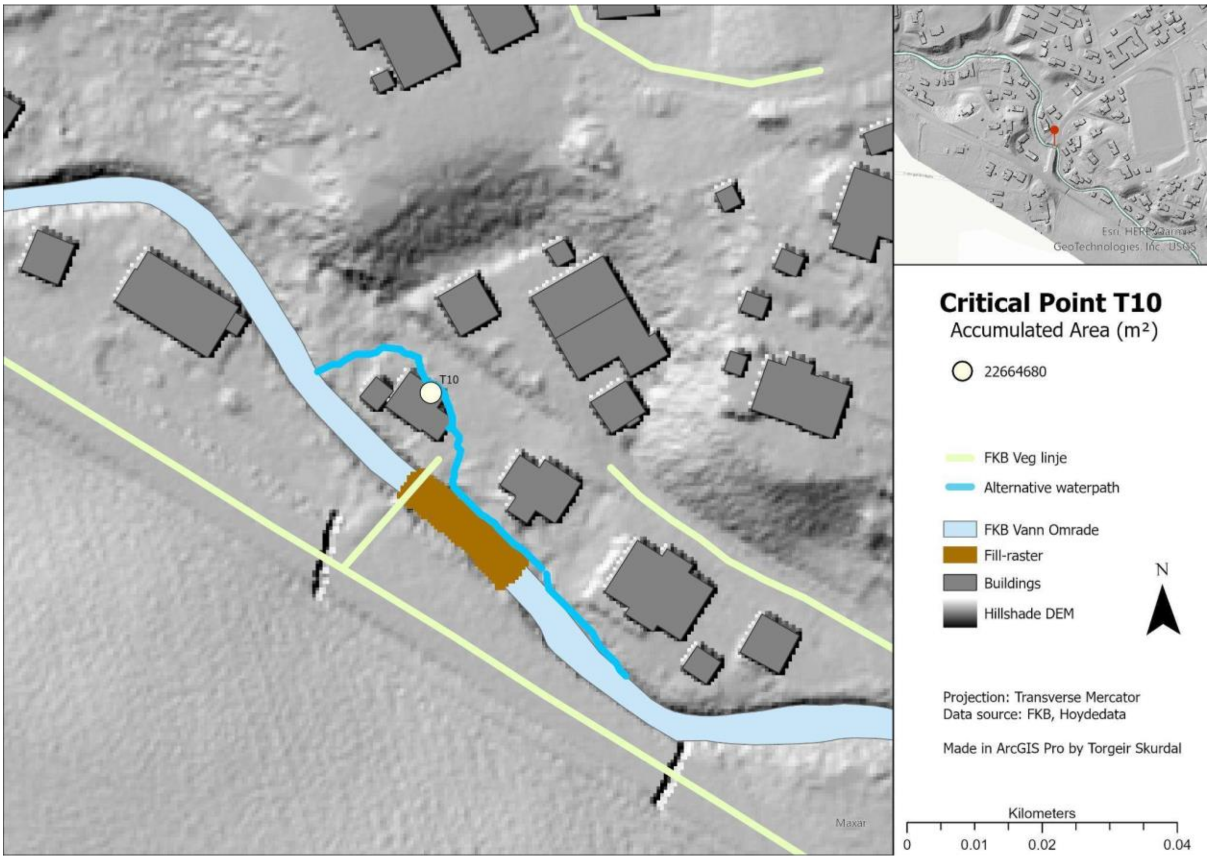


Figure 6.46: Showing the indicated damage caused to buildings by critical point 10, from running the model.

FID_disT10	objtype	bygningsnu	LayerName	Acc_Area	Damage_Class
1	Bygning	184539764	T10	22664680	5

Table 6.14: Showing identifying output attributes for critical point 10, building damage.

If critical point 10 is blocked it has potential to cause damage to a building and roads, this is highlighted in Figure 6.46. The output attributes describing the damage class and the name of the road that may be damaged are illustrated in Table 6.14. There are no bluespots in this area. To prevent damage, flood prevention measures in this location should be considered.

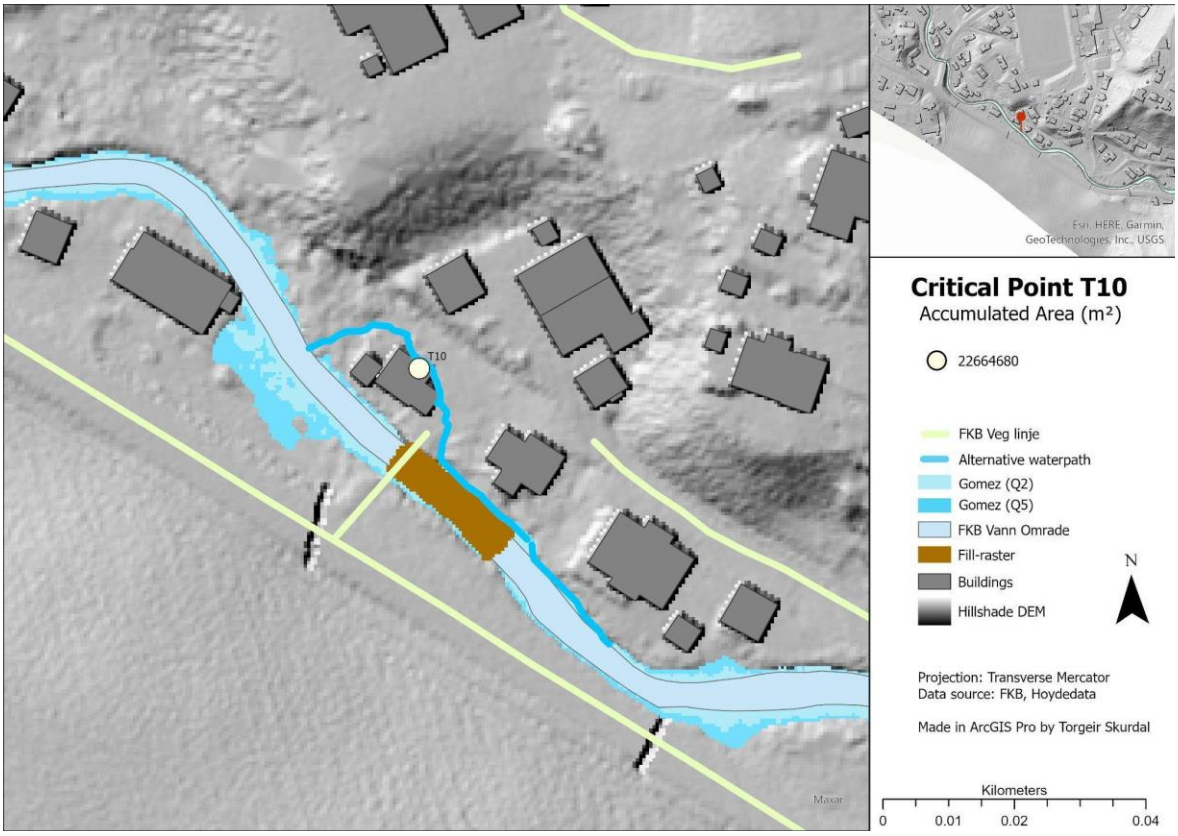


Figure 6.47: Showing the modeled damage caused to buildings by critical point 10, compared to Flood map by Gomez (2021).

Comparing the flood damage indicated by the model, to the flood maps by Gomez (Figure 6.47) this area will not flood in the case of a 5-year flood. On the other hand, the area to the left of the critical point will be flooded in the case of both a 2 and 5-year flood. This might point towards the critical point being placed wrong in this location.

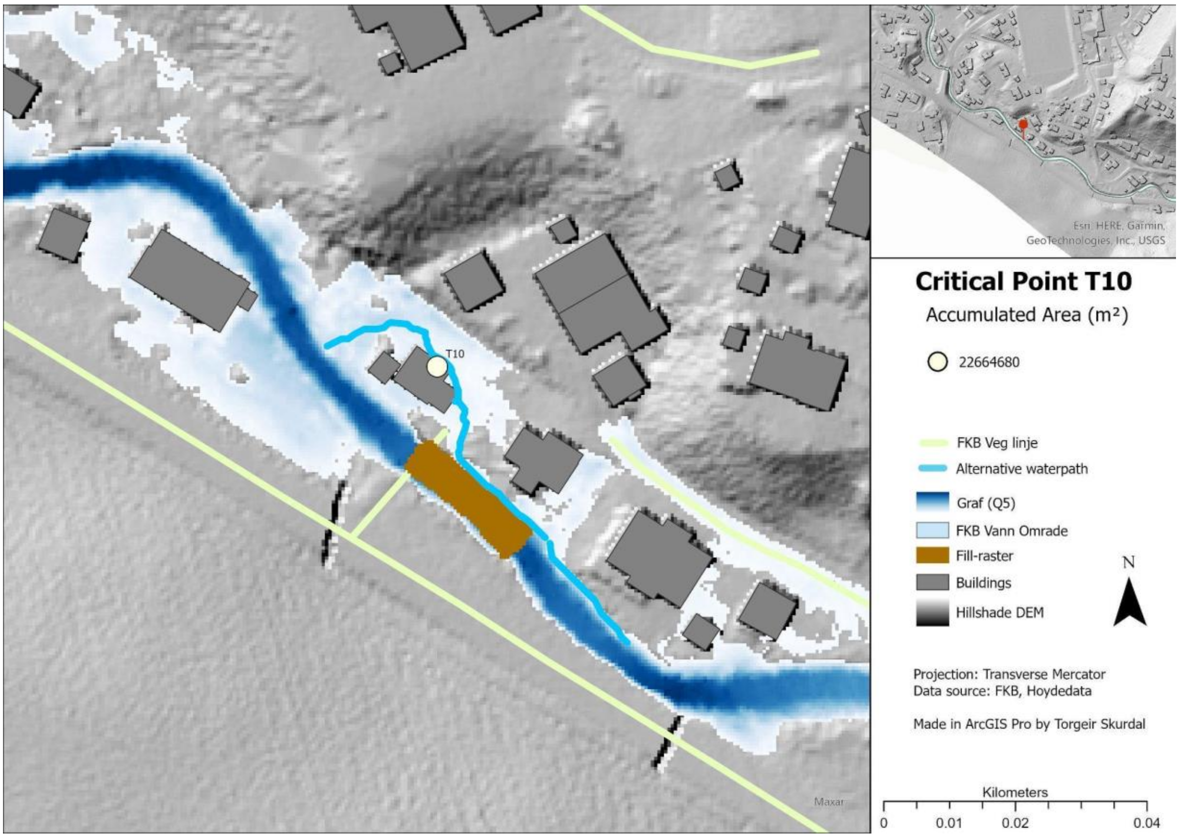


Figure 6.48: Showing the modeled damage caused to buildings by critical point 10, compared to Flood map by Graf (2021).

Comparing the flood damage indicated by the model, to the flood maps by Graf (Figure 6.48), this area will flood, and the water will accumulate, damaging several buildings in the case of a 5-year flood. The model output matches to some extent up with damage of a 5-year flood compared to map the map by Graf.

Critical point 11

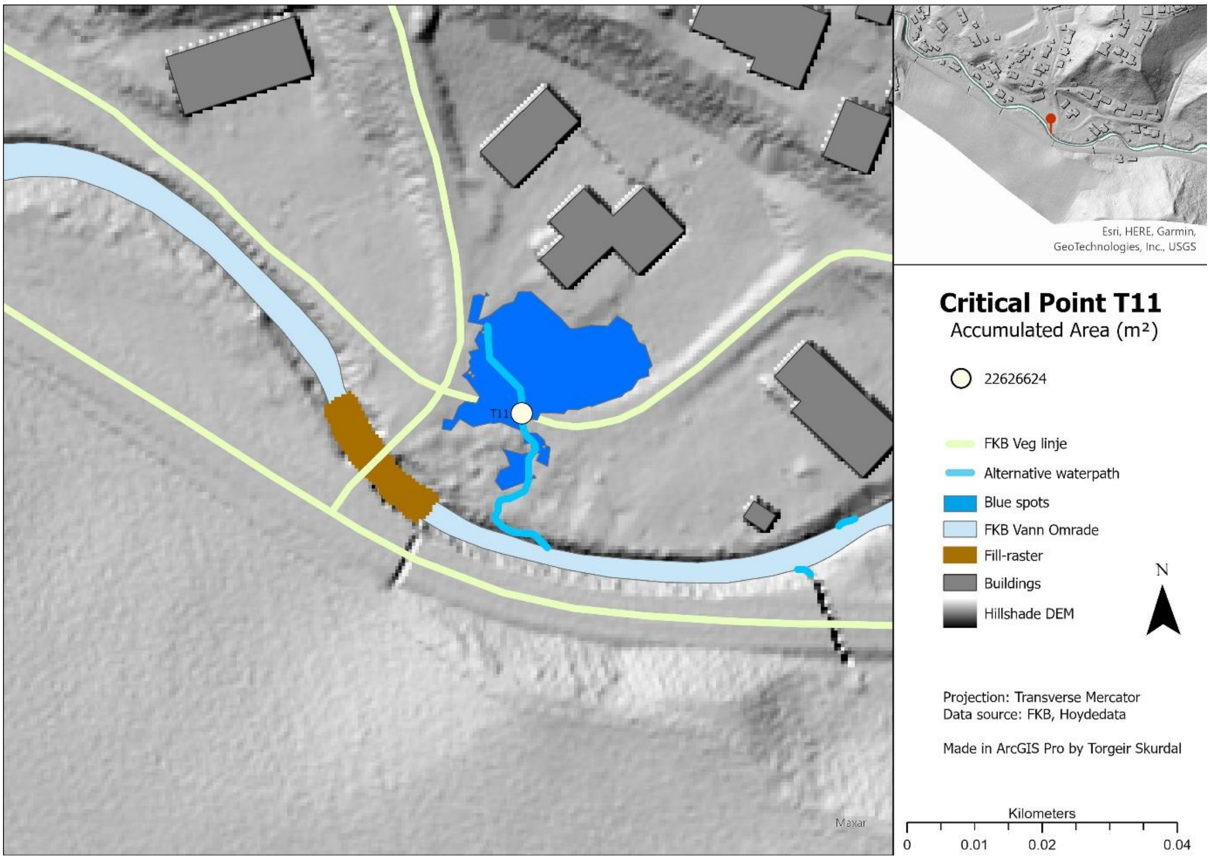


Figure 6.49: Showing the indicated damage caused to roads by critical point 11, from running the model.

FID_T11	GATENAVN	LayerName	Acc_Area	Damage_Class
15	Skolebakken	T11	22626620	5

Table 6.15: Showing identifying output attributes for critical point 11, road damage.

If critical point 11 is blocked it has potential to cause damage roads, this is highlighted in Figure 6.49. The output attributes describing the damage class and the name of the road that may be damaged are illustrated in Table 6.15. The bluespot in the area will flood a road. To prevent damage, flood prevention measures in this location should be considered.

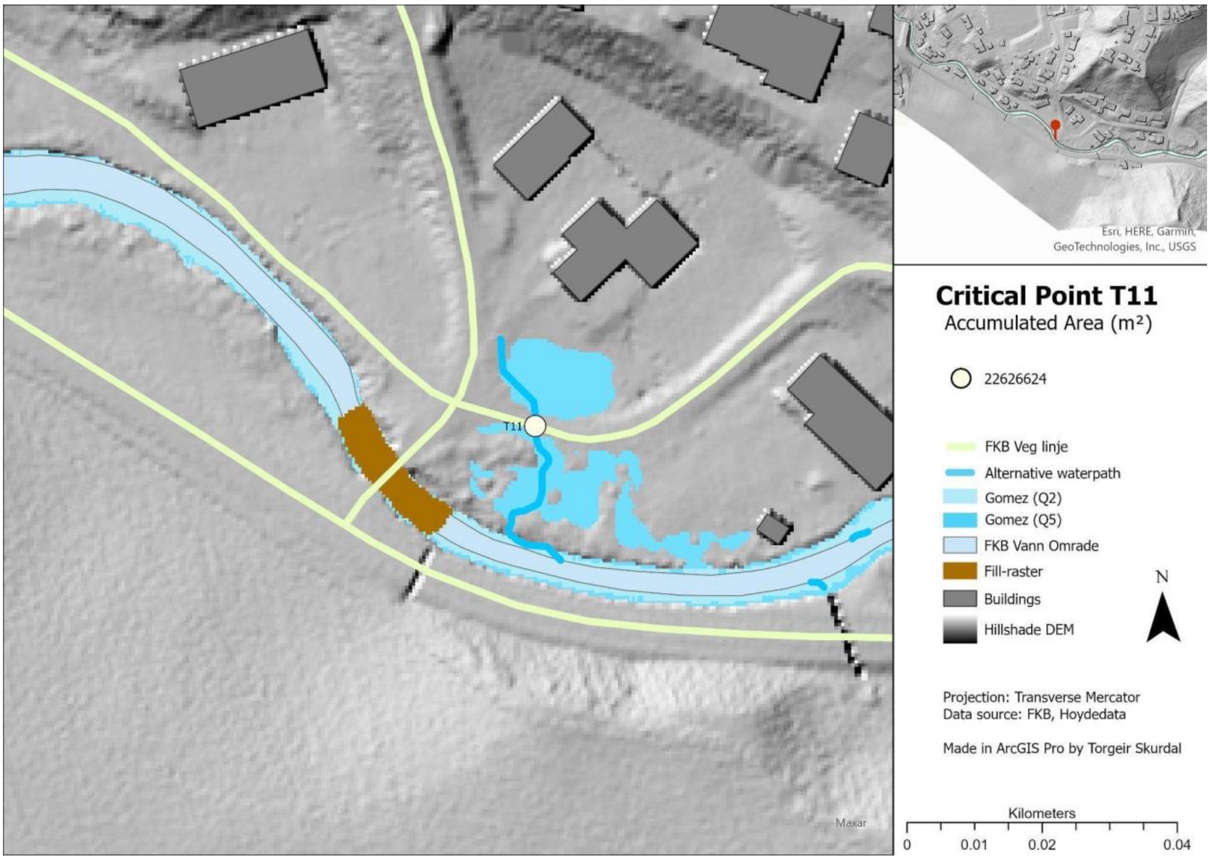


Figure 6.50: Showing the modeled damage caused to roads by critical point 11, compared to Flood map by Gomez (2021).

Comparing the model output to the flood maps by Gomez (Figure 6.50) the bluespot matched up well with the extent of the Q5 flood map by Gomez, meaning that the damage corresponds to ~5-year flood. Looking at the map moving the critical point or, adding an additional point might produce similar results as the flood Q5 flood map by Gomez.

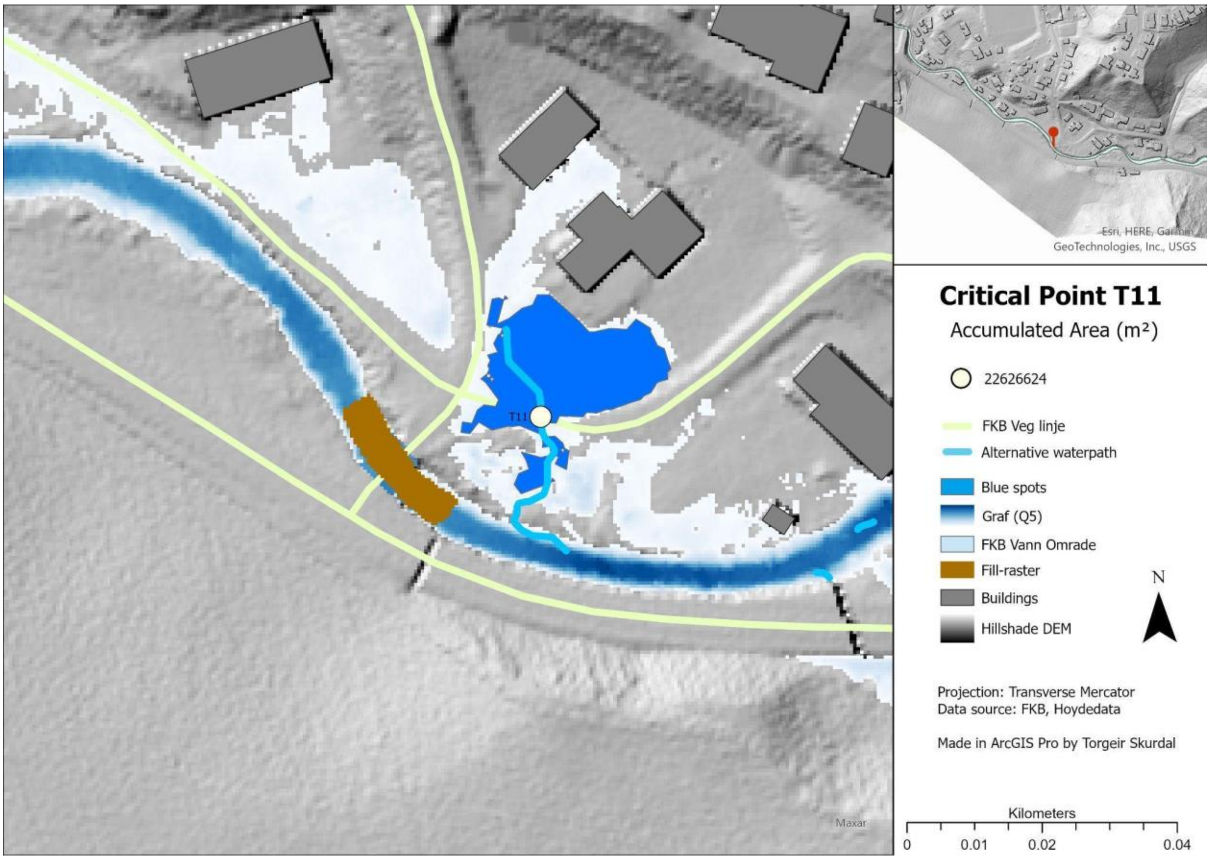


Figure 6.51: Showing the modeled damage caused to roads by critical point 11, compared to Flood map by Graf (2021).

Comparing the model output to the flood map by Graf (Figure 6.51), further validates the output of the model, even though the flooded area is larger compared to the modelled bluespot, the river overflows its banks at the same location in the model and the flood map by Graf. Further upstream the water also overflows. This area might have been captures if a fill raster had been placed in this location.

Critical point 12

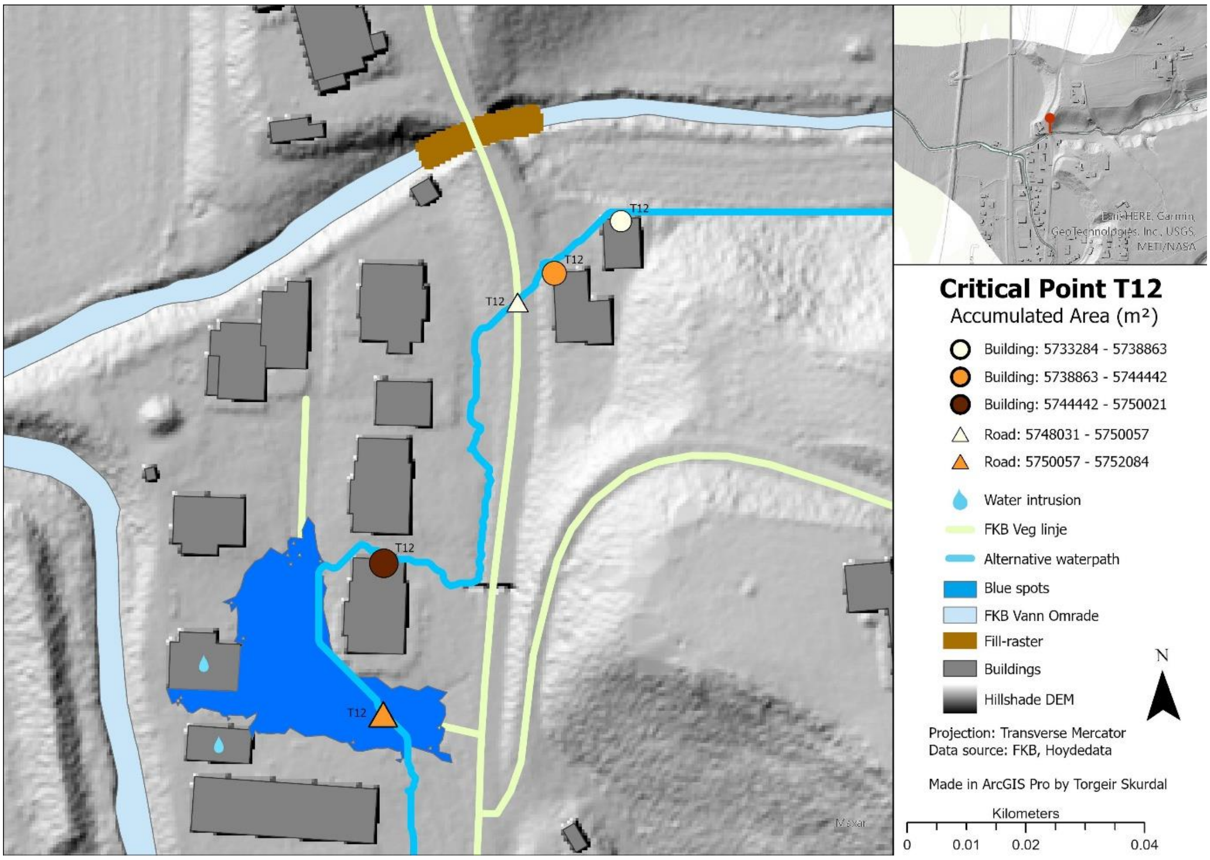


Figure 6.52: Showing the indicated damage caused to buildings and roads by critical point 12, from running the model.

ORIG_FID	objtype	bygningsnu	LayerName	Acc_Area	Damage_Class
1	Bygning	184537397	T12	5733284	5
2	Bygning	184537400	T12	5739380	5
3	Bygning	300160743	T12	5750021	5

Table 6.16: Showing identifying output attributes for critical point 12, building damage.

FID_T12	GATENAVN	LayerName	Acc_Area	Damage_Class
173	Nyhusvegen	T12	5748031	5
346	Nyhusvegen	T12	5752084	5

Table 6.17: Showing identifying output attributes for critical point 12, road damage.

The output from the model indicates that if critical point 12 is filled it has potential to cause great damage to several private properties, this is highlighted in Figure 6.52. The output attributes describing the damage class and the building numbers of the damage buildings are illustrated in Tables 6.16 and 6.17. It is also likely that water will accumulate in this area. To prevent damage, flood prevention measures in this location should be considered.

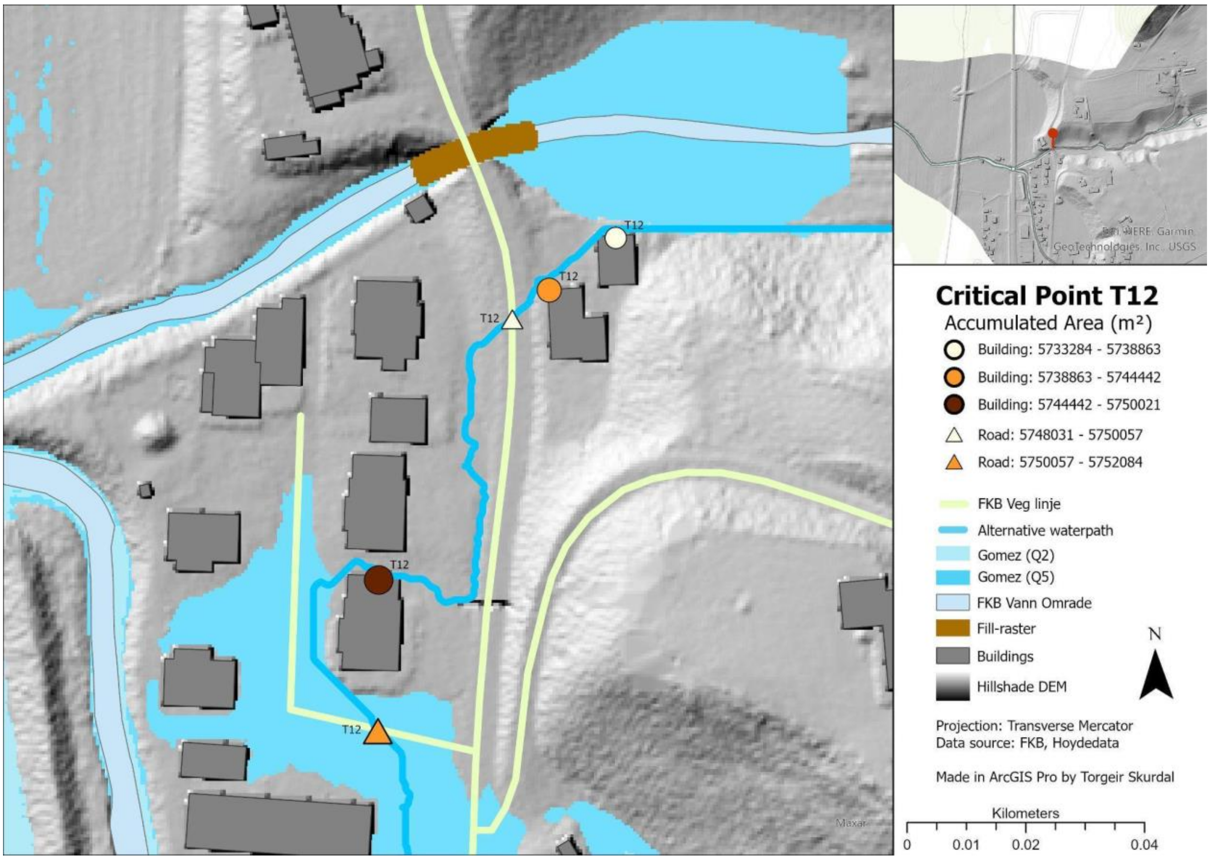


Figure 6.53 Showing the modeled damage caused to buildings and roads by critical point 12, compared to Flood map by Gomez (2021).

Comparing the model output with the flood maps by Gomez the modeled damage does not match up, so for this to happen a flood larger than a 5-year flood must happen. Looking at the water that accumulates top right in Figure 6.53 it looks like the water is on the verge of flooding over, and that the bridge that is critical point 12 will act as a bottle neck constricting the stream.

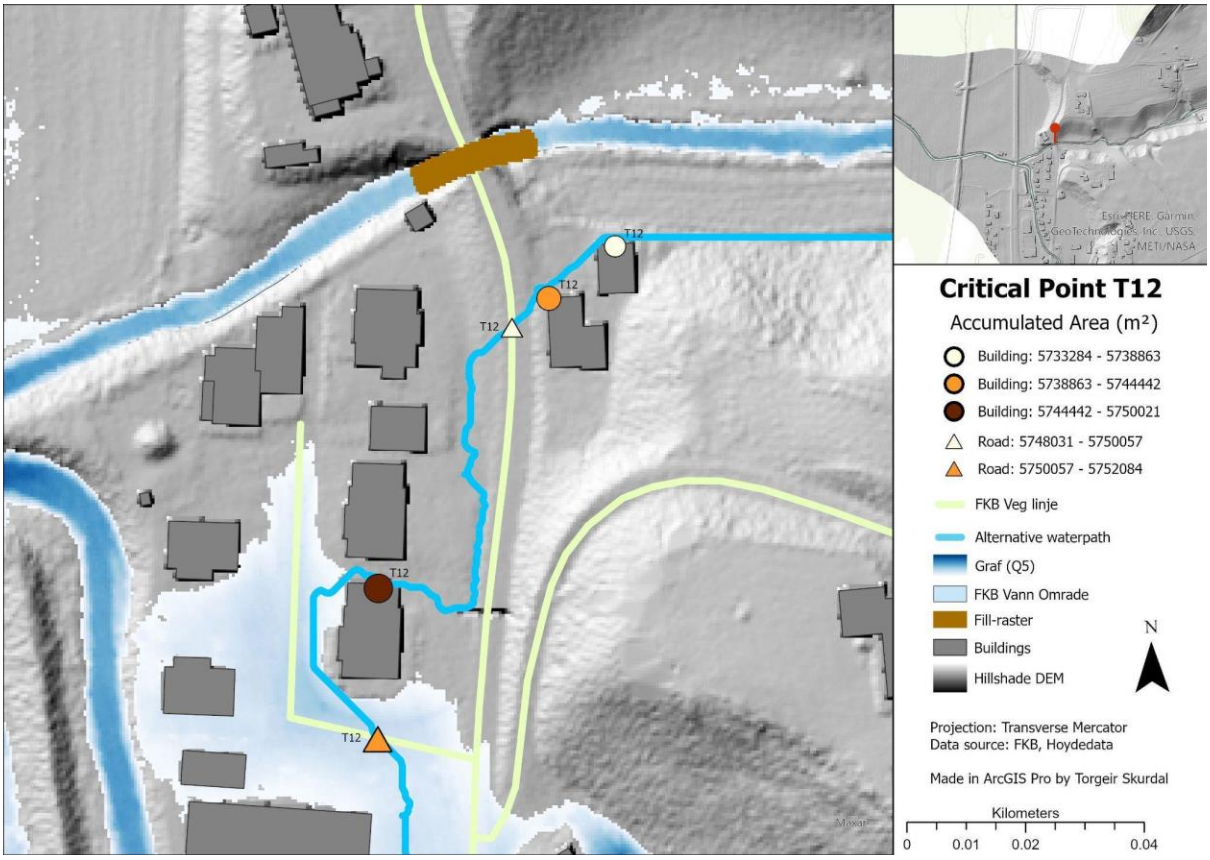


Figure 6.54: Showing the modeled damage caused to buildings and roads by critical point 12, compared to Flood map by Graf (2021).

Comparing the model output with the flood maps by Graf (Figure 6.54) the modeled damage does not match up with any of the maps, so for this to happen a flood larger than a 5-year flood must happen. Looking at the top right of the map, n the area where the maps by Gomez (Figure 6.53) indicates that the bridge will act as a bottleneck, this is not a problem, when comparing to Graf.

6.4 Flood Map comparison

To validate the output from the ModelBuilder models the damage points were compared to HEC-RAS flood zone maps by Graf (2021) and Gomez (2021). The flood map by Graf (2021) shows the flood zone for an event with a 5-year recurrence interval. Flood maps by Gomez (2021) show floods with recurrence intervals of 2 years and 5 years, illustrated in light blue (Q2) to blue (Q5), as seen in Figure 6.53. Overview maps showing the indicated damagepoints compared to the maps by Graf (2021) and Gomez (2021) can be found in Appendix A.

Flood map by Graf (2021) is illustrated in white to dark blue, symbolizing water depth, where dark color indicated deeper water (Figure 54). Despite modeling the same scenario, a Q5 flood in its entirety, they are quite different when comparing them. Graf's (2021) Q5 flood map covers a much larger area, using terrain covering much of the Kaldvella valley. Gomez's (2021) terrain only covers Ler. In addition, Graf's (2021) maps indicate larger damage compared to Gomez (2021), highlighted in Figure 6.55. The exact cause of this difference is not known, but it might be due to different depths in the stream channels. The extent of the terrains used in the two models might also play a role, where Graf has included a larger area, compared to Gomez, this is not shown in Figure 6.55. It might also come as a result of different channel depts used, or the difference in the flood duration, where Graf considered 12.00-18.00, and Gomez 00.00-01.00. For this study it will not make sense to further speculate on the differences in the two maps, and both present interesting results that are useful for comparing the output from this study.

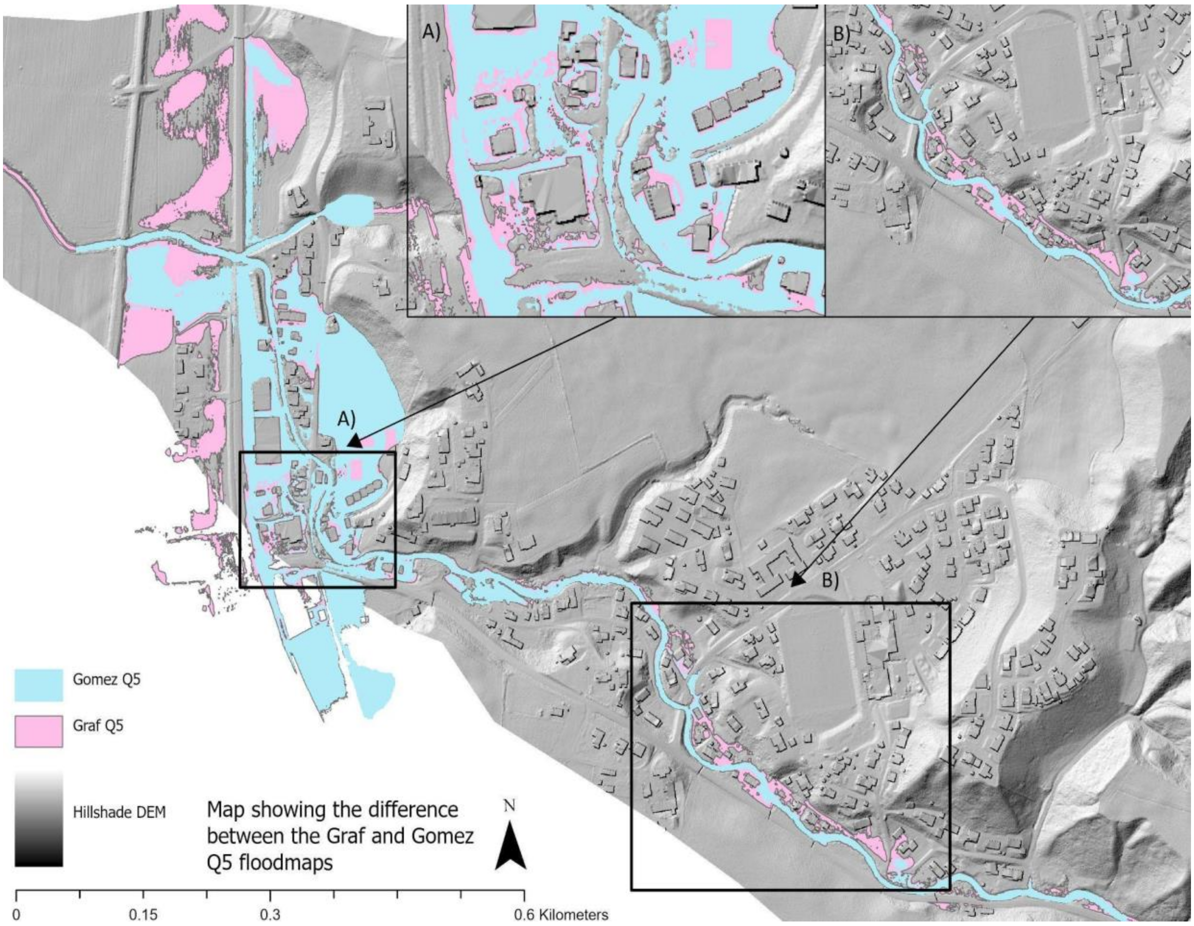


Figure 6.55: showing the difference between Q5 flood maps by Graf (2021) and Gomez (2021)

Despite the differences between the two model outputs as illustrated in, they are overall very similar in Ler village center (Figure 6.55; B), which is also one of the most important areas in the watershed, having both the E6 and several industrial buildings. In other areas larger differences can be observed, and Graf's output results in more damage compared to Gomez (Figure 6.55; A). The amount of area that will be flooded in the case of a 5-year flood is quite astonishing, considering that five years is quite likely to occur, with 20% change reach year. This really sets a 20-, 50- or 100-year flood into perspective.

The most important attributes that can be taken from the output attribute tables are the critical point that causes the damage, the total contributing area that will drain to the point in question, and the damage class ranging from 1 (low) to 6 (high). Knowing the contributing area, which is given in square meters it is possible to calculate the amount of water that will drain to a given point.

Comparing the flood maps by Graf (2021) and Gomez (2021) with the damage locations from the tools, puts things into perspective. Most of the damage points produced line up with the flood maps for Q5, with some points matching up with a Q2 flood. Seeing that a 2 year or 5-year flood has the potential to cause catastrophic damage in Kaldvella and Ler, puts things into perspective. The channel, spilling over in a 5-year flood, highlights exactly how sensitive Kaldvella is to flooding. This highlights just how disastrous a 10-, 20-, 50- or 100-year flood will be for this location. The only points that are not flooded in the case of a 5-year flood is the damage caused by critical point 12. When comparing this to the flood maps by Gomez, it is likely that this area can get flooded in the case of a larger flood than a 5-year flood.

It is important to consider the points as a whole, and it is very unlikely that the points will get affected individually. Many of the points are located close to each other and will therefore have a considerable impact on each other. Most of the Damage points from the ArcGIS model can be validated using the flood maps, where all of the indicated damage lines up with the extent of the flood maps. In some areas the ArcGIS model underestimates the extent of the water flow. Despite taking alternative paths the water will still accumulate, and the damages will propagate downstream. If flood protection measures are put in place it is important that this follows the principles of the three-step strategy, where the damages are handled in its entirety at its source, so that the damages won't accumulate downstream

7. Discussion

This study has developed GIS tools for evaluating the downstream damage a critical point can cause in the case of it getting blocked. To evaluate the performance of the tools a test has been run on the river Kaldvella in Melhus municipality. A complete set of tools from raw LAS data to potential damage points has been made. The evaluation of damage is based on the accumulated area, assuming that damage potential is directly correlated with accumulated area.

This thesis is one piece of a two-piece puzzle, where a complimentary study was meant to point out the critical locations, which were intended to be the input data for the tools developed in relation to this study. Since this did not happen, all the critical points that are used in this study are based on some known problem points, like shallow river stretches, bridges and sharp bends. Other points were strategically picked to investigate what will happen if the point is blocked. This study has the purpose of evaluating the performance of the tools. The output is therefore deemed more important than the input critical points being correct.

Faults and sources of error might have been introduced at different stages, and the whole study has to be seen in the light of these. The quality of the input data is important; everything from the field data, analysis and the interpretation of the output has to be considered. According to Longley et al. (2015) will the researchers' subjective assessments always be of importance in GIS representations. It is therefore important to reflect on these assessments and make them visible to the readers so they can make their own subjective assessments.

7.1 Evaluation of method

7.1.1 LiDAR-derived DEM

In this study the choice fell on DEM raster representation of the LiDAR derived terrain surface. There are several reasons for using raster rather than TIN, the first one being that there are simply more tools compatible with the raster format, and tools like flow direction and flow accumulation, are not compatible with TIN format. Since the raster data is a matrix composed of rectangular cells, manipulating the data, and adding or removing features is also relatively simple. The cell size of the raster was chosen on background of the results from the formula developed by McCullagh (1988). The formula recommends using a cell size of 0.45 meters. Using 0.5-meter raster cells makes more sense, due to map algebra and computations. A possibility could have been to use a cell size of 1 meter. This would have made the computations much faster, but the degree of detail would not be as good.

7.1.2 DEM modifications

In this study, one base raster was made, which was the basis for all analyses. To best reflect the real-world situation, some modifications to the DEM had to be made - buildings were raised, bridges removed, culverts were dropped, levees were added. Several changes have been made to the terrain since the capture of the LiDAR data, in addition to this a tool for creating rasterized blockages of the critical points has been made. When looking at the modeled flood paths, raising the buildings has the desired effect, as the water drains around the buildings rather than through them, although the optimal outcome in this situation would be to use the buildings in the LiDAR data. Since buildings have been built as well as demolished within the Kaldvella watershed, this would not represent the current situation. As a result, all the buildings have been raised by the same value of 5 meters, which produces satisfactory results.

Within the Kaldvella watershed, water and infrastructure such as roads, cross paths several times. A total of 30 road bridges can be found within the Kaldvella watershed (Figure 6.7). Removing the bridges was necessary since the bridges would act as obstacles in the stream path. This secures a more realistic flow path of the water. At the same time this eliminates the possibility of water to flow over the bridge surface. After visual inspection of the results from running the flow accumulation tool with bridges added to the DEM it was concluded that water flowing on top of the bridge surface has such a small impact that it was decided it would be best to remove them completely.

One of the biggest challenges in this project was related to the culvert and drainage pipes, since these can have a large impact on the flow of water and the inclusion or exclusion of these drainage features can have significant impact on the direction the water will flow. The data was acquired from the database of the Norwegian road authorities, through the database NVDB. In this database 63 registered culverts of varying size can be found within the Kaldvella watershed, and not adding these would have had significant impact on the flow of water. The probability that more culverts and drainage pipes exist within the Kaldvella watershed is quite high since drainage pipes related to farming or drainage of private roads might not be included in the dataset. This introduces some uncertainty, as it might have an impact on the accumulated water.

7.2. Hydrological modeling

The hydrological modeling in this study is based on the topographical features of the raster surface and the assumption that the water will follow the path of steepest descent. This is a simplification of reality and therefore introduces some limitations, since parameters like infiltration, evaporation and the amount of available water are not considered individually. These can be combined into one or more weight rasters to simulate the impeding effects of different surfaces, or local variation in rainfall. This was not done in this study, however, the hydrological modeling is nevertheless considered adequate and appropriate in the terms of scope and requirements for this thesis. Still, it is important to be aware of these limitations to help avoid misinterpreting the results. To give an example, the output value for the flood paths does not represent the amount of water, but rather the contributing area. This is still adequate to give an indication of the amount of water one can expect during a flood event.

7.2.1 Flow Direction

In the flow direction tool, there are several choices when it comes to flow direction algorithms. The D8 algorithm assumes that the water can only drain in eight different directions (Figure 4.8). This of course is an underestimation, and the DINF algorithm was also considered. When comparing the two, the results were very similar with only minor differences, where DINF seemed to spread out more in flat terrain. Due to the minor differences the choice fell on the D8 algorithm, since it is the simplest method and is less computationally heavy; it is also the standard method for the tool.

7.2.2 Flow Accumulation

The output raster from the Flow accumulation tool was compared and validated against the FKB vannområde. Since the FKB- vannområde only includes the larger streams, the data was supplemented with FKB-vanngrense which includes the smaller streams, these were combined into FKB_All_Water to be used in the validation. This showed that the modeled water line to a very large degree lined up with the FKB data, with some minor deviations in some areas. When comparing both with aerial photos the flow accumulation raster seemed to be more correct, compared to the FKB data, where the FKB data seemed to be more generalized in locations with small streams.

7.2.3 Critical points

The critical points used to test the model is a mixture of known problem points, and bridges. It is not guaranteed that all the bridge locations have the ability to cause any damage. The critical points was chosen to test the performance of the model. At present the critical points are converted to raster, which is mosaiced into the main DEM. The size of the fill rasters are uniform, this may influence the accuracy of the output drainage lines. In Ler village center, Kaldvella is running through relatively flat terrain; this should be considered when applying the mode to a river that is steeper than Kaldvella. Another study will point out the critical points that will be the input for this study.

7.2.4 Depressions

The depressions in the DEM are based exclusively on the DEM raster surface, and errors related to the raster can have significant impact on these. In this study the depressions are referred to as bluespots. The watershed tools used to delineate these locations are based on the same Flow direction and flow direction algorithm as the tools described above. This is of course a simplification and errors in relation to the generation of the DEM, as well as the simple nature of the D8 algorithm should be considered when judging the tools output. When comparing the bluespots to the more reliable flood maps the locations match up well, and flood maps can be used to verify the results.

7.2.5 Threshold values

Since the aim of this study is to find downstream damage potential, some threshold values had to be set. Sellæg (2016) used a threshold value of 25 000m². His choice was made based on threshold values used by Meiforth (2013) and Bratlie (2013). Values below the threshold are assumed as too small to cause any substantial damage.

Like the study by Sellæg (2016) investigating streams in Lillehammer, the Kaldvella watershed has a varied topography, at the same time the population and houses are concentrated in the village center of Ler, which also coincide with the area of highest water accumulation. A threshold value of 25 000m² is therefore used in this thesis.

To visualize the flood paths the vector format is the most suitable, since it gives a better visual representation of the drainage lines, compared to raster. The vector data can be scaled in the symbology tab to be more appropriately be used in maps of different scales, while this is not possible for raster. Vector format is also better for finding the points where water and roads or buildings intersect. This is also possible to do through raster operations, but at some point, the damage points have to be vectorized to be illustrated appropriately.

The classification of the damage points is also inspired by Sellægg (2016), Meiforth (2013) and Bratlie (2013). These have been divided into 6 damage classes, with 1 being the lowest possible damage and 6 being the highest possible. The values span from 25 000m² to 30 000 000m² which is rounded up from the maximum amount of water that can drain through the watershed. When evaluating damage potential, this scale has worked well, but the values are not directly transferable to flood size. NVE defines small rivers as rivers with watersheds less than 50km². At present the scale is based on the size of the watershed of Kaldvella, when applying the model to other watersheds the size of the watershed should be set as the highest value.

7.2.6 ModelBuilder models

The results were compared to HEC-RAS flood maps. This helped validate the performance of the tools, as well as point out some weaknesses, such as the models being sensitive to the location of the critical point input. The functions were to some degree inspired by SCALGO Live since this is the most comparable tool on the market, and both drainage line and depression are included in the models developed in this study.

8.Conclusion

This study has produced a method for mapping potential damage caused by blocked critical points in a small watershed during a heavy rainfall event. The method is based on the assumption that the water will follow low breaks in the terrain and the steepest gradient downstream. The critical points used in the study are known problem points, as well bridge locations that are sensitive to blocking and can act as bottlenecks in the stream path if sediments or driftwood finds its way into the stream.

Drainage directions were calculated using the relatively simple D8 algorithm, based on a high-resolution raster DEM derived from LiDAR data. Hydrological analysis in ArcGIS Pro was used to find flow paths, and the accumulated area in square meters was calculated to identify water paths that have large enough accumulated area to cause damage. The calculated flood paths were based on a hydrologically corrected DEM with bridges removed and culverts dropped. The critical points were filled, using GIS methods developed in this study.

This study has shown the importance of viewing all of the critical points as a whole, and not just as single points. Critical points one to eight (figure x) are located within a relatively small area and in the case of a flood they will have an impact on each other.

From this study it is clear that flood preventive measures should be considered. Everything from culverts to the stream channel itself should be evaluated. The damage caused by a 5-year flood is remarkable, and with more extreme weather and both larger amounts of annual precipitation, as well as an increase in heavy precipitation events Ler and Kaldvella (and presumably countless other small watersheds in Norway) are very sensitive to flooding.

In the planning of flood protective measures, it is also important to remember the three-step strategy (remind the reader what this is here), and the concept of handling the problem at its source in such a way that the problem is not simply pushed downstream.

The calculated damage based on the GIS tools and workflows developed in this thesis have been validated by comparing the accumulated flood paths with streams and river data from FKB data sets, as well as HEC-RAS flood maps by Graf (2021) and Gomez (2021). When compared to the flood maps it is clear that Kaldvella is sensitive to flooding. This is highlighted by the fact that the damages indicated by the tools developed in this thesis match up with flood maps. As mentioned in the discussion, there are possible sources of error linked to every part of this study, and the answers below should bear this in mind.

“Can GIS methodologies be applied to indicate downstream damage potential of critical points in a small watercourse?”

Considering all possible methods, it is clear that indication of downstream damage can be predicted with GIS tools. GIS tools is a wide term, and the quality of the output will vary, based on a range of factors, everything from the quality of the input the data to the ability of the GIS tools in questions. GIS will always be a strategic simplification of the world, where what is considered important will vary between individuals. Therefore, if the user is biased, lacks knowledge of the GIS tools or knowledge of the input data, this can impact both the output results produced during the process, as well as how the results are interpreted.

“Is it possible to automate this process in an ArcGIS ModelBuilder tool?”

From this thesis it is clear that indication of damage caused by critical points can be automated in a GIS tool. Still, it is important to bear in mind that indication does not guarantee that the results from the tools are reliable, and there are uncertainties linked to all parts of this study. When comparing the output from the tools developed in this study to other models, the results are however, very similar (see chapters/figures x-y, this thesis). The tools appear to underestimate the extent of water flow, compared to HEC-RAS flood maps. This is something that has been known from the very start of this study, and this is simply a limitation of the ArcGIS Pro program. The results from these tools should always be compared and validated against known damage or other more reliable models such as HEC-RAS flood maps. For indication of potential damages caused by critical points, the tools work as intended.

“Can the model output be used to evaluate if flood protective measures are needed?”

The results from this study produce adequate results for indicating problem points, but to use the results from the model to place flood protective measures are not enough. When evaluating the placement of flood protective measures, it is important to see the situation from all angles. Following a three-step strategy when planning the flood protection, is also important, as modifications made to the terrain might push the problem upstream or downstream. Handling the problem in its entirety should be prioritized. Until the model is further developed, the results should be validated using results from hydraulic models or known flood extents and flood damages.

8.1 Further work

Compared to hydraulic models such as HEC-RAS, GIS based hydrological modeling is simplistic, even though many factors can be simulated by adding features or weight rasters to simulate things such as precipitation, water infiltration etc. When further developing the tools from this study, adding such features should be a priority. Including precipitation data, for example, will help to better understand how local downpour in the watershed will affect the water accumulation and provide important insight into how different water volumes will impact the floodways, thus increasing understanding of the impact of a 10-, 50- or 200-year flood.

To help define the size of the flood needed to achieve the damage indicated by the tools in this study, the flood maps were limited to 5-year floods (Q5). Comparing flood maps showing floods larger than 5-year recurrence interval, could be helpful, when further developing the methodology.

To make these tools more accessible it is a very real possibility to develop the toolsets into web-based applications (e.g., ArcGIS Online). By utilizing crowd sourcing software, it would also be possible to verify and compare the modeled flood paths, with actual flood damages.

At present, the tools only block one critical point at a time. When considering what can be done and further developed with the tools, the possibility to iterate random combinations of critical points filled or iterate through all possible combinations of filled critical points, could help give more realistic results when considering the accumulated area.

In the study area there are many registered culverts but investigating the hillshade DEM and looking at aerial photos it seems very likely that there are more culverts and drainage pipes that are not included in the dataset. This is probably because there are several smaller roads, such as gravel roads and mountain roads, and roads used in relation to forestry that are privately owned, and therefore not included in the national data. Developing methods for accurately extracting these culvert locations will be important in further development. To some degree it is possible to verify these locations manually by looking at aerial photographs, but some areas are very hard to see, and

Some other ideas for future research include:

- Comparing the volume of floodwater with the volume of the stream channel. This can help further develop the classification to reflect the size of the floods.
- Find better methods for applying the symbology, at current this must be done manually.
- Reevaluate the damage classification scale to work better with standard scales, such as recurrence interval. It would also be possible, to go the other way, and adapt the scale (1 to 6) to the recurrence interval.
- Use of elevation data with finer resolution and more sophisticated algorithms in the hydrological modeling can possibly provide a more detailed representation of the drainage patterns. This requires high quality input data to avoid errors, and powerful hardware to avoid time consuming processing.

9. Reference list

Aber, J.S, Marzolff, I, Ries, J.B & Aber, S.W.E. (2019) Chapter 3 - Principles of Photogrammetry, Small-Format Aerial Photography and UAS Imagery (Second Edition), Academic Press, https://doi.org/10.1016/B978-0-12-812942-5.00003-3

Arun, P. V. (2013). A comparative analysis of different DEM interpolation methods. The Egyptian journal of remote sensing and space science., Volume 16(Issue 2). doi: http://doi.org/10.1016/j.ejrs.2013.09.001

Balstrøm, T (2021). Model bluespots to map flood risk, Retrieved 15.11.2021 from https://learn.arcgis.com/en/projects/model-bluespots-to-map-flood-risk/

Baltsavias, E. P. (1999). Airborne laser scanning: Basic relations and formulas. ISPRS Journal of Photogrammetry and Remote Sensing, 54, 199–214

Barr, D.R, Mansager,B.(1996)Terrain map resolution,Mathematical and Computer Modelling, Volume 23, Issues 1–2,p.39, doi:https://doi.org/10.1016/0895-7177(95)00214-6.

Bruland, O. (2020). Trygg Elv: Identifisering og sikring av kritiske vassdrag. Retrieved 16.03.2022 from https://prosjektbanken.forskningsradet.no/project/FORISS/309731

Bratlie, R. (2013) GIS finner flomveiene, Retrieved 07.05.2022 from https://vannforeningen.no/wp-content/uploads/2015/06/2013_875979.pdf

Brunner, G.W. (2021). HEC-RAS River Analysis System. In U. A. C.o. Engineers, Retrieved 14.03.2022 from https://www.hec.usace.army.mil/software/hec-ras/documentation/HEC-RAS%205.0%20Users%20Manual.pdf

Dannevig, P. (2020) Trøndelag (klima). Retrieved 16.03.2022 from https://snl.no/Tr%C3%B8ndelag_-_klima

Direktoratet for byggkvalitet. (n.d.). Hvordan defineres flomfaren? Retrieved 03.05.2022 from https://dibk.no/saksbehandling/kommunalt-tilsyn/temaveiledninger/utbygging-i-fareomrader-bokmal/4.-flom/4.2.-sikkerhet-mot-flom/

Elias, S. A. (2013). The Quarternary. Reference Module in Earth Systems and Environmental Sciences. doi: https://doi.org/10.1016/B978-0-12-409548-9.05350-1

Esri. (2015). Independent Report Highlights Esri as Leader in Global GIS Market. Retrieved 14.03.2022 from https://www.esri.com/about/newsroom/announcements/independent-report-highlights-esri-as-leader-in-global-gis-market/#:~:text=Independent%20Report%20Highlights%20Esri%20as%20Leader%20in%20Global%20GIS%20Market,-March%2002%2C%202015&text=Redlands%2C%20California%E2%80%94ARC%20Advisory%20Group,from%20the%20second%2Dlargest%20supplier

Esri. (n.d.a). Add a submodel to a model. Retrieved 02.03.2022 from https://pro.arcgis.com/en/pro-app/2.8/help/analysis/geoprocessing/modelbuilder/add-a-submodel-to-a-model.htm

Esri. (n.d.b). An overview of the interpolation toolset. Retrieved 02.11.2021 from https://pro.arcgis.com/en/pro-app/2.8/tool-reference/spatial-analyst/an-overview-of-the-interpolation-tools.htm

Esri (n.d.c). Con (Spatial Analyst). Retrieved 02.11.2021 from https://pro.arcgis.com/en/pro-app/2.8/tool-reference/spatial-analyst/con-.htm

Esri. (n.d.d). Comparing interpolation methods. Retrieved 02.11.2021 from https://desktop.arcgis.com/en/arcmap/latest/tools/spatial-analyst-toolbox/comparing-interpolation-methods.htm

Esri. (n.d.e). IDW (Spatial Analyst). Retrieved 02.11.2021 from https://pro.arcgis.com/en/pro-app/2.8/tool-reference/spatial-analyst/idw.htm

Esri. (n.d.f). Iterators. Retrieved 03.03.2021 from https://pro.arcgis.com/en/pro-app/latest/help/analysis/geoprocessing/modelbuilder/iterators-for-looping.htm

Esri. (n.d.g). Kriging (Spatial Analyst). Retrieved 02.11.2021 from https://pro.arcgis.com/en/pro-app/2.8/tool-reference/spatial-analyst/kriging.htm

Esri. (n.d.h). How Fill works. Retrieved 03.02.2022 from https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/how-fill-works.htm

Esri. (n.d.i). How Flow Accumulation works. Retrieved 03.02.2021 from https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/how-flow-accumulation-works.htm

Esri. (n.d.j). How Flow Direction works. Retrieved 03.02.2021 from https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/how-flow-direction-works.htm

Esri. (n.d.k). How Natural Neighbor works. Retrieved 02.11.2021 from https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/how-natural-neighbor-works.htm

Esri. (n.d.l). How Sinks works. Retrieved 03.02.2022 from https://pro.arcgis.com/en/pro-app/2.8/tool-reference/spatial-analyst/how-sink-works.htm#:~:text=A%20digital%20elevation%20model%20(DEM,of%20elevations%20to%20integer%20numbers.

Esri (n.d.m). LAS dataset properties: Surface Constraints tab. Retrieved 22.03.2022 from https://pro.arcgis.com/en/pro-app/2.7/help/data/las-dataset/las-dataset-properties.htm

Esri (n.d.n). Storing lidar data. Retrieved 21.03.2022 from https://pro.arcgis.com/en/pro-app/2.7/help/data/las-dataset/storing-lidar-data.htm

Esri. (n.d.o). Use ModelBuilder. Retrieved 01.03.2022 from https://pro.arcgis.com/en/pro-app/latest/help/analysis/geoprocessing/modelbuilder/modelbuilder-quick-tour.htm

Esri (n.d.p). Weighted Overlay (Spatial Analyst). Retrieved 02.11.2021 from https://pro.arcgis.com/en/pro-app/2.8/tool-reference/spatial-analyst/weighted-overlay.htm

Finans Norge. (2022) Klimarapport Finans Norge 2022. Retrieved 02.05.2022 from https://www.finansnorge.no/siteassets/statistikk/statistikk-og-nokkeltall-skade/klimarapporten/finans-norge-klimarapport-2022-dobbeltsider.pdf

Geovekst. (2022). Felles Kart dataBase (FKB). Retrieved 31.01.2022 from: https://register.geonorge.no/geodatalov-statusregister/felles-kartdatabase-fkb/0e90ca71-6a02-4036-bd94-f219fe64645f

Goodchild, M. F., & Zhang, J. (2002). Uncertainty in Geographic Information. doi:10.4324/9780203471326

Graf, M. S. (2021). Identification of critical locations during floods in the river Kaldvella. (Bachelor Thesis). Norwegian University of Science and Technology, Guarneri, J. C., & Weih, R. C. (2012). Comparing methods for interpolation to improve raster digital elevation models. Journal of Arkansas academy of science, 66. doi:https//:doi.org/10.54119/jaas/2012.6609

Halleraker, J. H. (2022). Bunnis. Retrieved 18.03.2022 from https://snl.no/bunnis

Hanssen-Bauer, I., Førland, E. J., Haddeland, I., Hisdal, H., Mayer, S., Nesje, A., . . . Ådlandsvik, B. (2015). Klima i Norge 2100 (2/2015). Retrieved 26.04.2022 from https://www.miljodirektoratet.no/globalassets/publikasjoner/m406/m406.pdf

Hengl, T., & Evans, I. (2009). Mathematical and digital surface models of the land surface. Development in soil studies, Volume 33, 31-63.

Henning, I. M. (2015). Flomvei og aktsomhetskartlegging ved bruk av GIS over nedbørsfeltet Brandrudsåa i Gudbrandsdalen. (Master thesis). Norges teknisk-naturvitenskapelige universitet, ntnuopen. Retrieved 03.05.2022 from https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2372999?locale-attribute=no

Høydal, I. (2016). LASERSKANNING FOR ELVEKARTLEGGING: Kartlegging av Gaula. NVE sak 201603593. Retrieved 31.03.2022 from Hoydedata.no

Husebye, S. (2016). Flomberegning og vannlinjeberegning for Kaldvella på Ler i Sør-Trøndelag. Retrieved 26.04.2022 from https://docplayer.me/222406897-Flomberegning-og-vannlinjeberegning-for-kaldvella-pa-ler-i-sor-trondelag.html

Jukil, G., & Al-Hadad, B. (2017). Highway Vertical Alignment Optimization Using Genetic Algorithm. In V. a. r. d. m. i. g. space (Ed.).

Kalsnes, B., Solheim, A., Sverdrup-Thygeson, S., Dingsør-Dehlin, F., Wasrud, J., Indrevær, K., & Bergbjørn, K. (2021) Flom og skred – sikringsbehov for eksisterende bebyggelse (FOSS), Rapport Nr. 20/2021, Norges vassdrags- og energidirektorat, Retrieved 09.10.2021 from https://publikasjoner.nve.no/rapport/2021/rapport2021_20.pdf

Kartverket. (2021). Høgdedata og djupnedata. Retrieved 21.02.2022 from https://www.kartverket.no/api-og-data/terrengdata

Kielland, O. F. (2020). Development of a GIS-tool for stormwater calculations – what can be achieved given few input values? (Master thesis). Norges miljø- og biovitenskapelige universitet., Brage. Retrieved 15.03.2022 from https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2682065

Leine, A-l.Ø. (2017) Flomberegning for Storelva I Utvik (087.4Z) Norges vassdrags- og energidirektorat, NVEs hustrykkeri. Retrieved 24.04.2022 from https://publikasjoner.nve.no/rapport/2017/rapport2017_94.pdf

Lindholm, O. (2008). Veiledning i klimatilpasset overvannshåndtering. Norsk Vann. Retrieved 15.03.2022 from https://www.nb.no/items/80200efa602d5a6855f622af3cc4a31f?page=0&searchText=Veiledning%20i%20klimatilpasset%20overvannsh%C3%A5ndtering

Liu, X. (2008). Airborne LiDAR for DEM generation: some critical issues. Progress in Physical Geography: Earth and Environment, 32(1), 31-49. doi:10.1177/0309133308089496

Lohani, B. (2008). Airborne Altimetric LiDAR: Principle, Data collection, processing and Applications In.

Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). Geograpchical information science and systems (4.th ed.): Wiley.p. 61, 67, 68, 313, 314, 316.

Martinn, O. (2011). Surfaces represented by a Digital Surface Model and Digital Terrain Model.

Melhus Kommune. (2016). ROS-ANALYSE - Områdeplan Ler. Retrieved 26.04.2022 from http://webhotel3.gisline.no/GisLinePlanarkiv/5028/2015002/Dokumenter/Ros%20analyse%20Ler.pdf

Miljødirektoratet. (2021). Overvann. Retrieved 14.03.2022 from https://www.miljodirektoratet.no/ansvarsomrader/vann-hav-og-kyst/overvann/

Muhadi, N. A., Abdullah, A. F., Bejo, S. K., Mahadi, M. R., & Mijic, A. (2020). The Use of LiDAR-Derived DEM in Flood Applications: A Review. Remote Sensing, 12(14), 2308. Retrieved 02.11.2021 from https://www.mdpi.com/2072-4292/12/14/2308

Myhre, G., Alterskjær, K., Stjern, C.W., Hodnebrog, Ø., Marelle, L., Samset, B.B., Sillmann, J., Schaller, N., Fischer, E., Schulz, M. & Stohl, A. Frequency of extreme precipitation increases extensively with event rareness under global warming. *Sci Rep* **9,** 16063 (2019). https://doi.org/10.1038/s41598-019-52277-4

NEVINA. (n.d) Brukerveiledning. Retrieved 02.10.2021 form https://nevina.nve.no/help/Brukerveiledning_NEVINA.pdf

Nordeidet, B., Schow, C., & Killerich, T. (2015). Overvannsflom - metoder for kartlegging og
analyser. Miljødirektoratet. Retrieved 03.05.2022 from https://bit.ly/2T9BcpG

Norges vassdrags- og energidirektorat. (2014). Flaum- og skredfare i arealplanar; Revidert 22. mai 2014. Retrieved 31.01.2022 from https://publikasjoner.nve.no/retningslinjer/2011/retningslinjer2011_02.pdf

Norges vassdrags- og energidirektorat. (2015). Flaumfare langs bekker, NVEs hustrykkeri Retrieved 09.02.2022 from http://publikasjoner.nve.no/veileder/2015/veileder2015_03.pdf

Norges vassdrags- og energidirektorat. (2020*a*). Hva er flom? Retrieved 15.03.2022 from https://www.nve.no/naturfare/laer-om-naturfare/hva-er-flom/

Norges vassdrags- og energidirektorat. (2020*b*). Overvann i arealplanlegging. Retrieved 15.03.2022 from https://www.nve.no/arealplanlegging/overvann-i-arealplanlegging/

Norges vassdrags- og energidirektorat. (2022). Regnflom. Retrieved 15.03.2022 from https://nve.no/naturfare/laer-om-naturfare/flom/regnflom/

Norges vassdrags- og energidirektorat. (2008). Planlegging og utbygging i fareområder langs vassdrag. Retningslinjer nr. 1/2008. Retrieved 07.05.2022 from: https://publikasjoner.nve.no/retningslinjer/2008/retningslinjer2008_01.pdf

Norges vassdrags- og energidirektorat. (2018). Tilskudd til kartlegging av kritiske punkt i bekker og bratte vassdrag [Press release]. Retrieved 09.02.2022 from https://www.nve.no/naturfare/utredning-av-naturfare/om-kart-og-kartlegging-av-naturfare/om-kartlegging-av-flaumfare/tilskudd-til-kartlegging-av-kritiske-punkt-i-bekker-og-bratte-vassdrag/

O'Callaghan, J. F. M., D.M. (1984). The extraction of drainage networks from digital elevation data. Computer Vision, Graphics, and Image Processing, Volume 28(Issue 3), Pages 323-344. doi: https://doi.org/10.1016/S0734-189X(84)80011-0.

Olje- og energidepartementet. (2021). Regjeringen gir mer penger til flom- og skredsikring [Press release]. Retrieved 18.04.2022 from https://www.regjeringen.no/no/aktuelt/regjeringen-gir-mer-penger-til-flom-og-skredsikring/id2884655/

Paus, K. (2018). Forslag til dimensjonerende verdier for trinn 1 i

Norsk Vann sin tre-trinns strategi for håndtering
av overvann Vann. Retrieved 15.03.2022 from https://vannforeningen.no/wp-content/uploads/2018/07/Paus.pdf

Pedersen, T. B., Bratlie, R., Verbaan, I. J., Sandal, B., Solbrå, S. T., & Hagerup, T. G. (2022). Rettleiar for handtering av overvatn i arealplanar Retrieved 15.03.2022 from https://www.nve.no/arealplanlegging/overvann-i-arealplanlegging/

Polat, Z. A., & Alkan, M. (2019). Raster and Vector data models. In.

Roald, L. A. (2015). Flommen på Østlandet i mai 2013. NVEs hustrykkeri Retrieved 24.04.2022 from https://publikasjoner.nve.no/rapport/2015/rapport2015_21.pdf

Rohr-Torp, E.(1994) Present uplift rates and groundwater potential in Norwegian hard rocks. Noreges geologiske undersøkelse, Bull. 426, 47-52. Retrieved 02.03.2022 from:
https://www.ngu.no/filearchive/NGUPublikasjoner/Bulletin426_47-52.pdf

Romano, M. E. (2004). Innovation in LiDAR processing technology. Photogrammetricengineering and remote sensing, 1201-1206. Retrieved 03.05.2022 from
http://www.lidar.com.br/arquivos/Innovation_in_Lidar_Processing_Technology.pdf

Rød, J. K. (2015). GIS; Verktøy for å forstå verden: Fagbokforlaget. p. 191

Røstum, J., Bruaset, S., Sekse, T., Bjørnsen, B., Uribe, C., & Markhus, E. (2014). Åpne flomveger i bebygde områder. Norsk Vann Rapport (204 – 2014). Retrieved 15.03.2022 from
https://www.mercell.com/m/file/GetFile.ashx?id=109745313&version=0

SCALGO. (u.dat). SCALGO Live Documentation. Retrieved 14.03.2022 from https://scalgo.com/en-US/scalgo-live-documentation/analysis/flash-flood-map

Sellæg, S. A. (2016). Modellering av flomveier på Lillehammer, en GIS-studie av åpne og blokkerte dreneringsveger i skadepunkter fra flommen juli 2014. (Master). NTNU, Retrieved 09.02.2022 from https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2449675

Silva, K., Calasans, N., Mattos, J., & Santos, J. (2016). Differences between a TIN and a Raster. In. Journal of Hyperspectral Remote Sensing.

Skaaraas, H., Hansen, A. J., Riise, E., Stenersen, J., Refling, D., Johansen, R., Kippeberg, G. (2015). Overvann i byer og tettsteder; som problem og ressurs.

Retrieved 15.03.2022 from
https://www.regjeringen.no/contentassets/e6db8ef3623e4b41bcb81fb23393092b/no/pdfs/nou201520150016000dddpdfs.pdf

Skjåk Kommune (2021) 20648 Flomsikring av Bismo mot Ottaelva, Skjåk kommune, Innlandet fylke. Municipal hearing 19.05.2021. Retrieved 24.04.2022 from https://www.skjaak.kommune.no/_f/p3/i095f34c0-2fb1-4097-ac49-58c6b5d54bfc/nves-tiltaksplan-horingsutgave-20648-flomsikring-av-bismo-mot-ottaelva-skjak-kommune-datert-19052021.pdf

Solberg, I. L. (2018). Geologi og landskapsutvikling langs Gaula for lokaliteter knyttet til arkeologiske utgravinger Retrieved 31.01.2022 from https://www.ngu.no/upload/Publikasjoner/Rapporter/2018/2018_022.pdf

Steffen, H & Wu, P (2011)
Glacial isostatic adjustment in Fennoscandia—A review of data and modeling, Journal of Geodynamics, Volume 52, Issues 3–4, p.169-204, https://doi.org/10.1016/j.jog.2011.03.002.

Stenius, S., Glad, A. P., Wang, K. T. & Væringstad, T. (2015). Veileder for flomberegning i små uregulerte felt, NVEs hustrykkeri. Retrieved 09.02.2022 from https://publikasjoner.nve.no/veileder/2015/veileder2015_07.pdf

Stenius, S., Glad, P. A., & Wilson, D. (2014). Karakterisering av flomregimer. Retrieved 15.03.2022 from Norges vassdrags- og energidirektorat: http://publikasjoner.nve.no/rapport/2014/rapport2014_35.pdf

Stoeckle, P., & Montgomery, C. (2013). Vector and raster data In.

Stusdal, V. (2005). Oppfølging av særlig verdifulle kulturlandskap i Sør-Trøndelag.Fylkesmannen i Sør-Trøndelag, avdeling for landbruk og bygdeutvikling.

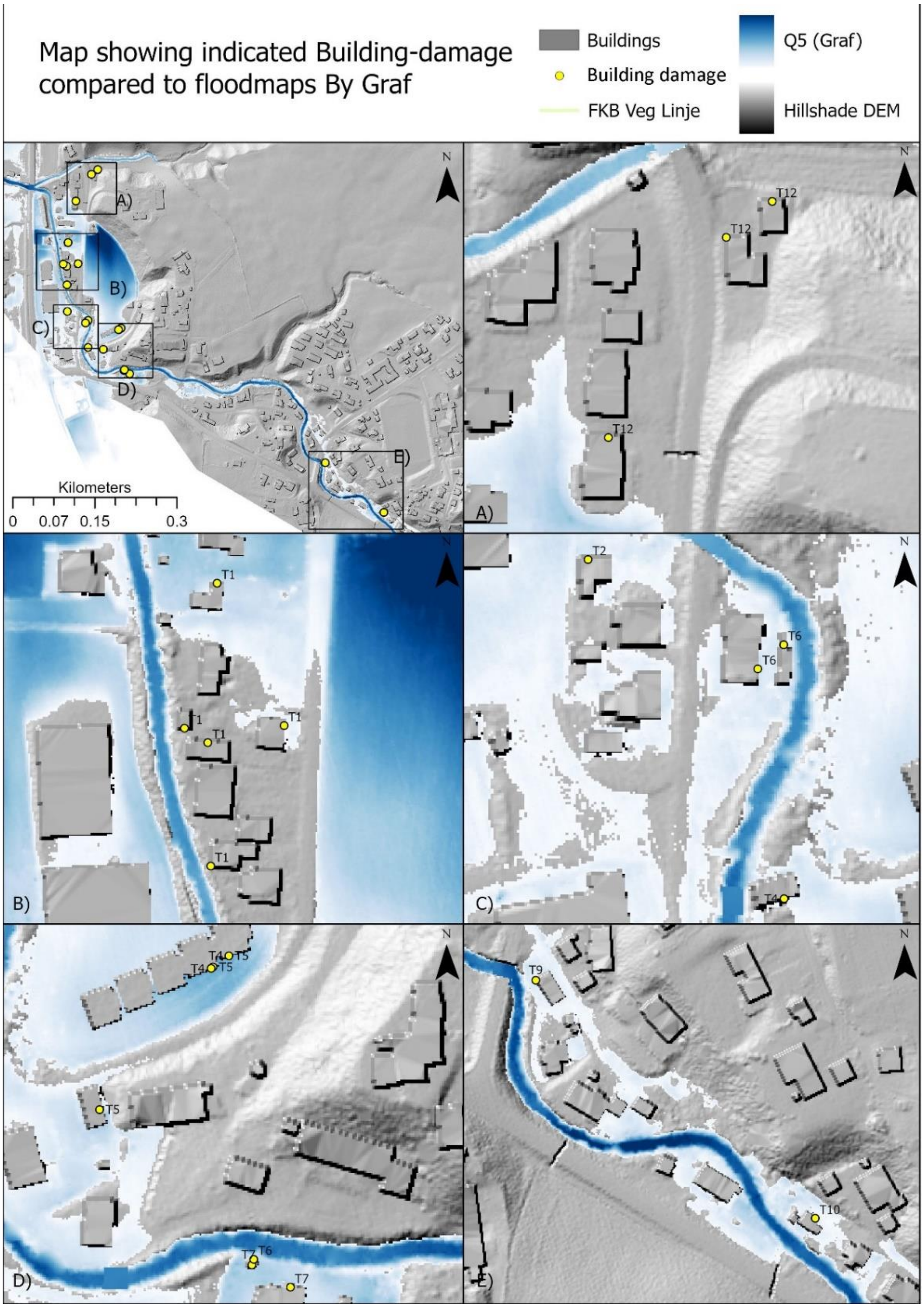
Tarboton, D. G. (1997). A new method for the determination of flow directions and upslope areas in grid digital elevation models. WATER RESOURCES RESEARCH, VOL. 33, 309-319. doi: https://doi.org/10.1029/96WR03137

Thorsnæs, G. (2021). Gauldalen. In Store norske leksikon. Retrieved 31.01.2022 from https://snl.no/Gauldalen

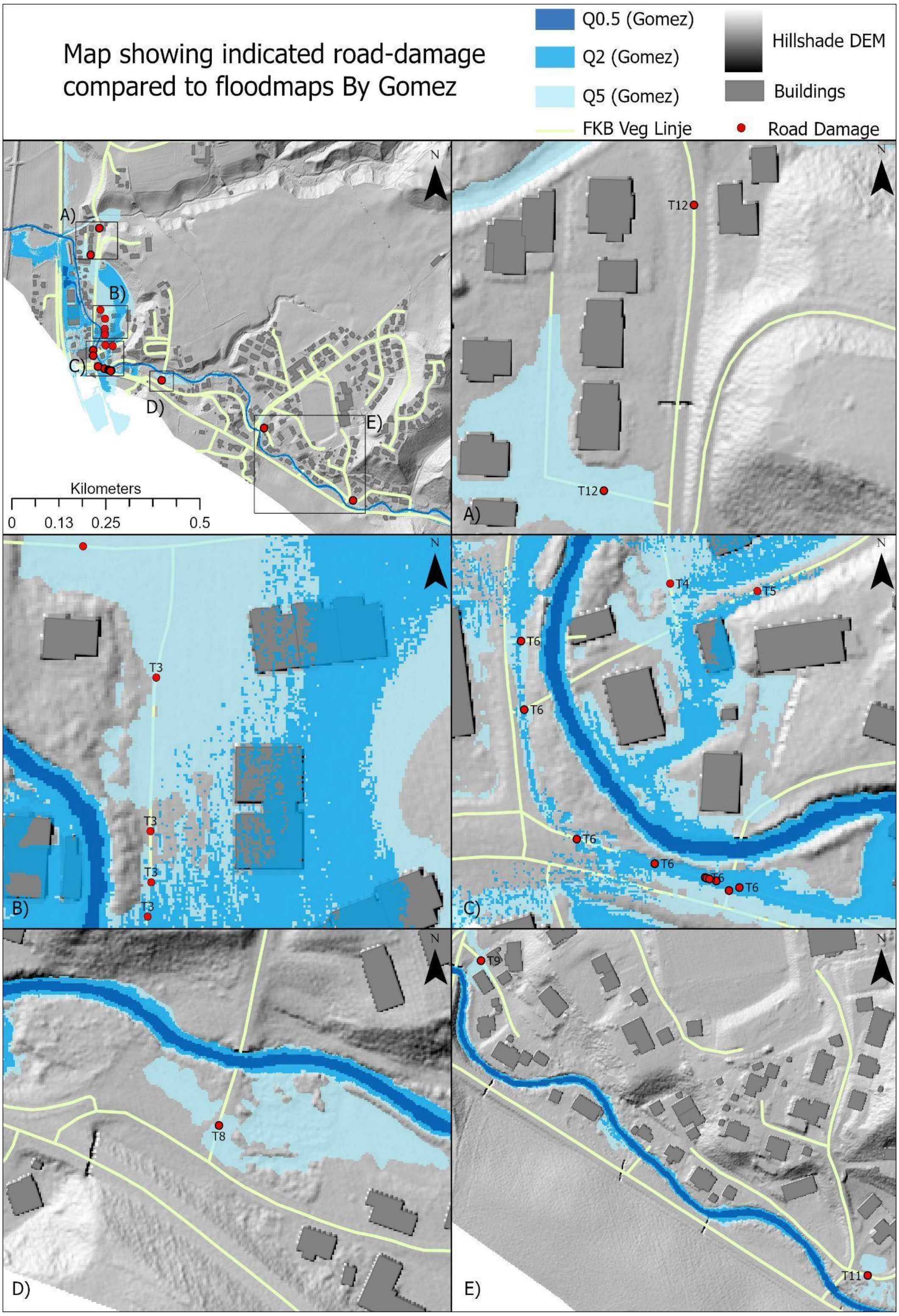
Viréhn, P. (2014). Water on Devious ways - A GIS Analysis. Retrieved 09.02.2022 from https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2448421/Per%20Lars%20Erik%20Vihr%c3%a8n.pdf?sequence=1&isAllowed=

10. Appendix A

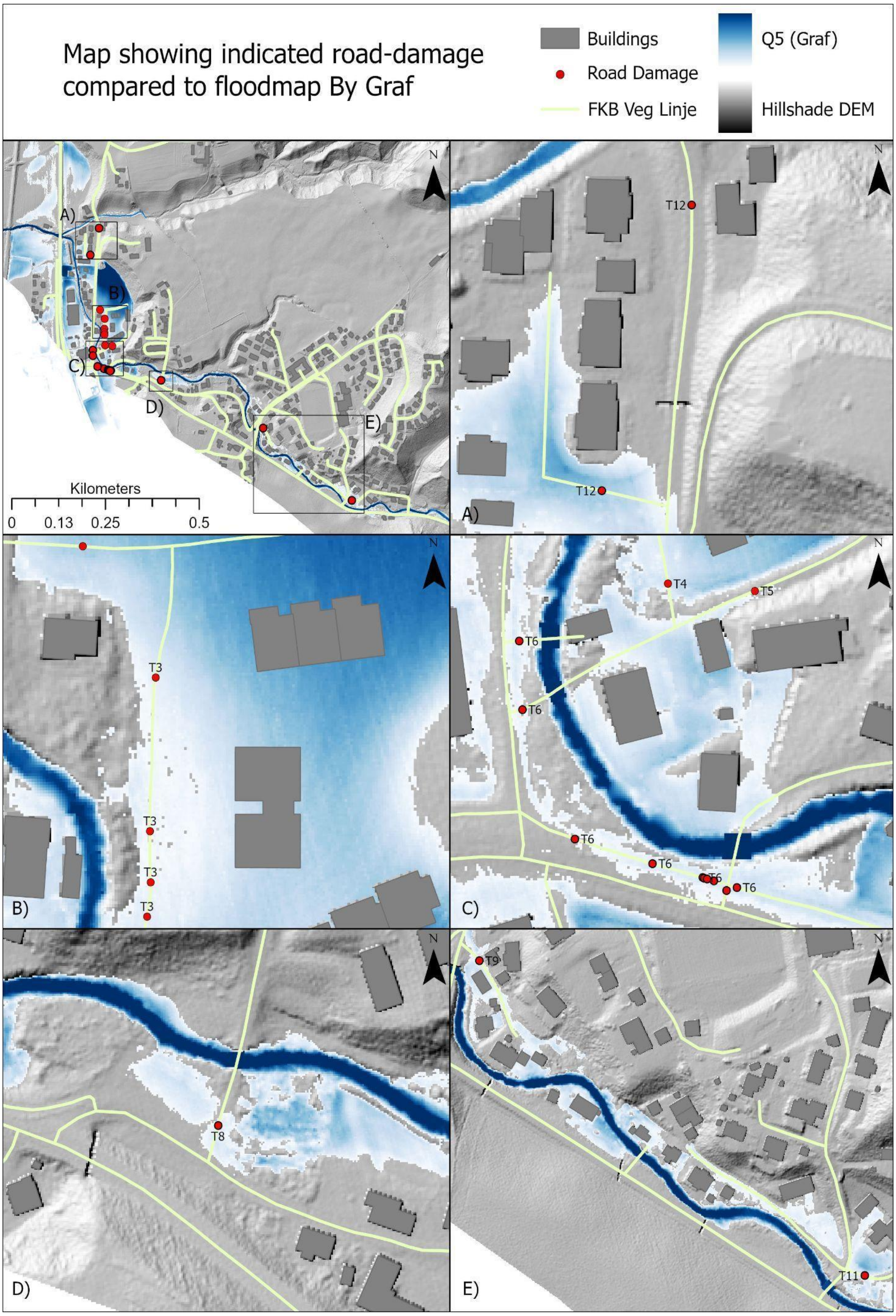
10.1; Comparison of damage caused to Buildings by critical points with flood map by Graf (2021) illustrating a flood event with 5-year recurrence interval.



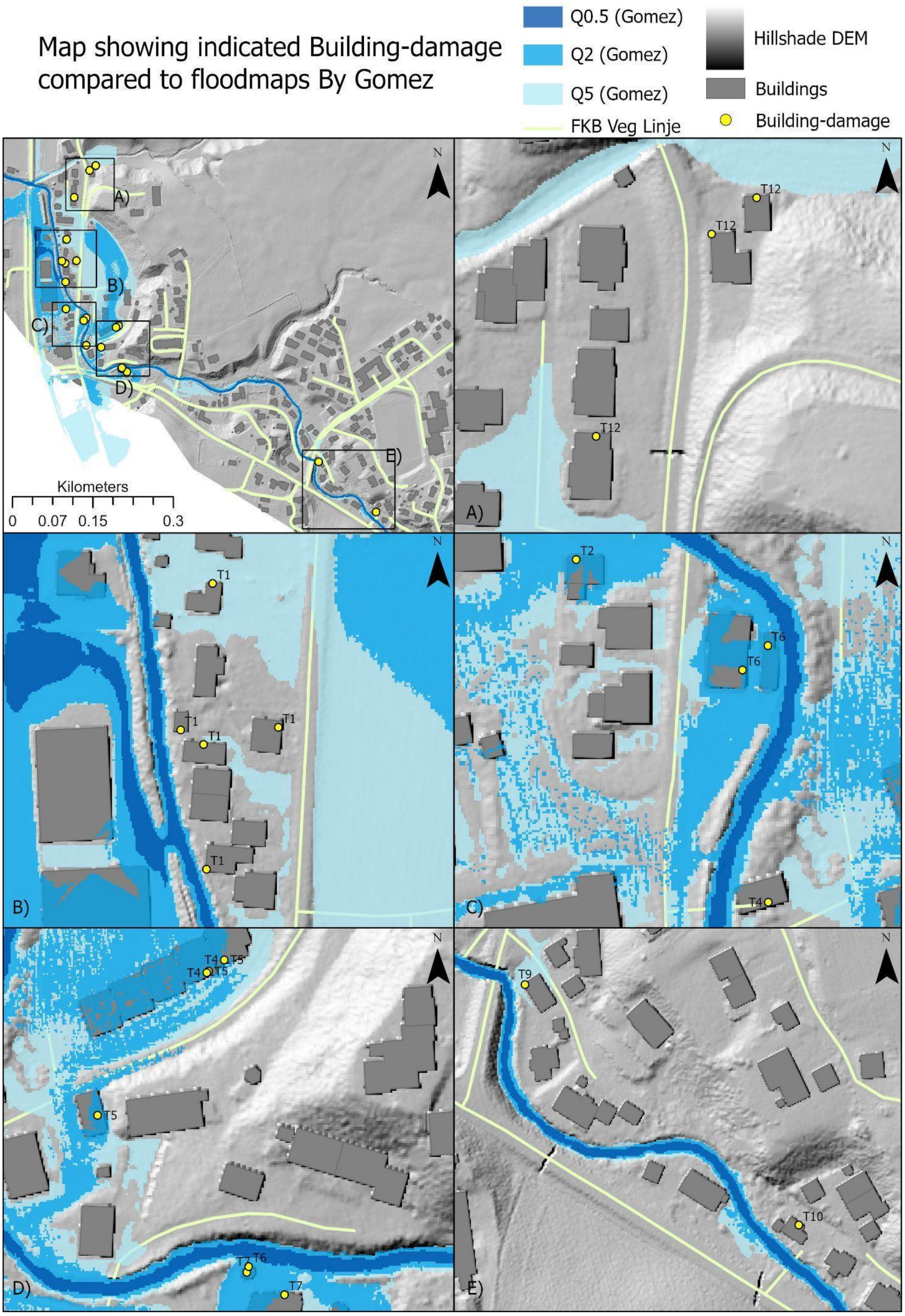
10.2; Comparison of damage caused to roads by critical points with flood map by Gomez (2021) illustrating flood events with 2 and 5-year recurrence intervals.



10.3; Comparison of damage caused to roads by critical points with flood map by Graf (2021) illustrating a flood event with 5-year recurrence interval.

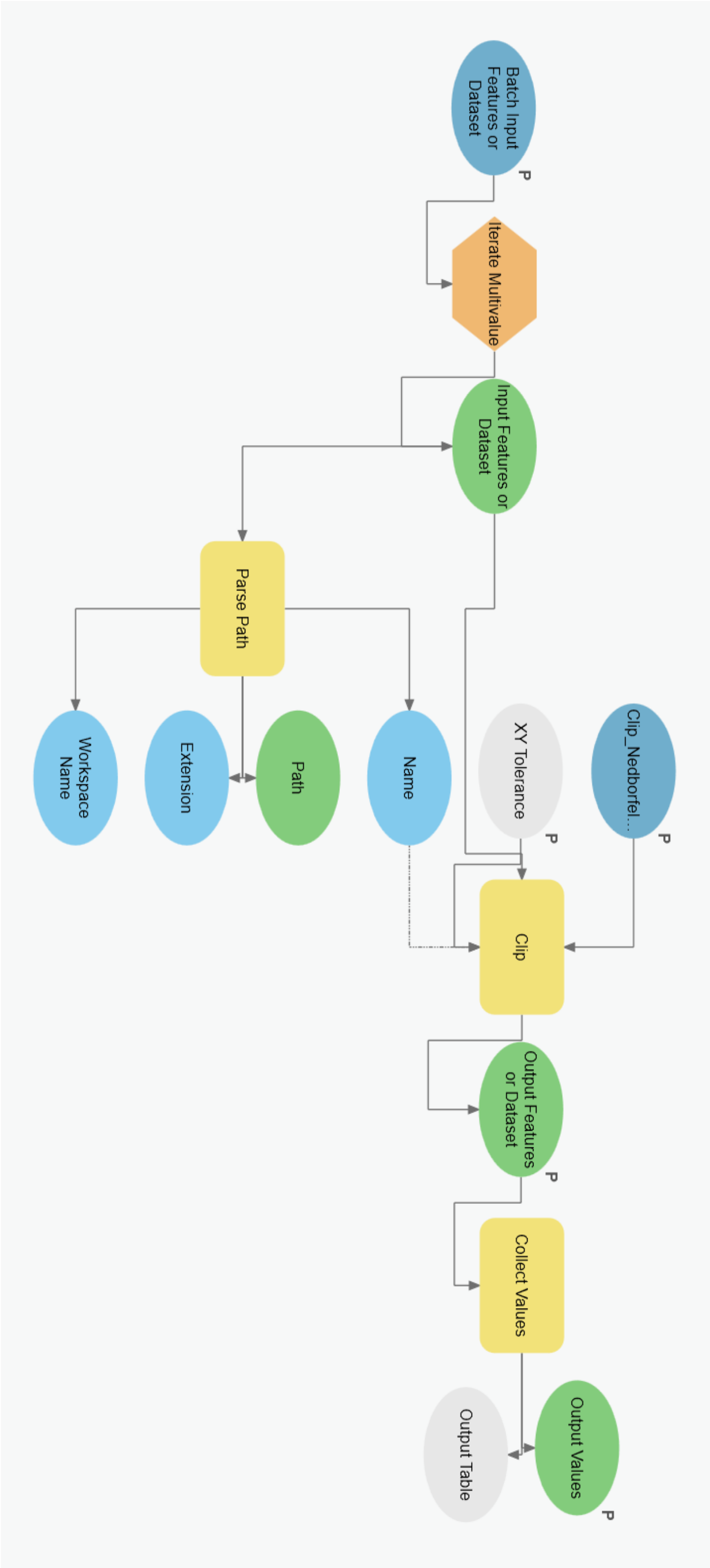


10.4; Comparison of damage caused by critical points with flood map by Graf (2021) illustrating a flood event with 5-year recurrence interval.

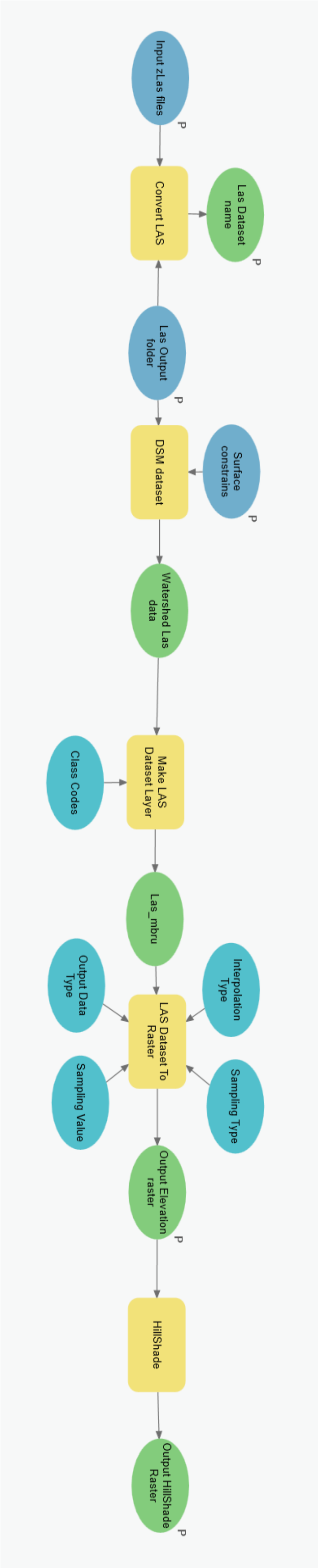


11. Appendix B

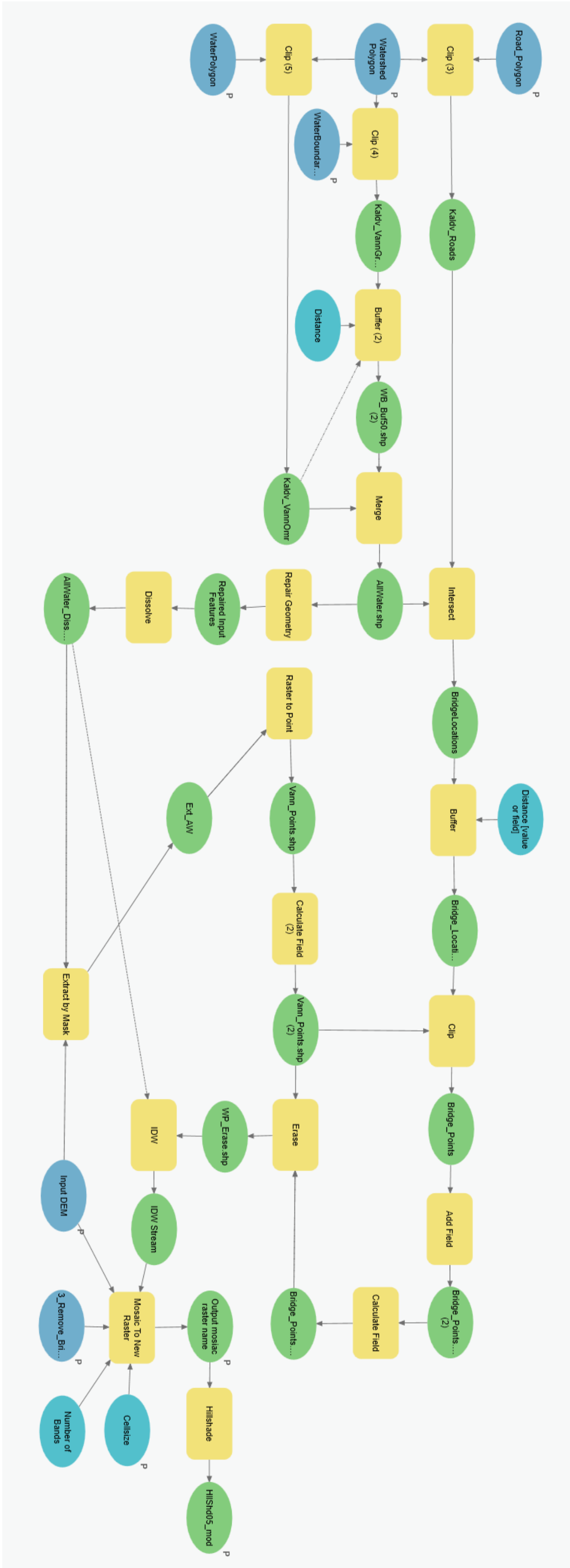
11.1. Batch clip tool



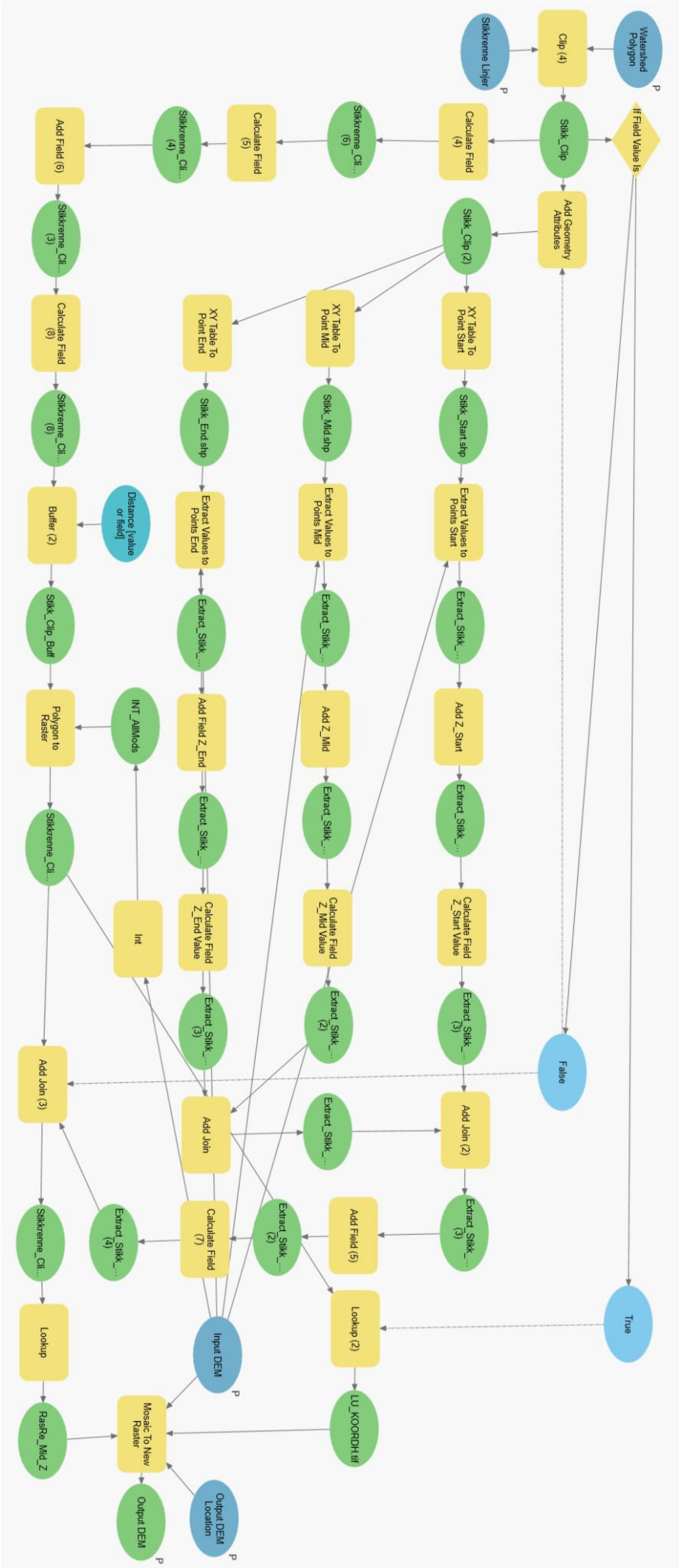
11.2. Raster from LAS



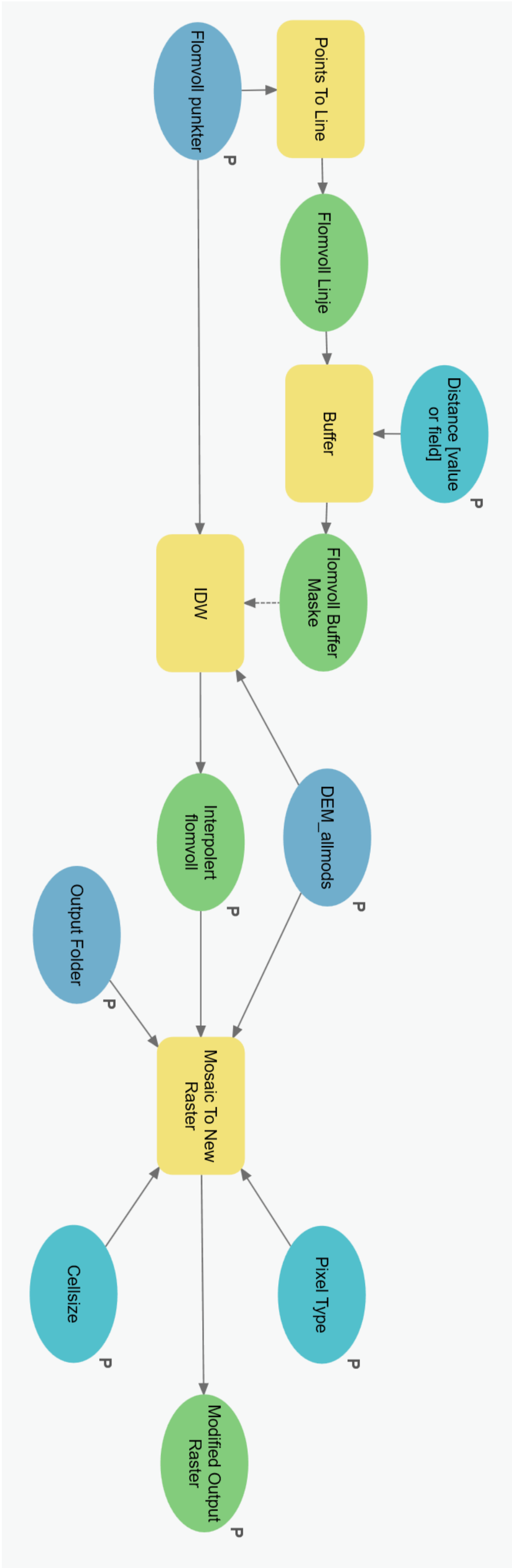
11.3. Remove bridges tool



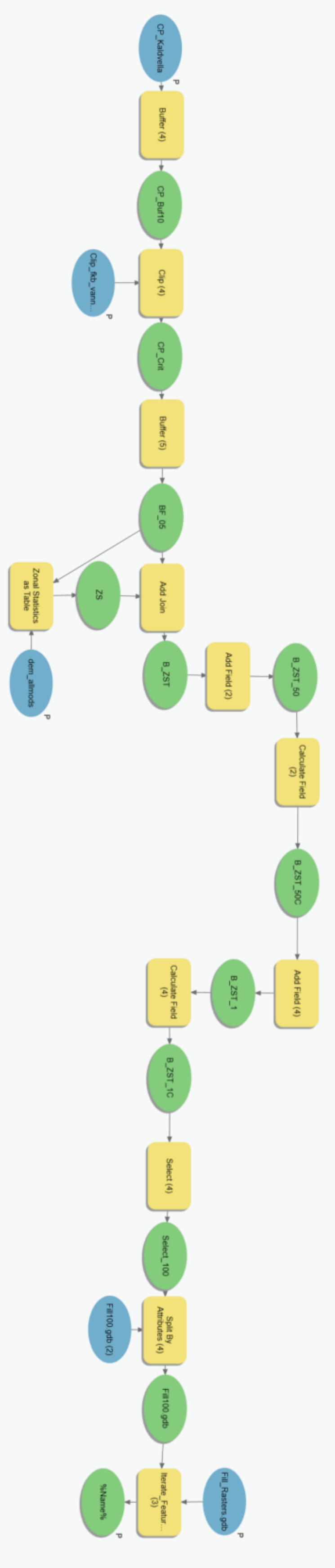
11.4. Add Culvert tool



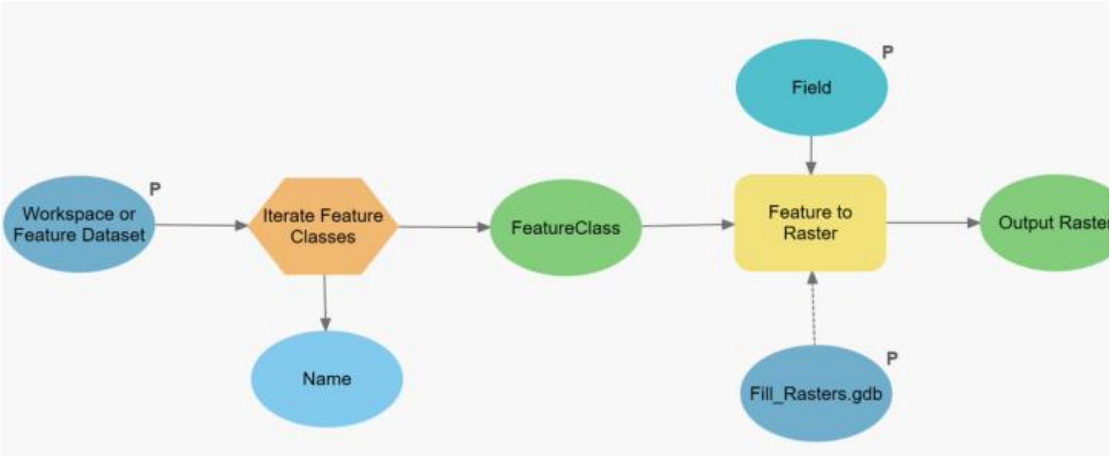
11.5 Add levee tool



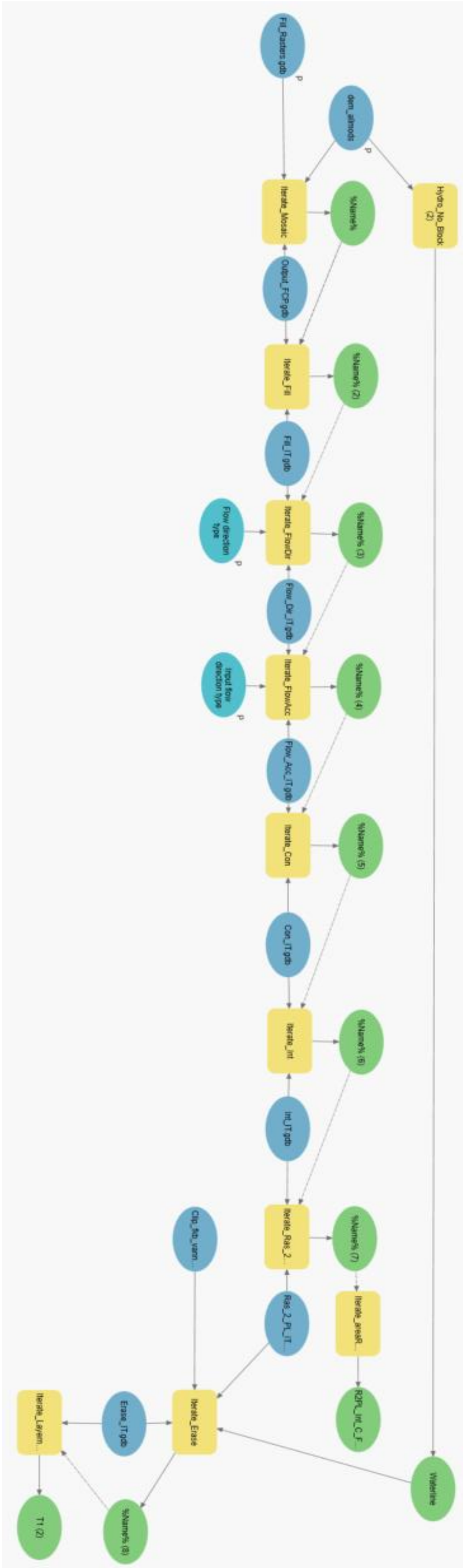
11.6 Generate Blockage



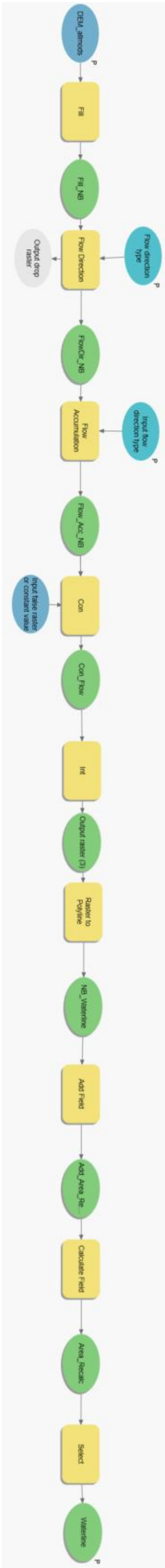
11.6.1 Sub model feature to raster



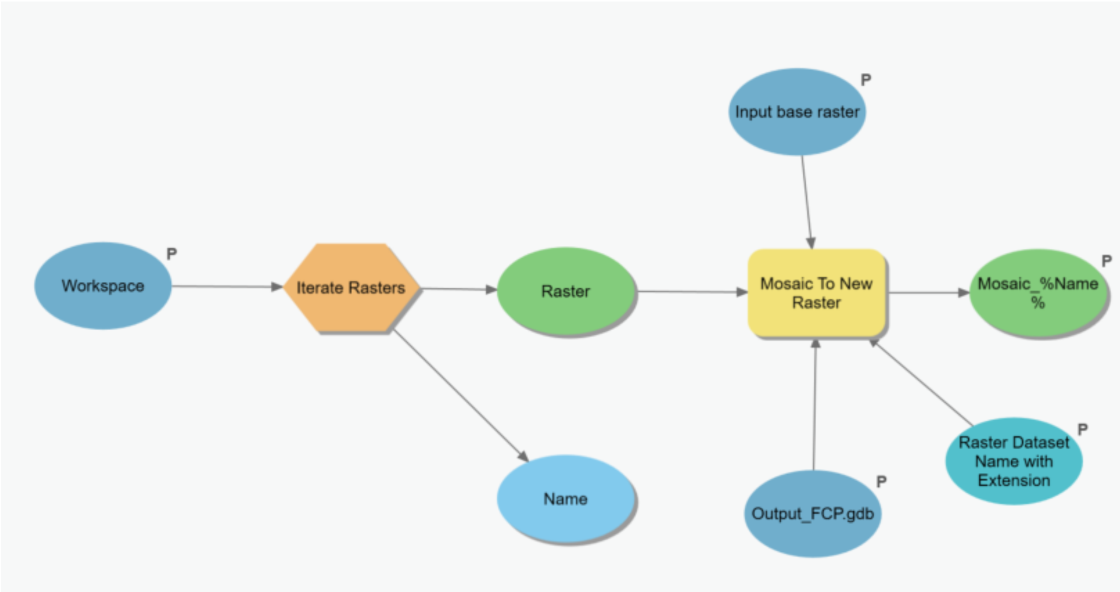
11.7 Iterate Hydro



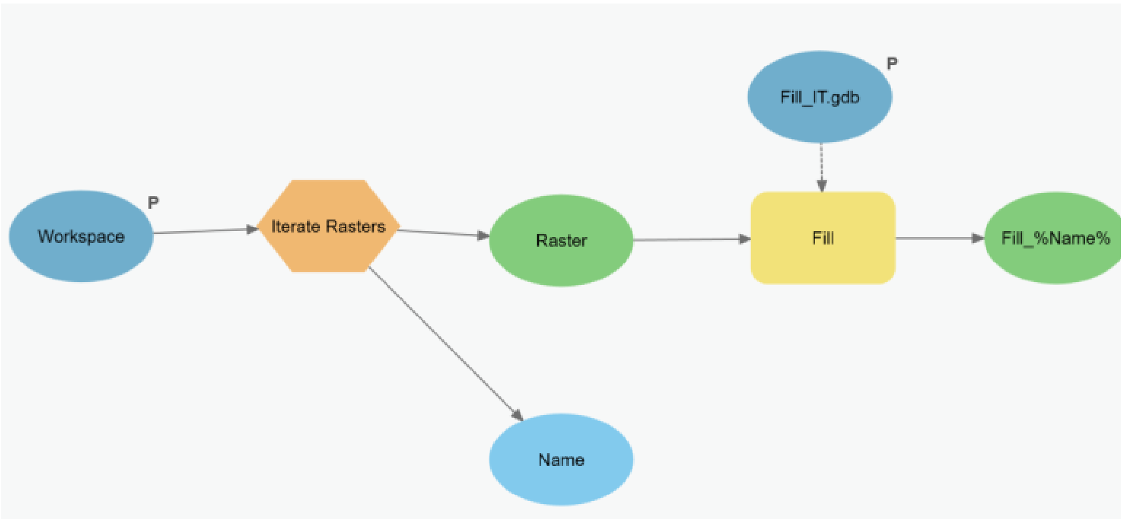
11.7.1 Submodel, Iterate_hydro_noblock



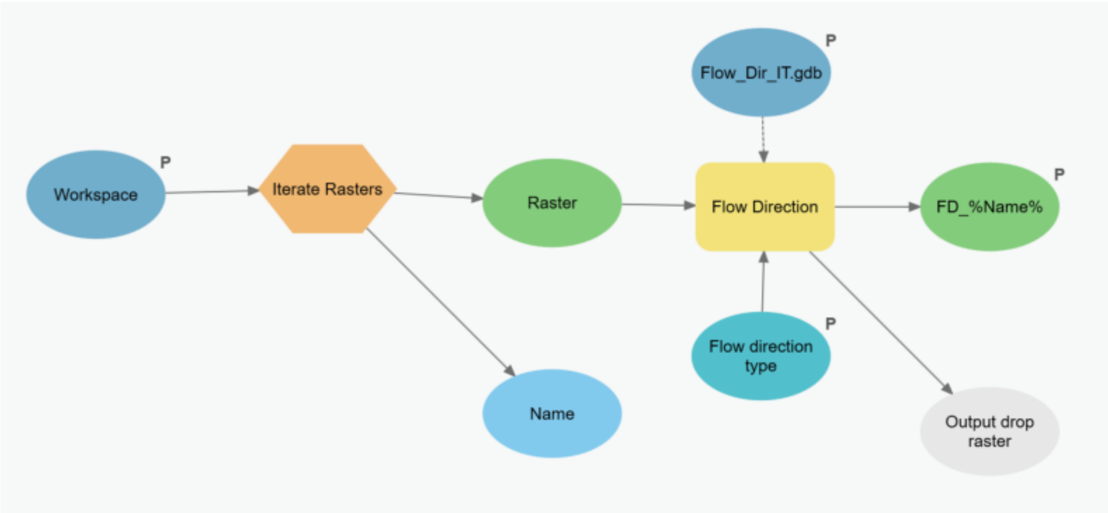
11.7.2 Submodel Iterate Mosaic to new raster



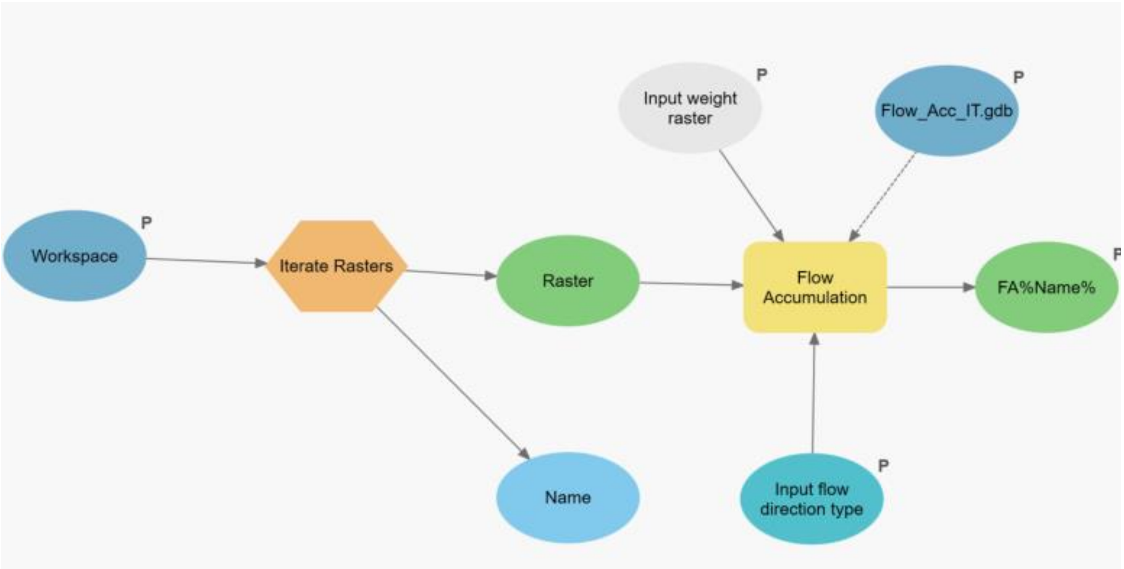
11.7.3 Submodel Iterate Fill



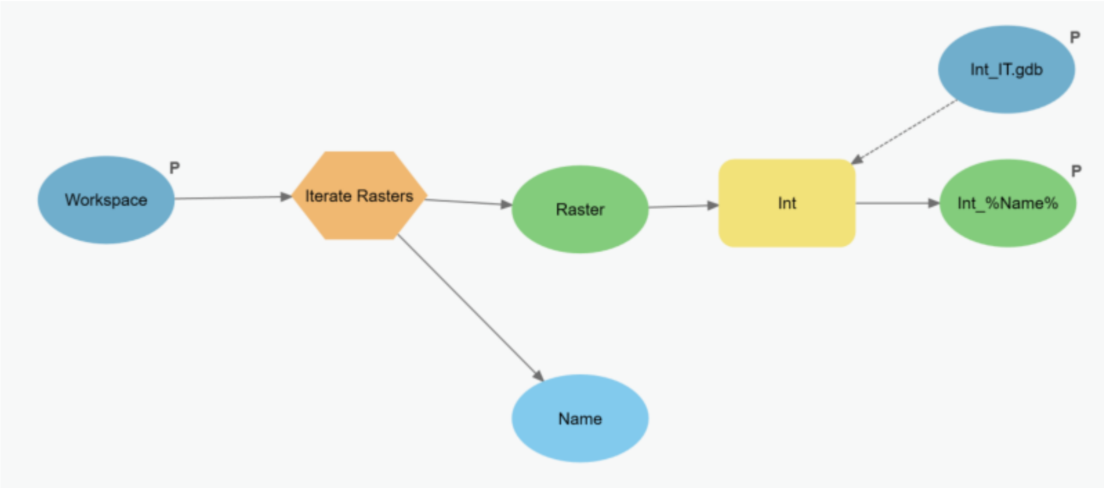
11.7.4 Submodel Iterate Flow Direction



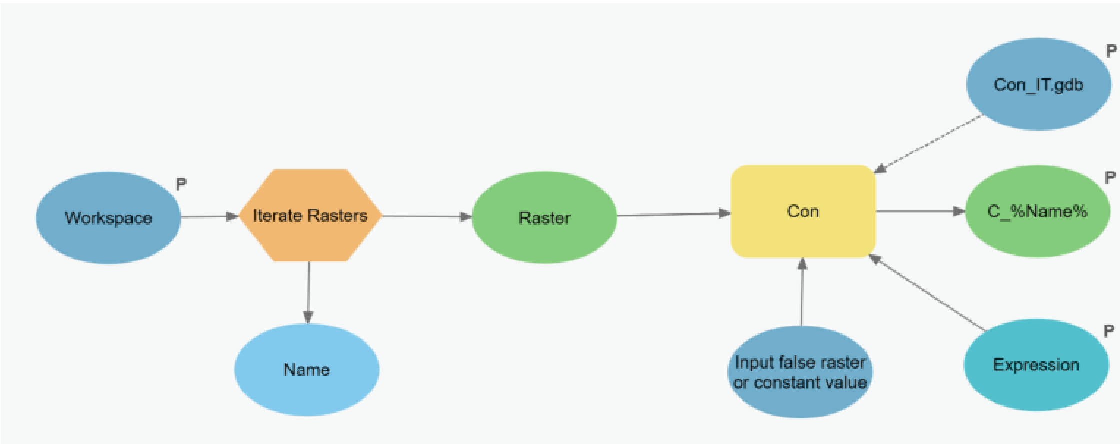
11.7.5 Submodel Iterate Flow Accumulation



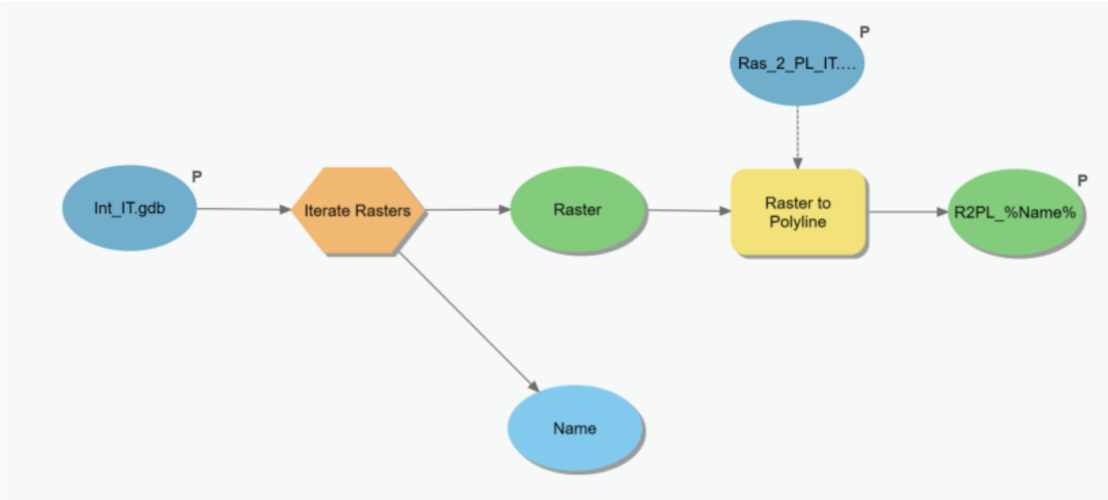
11.7.6 Submodel Iterate Integer



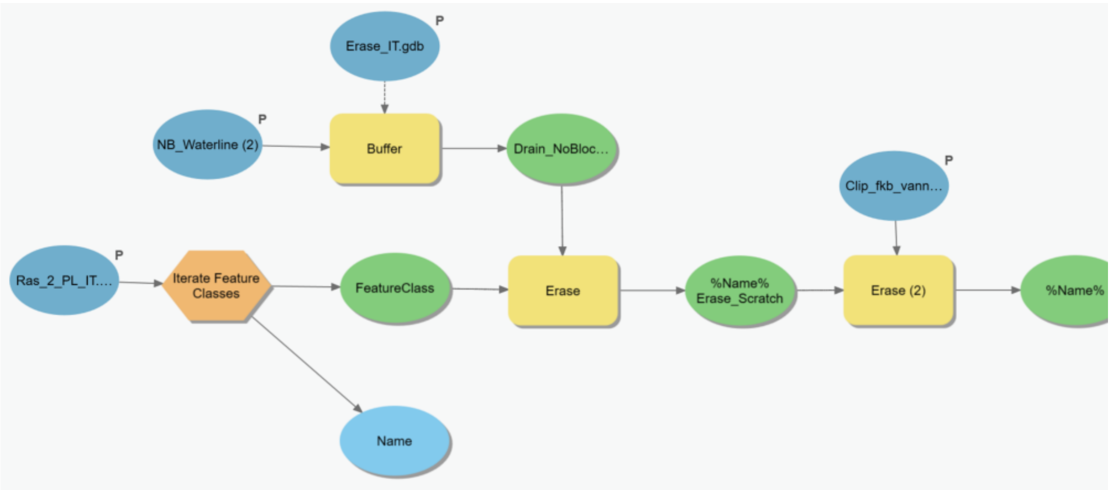
11.7.7 Submodel Iterate Con



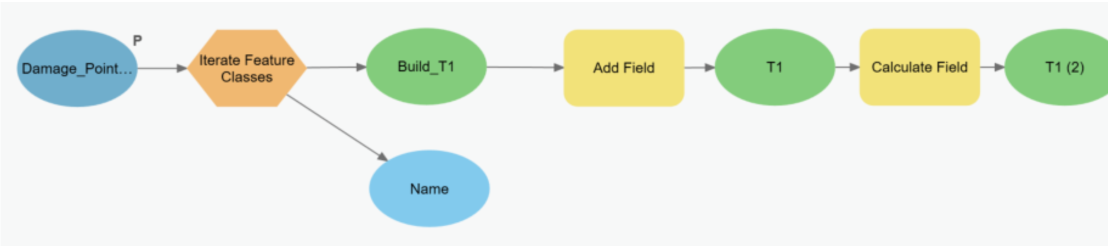
11.7.8 Submodel Iterate Raster to polyline



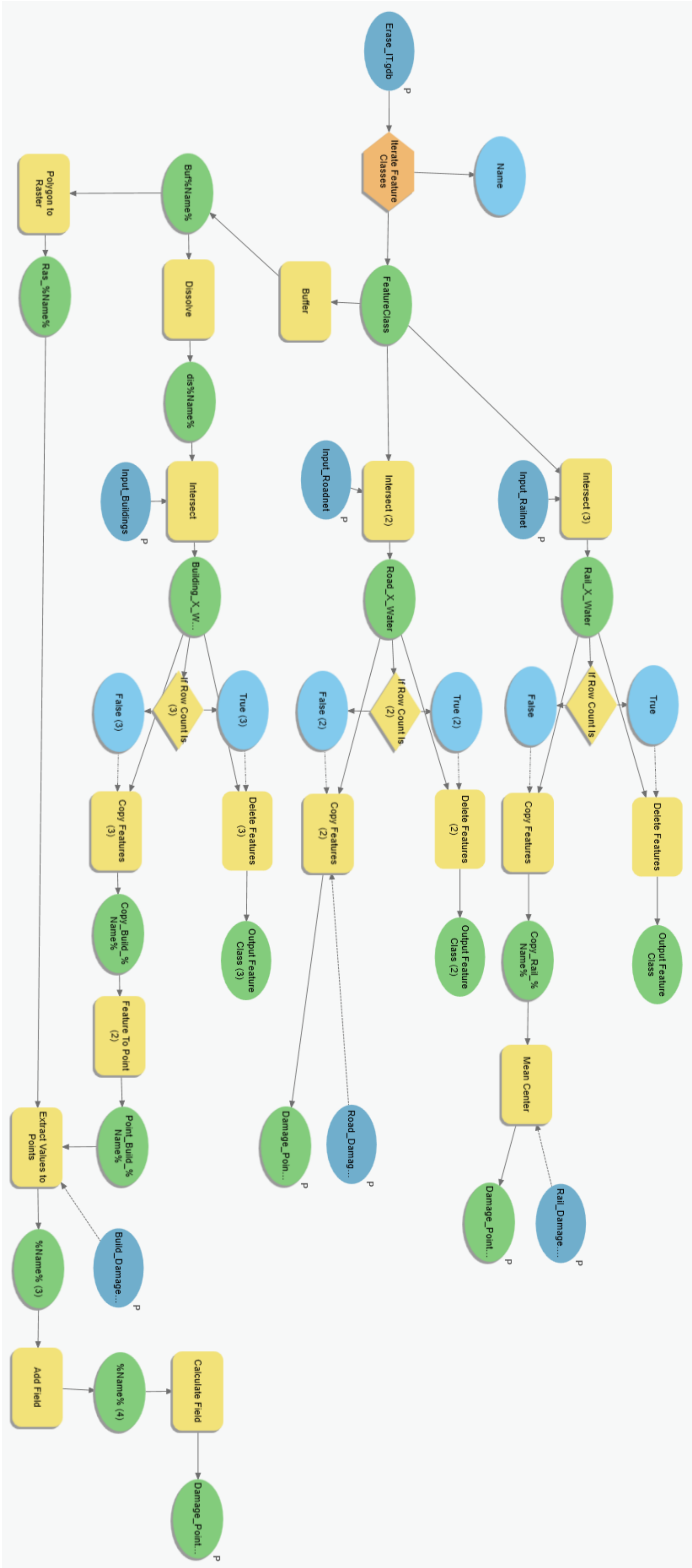
11.7.9 Submodel Iterate Erase



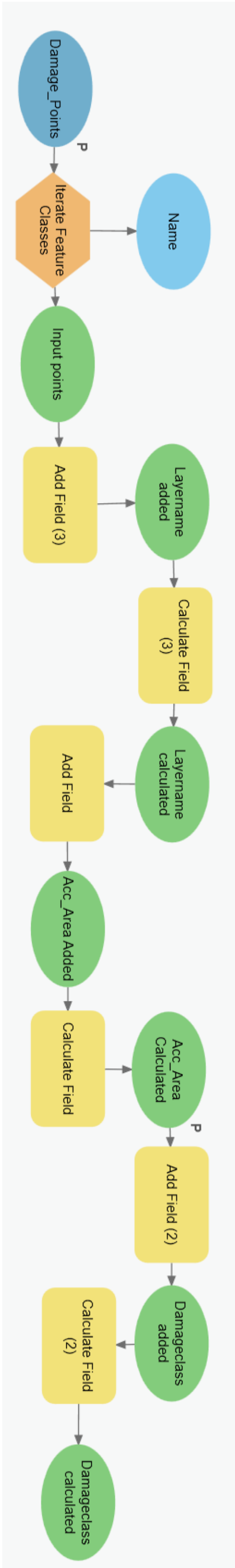
11.7.10 Submodel Iterate identifier; Layername



11.8 Iterate find damage points



11.9 Iterate calculate damage class



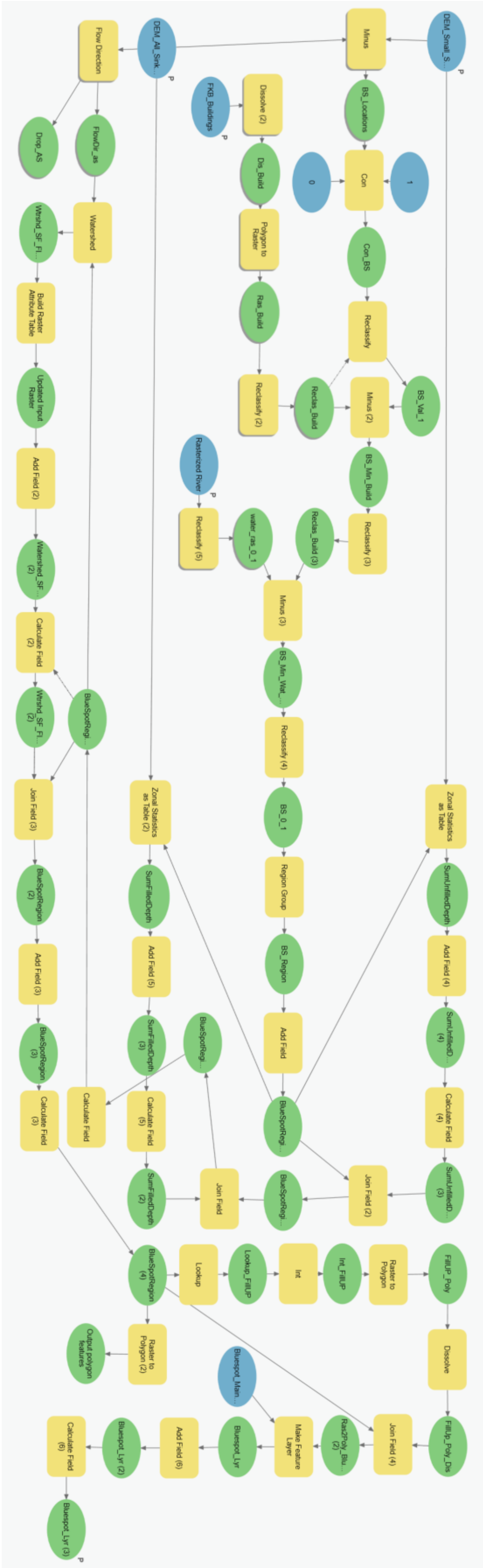
11.10 Runoff coefficient weight raster

The script assigns runoff coefficient based on the variable Artype in the AR50 areal use data. When converted to raster this can be used as input weight raster in Submodel 11.7.5; Iterate Flow Accumulation.

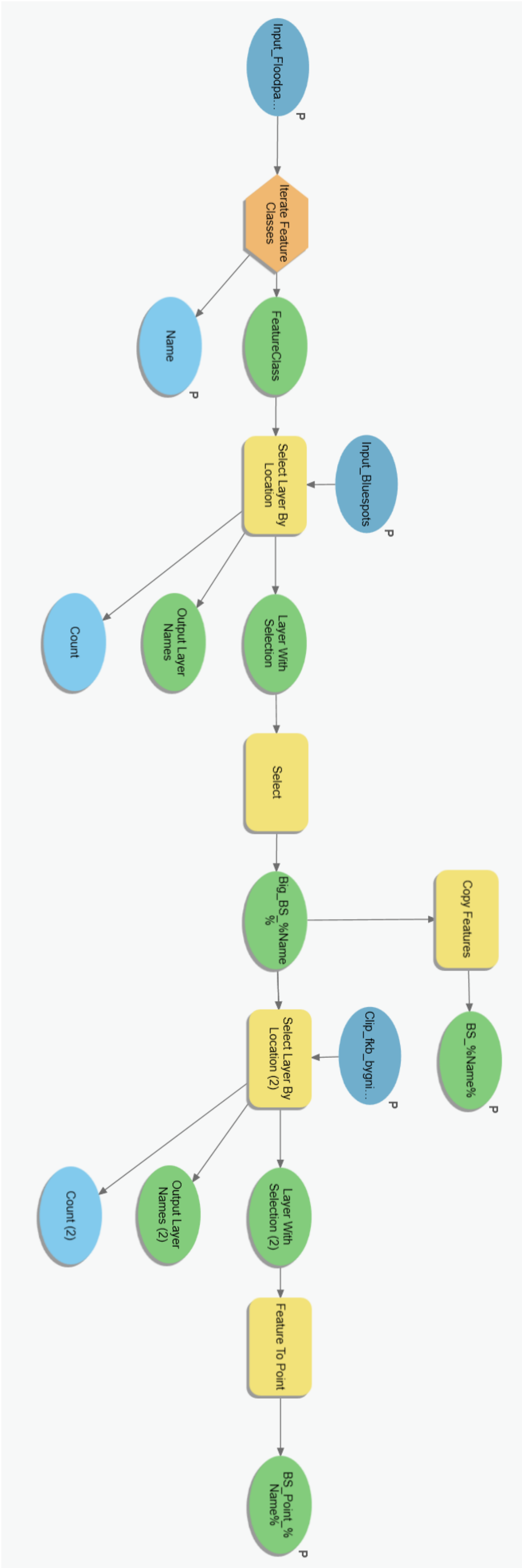
```
Avrenning(!AVRENNKO!,!ARTYPE!)

def Avrenning(Avrennko,Artype):
    if Artype== 10:
        return (0.6)
    elif Artype== 20:
        return (0.3)
    elif Artype== 30:
        return (0.3)
    elif Artype== 50:
        return (0.5)
    elif Artype== 60:
        return (0.3)
    elif Artype== 70:
        return (0.6)
    elif Artype== 81:
        return (1)
    elif Artype== 82:
        return (1)
    elif Artype== 99:
        return (0.5)
```

11.11: Bluespot_Mapping



11.12: Iterate_Bluespot_ID



12. Appendix C

12.1 Indicated building damage with open flood paths.

Indicated building damage, with open flood paths					
ORIG_FID	objtype	bygningsnu	LayerName	Acc_Area	Damage_Class
3	AnnenBygning	0	Waterline	23205100	5
25	Bygning	184537575	Waterline	23201600	5
5	AnnenBygning	0	Waterline	291250	3
15	Bygning	184580705	Waterline	517264	3
26	Bygning	184577305	Waterline	586289	3
1	AnnenBygning	0	Waterline	53435.5	2
2	AnnenBygning	0	Waterline	200444	2
4	AnnenBygning	0	Waterline	208674	2
8	Bygning	22011715	Waterline	146519	2
9	Bygning	22013238	Waterline	214574	2
10	Bygning	21990302	Waterline	135868	2
11	Bygning	184536439	Waterline	105326	2
12	Bygning	184577542	Waterline	99411.20313	2
13	Bygning	184578700	Waterline	55200	2
14	Bygning	184578166	Waterline	247768	2
16	Bygning	184578158	Waterline	248249	2
17	Bygning	184580551	Waterline	159348	2
18	Bygning	21995851	Waterline	99415.20313	2
19	Bygning	22004530	Waterline	67117.79688	2
20	Bygning	21994812	Waterline	208210	2
22	Bygning	184581191	Waterline	156902	2
23	Bygning	184537435	Waterline	58748.10156	2
24	Bygning	184536447	Waterline	106003	2
6	AnnenBygning	0	Waterline	32538.19922	1
7	AnnenBygning	0	Waterline	0	0
21	Bygning	184537427	Waterline	0	0

12.2 Indicated road-damage with open flood paths.

Indicated road damage, with open flood paths						
FID_Waterl	GATENAVN	VEGKAT	VEGNR	LayerName	Acc_Area	Damage_Cla
214460			0	Waterline	23201500	5
232288			0	Waterline	22664400	5
117425		P	98592	Waterline	6816490	5
106630		P	99856	Waterline	6627720	5
99241		P	99856	Waterline	6092260	5
97819		P	99766	Waterline	6092240	5
95773	Fremovegen	F	6594	Waterline	6085140	5
87907	Fremovegen	P	99555	Waterline	6021300	5
87596	Fremovegen	P	97355	Waterline	6020660	5
223421	Langlandsvegen	P	97935	Waterline	1456610	4
153176	Fremovegen	P	98552	Waterline	1392300	4
153064	Fremovegen	P	97338	Waterline	1392300	4
152414	Fremovegen	P	99558	Waterline	1392280	4
16793	Flåsætervegen	S	18	Waterline	1105660	4
219668	Langlandsvegen	P	97972	Waterline	1042180	4
219671	Langlandsvegen	P	97972	Waterline	1041800	4
219672	Langlandsvegen	P	97972	Waterline	1041790	4
184591	Langlandsvegen	P	98632	Waterline	693592	3
69908	Langvassvegen	S	11	Waterline	516960	3
99462		P	99856	Waterline	422480	3
121929	Rangåvegen	P	3026	Waterline	412180	3
121466	Rangåvegen	P	3026	Waterline	410628	3
121408	Rangåvegen	P	3026	Waterline	410615	3
121352	Rangåvegen	P	3026	Waterline	410609	3
188424	Fremovegen	P	99370	Waterline	387276	3
96592	Fremovegen	F	6594	Waterline	385783	3
1775	Flåsætervegen	S	18	Waterline	359993	3
38509	Langmyrvegen	P	98580	Waterline	339979	3
48831	Bjørgavegen	P	3027	Waterline	323493	3
120106	Rangåvegen	P	3026	Waterline	279447	3
156460	Fremovegen	P	98552	Waterline	244076	2
154906	Fremovegen	P	98552	Waterline	240541	2
154850	Fremovegen	P	98552	Waterline	240540	2
154507	Fremovegen	P	98552	Waterline	240530	2
152187	Fremovegen	P	98552	Waterline	240158	2
152732	Fremovegen	P	98552	Waterline	236161	2
153021	Fremovegen	P	98552	Waterline	235968	2
100696	Langvassvegen	P	97361	Waterline	228681	2
89049	Langmyrvegen	P	3012	Waterline	221244	2
83370	Langvassvegen	S	11	Waterline	207980	2
94756	Langmyrvegen	P	3012	Waterline	194210	2
48622	Langvassvegen	S	11	Waterline	184965	2
154045	Reitvegen	P	98627	Waterline	172315	2
61692	Langvassvegen	S	11	Waterline	170976	2
24368	Langvassvegen	S	11	Waterline	168179	2
87145	Langvassvegen	P	97366	Waterline	146577	2
86808	Langvassvegen	S	11	Waterline	146392	2
87834	Langvassvegen	P	97367	Waterline	145931	2
130723	Rangåvegen	P	3026	Waterline	140375	2

98567	Langvassvegen	P	97361	Waterline	134468	2
190306	Langlandsvegen	K	3018	Waterline	127180	2
47803	Flåsætervegen	S	18	Waterline	122152	2
132368	Rangåvegen	P	3026	Waterline	114872	2
160671	Flååsvegen	P	98571	Waterline	113394	2
160318	Flååsvegen	P	98571	Waterline	113056	2
160319	Flååsvegen	P	98571	Waterline	113051	2
160094	Flååsvegen	P	98571	Waterline	111979	2
160050	Flååsvegen	P	98571	Waterline	111977	2
161886	Flååsvegen	P	98571	Waterline	107428	2
106469		P	99856	Waterline	106692	2
121652	Røddebakken	P	97374	Waterline	103685	2
55804	Flåsætervegen	S	18	Waterline	103534	2
86700	Langmyrvegen	P	98580	Waterline	103344	2
156210	Fremovegen	P	98579	Waterline	101304	2
175313		P	98684	Waterline	99055	2
106050		P	99856	Waterline	98979	2
177372	Langlandsvegen	P	99584	Waterline	98420	2
77865	Langmyrvegen	P	99673	Waterline	98286	2
77404	Langmyrvegen	P	99673	Waterline	95692	2
115818	Fremovegen	P	98579	Waterline	94696	2
77200	Langmyrvegen	P	99673	Waterline	93883	2
180145	Langlandsvegen	P	99672	Waterline	92732	2
13219	Flåsætervegen	S	18	Waterline	92717	2
95405	Rangåvegen	S	32	Waterline	89419	2
175031		P	98684	Waterline	84188	2
37978	Langvassvegen	S	11	Waterline	82504	2
177999	Langlandsvegen	P	98684	Waterline	81515	2
192193	Langlandsvegen	P	98625	Waterline	79854	2
84912	Fremovegen	P	99855	Waterline	79117	2
84926	Fremovegen	P	97356	Waterline	79113	2
134597	Reitvegen	P	98636	Waterline	74933	2
134565	Reitvegen	P	98636	Waterline	74897	2
91229	Reitvegen	P	98636	Waterline	74347	2
134190	Reitvegen	P	98636	Waterline	72320	2
134149	Reitvegen	P	98636	Waterline	72318	2
133993	Reitvegen	P	98636	Waterline	72308	2
133948	Reitvegen	P	98636	Waterline	72304	2
133923	Reitvegen	P	98636	Waterline	72300	2
133872	Reitvegen	P	98636	Waterline	72291	2
133504	Reitvegen	P	98636	Waterline	71375	2
133464	Reitvegen	P	98636	Waterline	71367	2
230997	Møstadalsvegen	S	108	Waterline	69731	2
231022	Møstadalsvegen	S	108	Waterline	69731	2
231121	Møstadalsvegen	P	98118	Waterline	69725	2
131744	Rangåvegen	P	3026	Waterline	69308	2
35095	Flåsætervegen	S	18	Waterline	68387	2
71576	Langvassvegen	S	11	Waterline	67661	2
15354	Langvassvegen	S	11	Waterline	67266	2
15501	Fremovegen	F	6594	Waterline	66974	2
148110	Fremovegen	P	98579	Waterline	66682	2
147770	Fremovegen	P	98579	Waterline	66565	2
147710	Fremovegen	P	98579	Waterline	66564	2

32332	Langvassvegen	S	11	Waterline	65790	2
75152	Langvassvegen	S	11	Waterline	65674	2
75919	Langvassvegen	S	11	Waterline	65496	2
76572	Langvassvegen	S	11	Waterline	65409	2
76646	Langvassvegen	S	11	Waterline	65396	2
5437	Flåsætervegen	S	18	Waterline	65384	2
137012	Flååsvegen	K	3024	Waterline	63391	2
86164	Fremovegen	F	6594	Waterline	63139	2
17452	Flåsætervegen	S	18	Waterline	61934	2
236571	Møstadalsvegen	S	108	Waterline	61072	2
236629	Møstadalsvegen	S	108	Waterline	61069	2
236657	Møstadalsvegen	S	108	Waterline	61055	2
104239	Reitvegen	P	98636	Waterline	60895	2
22733	Langvassvegen	P	97363	Waterline	60863	2
136537	Flååsvegen	K	3024	Waterline	58882	2
136460	Flååsvegen	K	3024	Waterline	58880	2
225048		K	3004	Waterline	57785	2
224971	Lykkjvegen	K	3004	Waterline	57631	2
237983	Møstadalsvegen	P	99756	Waterline	57558	2
153411	Fremovegen	P	98552	Waterline	57183	2
153114	Fremovegen	P	97338	Waterline	56943	2
132907	Reitvegen	P	98636	Waterline	55747	2
132873	Reitvegen	P	98636	Waterline	55738	2
132677	Reitvegen	P	98636	Waterline	55720	2
132581	Reitvegen	P	98636	Waterline	55689	2
132525	Reitvegen	P	98636	Waterline	55684	2
132503	Reitvegen	P	98636	Waterline	55684	2
132088	Reitvegen	P	98636	Waterline	55579	2
132062	Reitvegen	P	98636	Waterline	55577	2
150532	Fremovegen	P	99558	Waterline	55531	2
96609	Langvassvegen	P	3013	Waterline	55515	2
120903	Reitvegen	P	98636	Waterline	54976	2
223606	Hindbergbakken	K	3005	Waterline	53901	2
223918	Hindbergbakken	K	3005	Waterline	53199	2
64468	Langmyrvegen	P	98580	Waterline	53185	2
224081	Hindbergbakken	K	3005	Waterline	53179	2
113608	Fremovegen	P	98680	Waterline	52881	2
113536	Fremovegen	P	99737	Waterline	52879	2
196579	Nyhusvegen	F	6600	Waterline	52407	2
150215	Fremovegen	P	99558	Waterline	52123	2
224632	Hindbergbakken	K	3005	Waterline	50620	2
113095	Fremovegen	P	99737	Waterline	49951	1
113018	Fremovegen	P	99737	Waterline	49938	1
112950	Fremovegen	P	99737	Waterline	49931	1
226019	Hindbergbakken	K	3005	Waterline	49928	1
243850	Møstadalsvegen	P	99756	Waterline	49669	1
21729	Flåsætervegen	S	18	Waterline	49568	1
147682	Fremovegen	P	98579	Waterline	49435	1
147562	Fremovegen	P	98579	Waterline	49431	1
158105	Flååsvegen	P	98571	Waterline	48682	1
137192	Rangåvegen	P	96784	Waterline	48315	1
150137	Fremovegen	P	99558	Waterline	47454	1
117886	Fremovegen	P	98579	Waterline	47317	1

42150	Langmyrvegen	P	98580	Waterline	45208	1
23783	Fremovegen	F	6594	Waterline	44903	1
33929	Flåsætervegen	S	18	Waterline	44561	1
33894	Flåsætervegen	S	18	Waterline	44560	1
104815	Fremovegen	F	6594	Waterline	43239	1
96827	Langvassvegen	P	3013	Waterline	42092	1
114555	Fremovegen	F	6594	Waterline	41624	1
137599	Fremovegen	P	98579	Waterline	41037	1
32729	Flåsætervegen	S	18	Waterline	40471	1
32624	Flåsætervegen	S	18	Waterline	40427	1
32608	Flåsætervegen	S	18	Waterline	40390	1
32593	Flåsætervegen	S	18	Waterline	40390	1
32563	Flåsætervegen	S	18	Waterline	40377	1
32544	Flåsætervegen	S	18	Waterline	40376	1
111576	Fremovegen	P	99737	Waterline	39725	1
58258	Fremovegen	P	97353	Waterline	38669	1
57719	Fremovegen	P	97349	Waterline	38233	1
21120	Langvassvegen	P	97363	Waterline	38035	1
116913	Rangåvegen	S	32	Waterline	37830	1
135886	Fremovegen	P	98579	Waterline	37302	1
135240	Fremovegen	P	98579	Waterline	37221	1
139490	Reitvegen	S	48	Waterline	37059	1
181028	Langlandsvegen	K	3018	Waterline	36204	1
167576	Flååsvegen	P	98571	Waterline	36188	1
37153	Langvassvegen	S	11	Waterline	35899	1
143106	Reitvegen	P	97397	Waterline	35689	1
172599	Fremovegen	P	96723	Waterline	34672	1
72577	Fremovegen	F	6594	Waterline	34454	1
132241	Fremovegen	F	6594	Waterline	33461	1
17210	Langvassvegen	S	11	Waterline	33306	1
166622		P	96472	Waterline	32894	1
29665	Fremovegen	F	6594	Waterline	32244	1
118296	Rangåvegen	S	32	Waterline	31555	1
124299	Fremovegen	P	98579	Waterline	31488	1
51650	Bjørgavegen	P	3027	Waterline	31311	1
116414	Fremovegen	P	98579	Waterline	31261	1
116536	Fremovegen	P	98579	Waterline	31022	1
116588	Fremovegen	P	98579	Waterline	31021	1
161392	Flååsvegen	P	98571	Waterline	29725	1
204696	Fremovegen	P	98579	Waterline	29720	1
51769	Langmyrvegen	P	98580	Waterline	29396	1
51663	Langmyrvegen	P	98580	Waterline	29342	1
51596	Langmyrvegen	P	98580	Waterline	29339	1
122394		P	98592	Waterline	29160	1
51309	Bjørgavegen	P	3027	Waterline	28590	1
51310	Bjørgavegen	P	3027	Waterline	28589	1
57195	Fremovegen	F	6594	Waterline	28586	1
140714	Fremovegen	P	98579	Waterline	28230	1
51277	Flåsætervegen	S	18	Waterline	27310	1
17135	Langvassvegen	S	11	Waterline	25212	1
260545	Møstadalsvegen	P	99306	Waterline	25108	1

13. Appendix D

13.1 Data overview

Data overview		
Data	**Source**	**Comment**
LiDAR elevation data (Høydedata)	https://hoydedata.no/LaserInnsyn/	"Velg kommune [Klikk i kart]"--> Melhus
FKB vann	https://kartkatalog.geonorge.no/metadata/fkb-vann/595e47d9-d201-479c-a77d-cbc1f573a76b	Geografisk område: “Melhus” Projeksjon: “EUREF89 UTM zone 32, 2d + NN2000” Format: FGDB 10.0
FKB Veg	https://kartkatalog.geonorge.no/metadata/fkb-veg/4920b452-75cc-45f2-964c-3378204c3517	Geografisk område: “Melhus” Projeksjon: “EUREF89 UTM zone 32, 2d + NN2000” Format: FGDB 10.0
Areal Ressurs kart	https://kartkatalog.geonorge.no/metadata/arealressurskart-ar50-arealtyper/41f6b000-c394-41c5-8ebb-07a0a3ec914f	Geografisk område: “Melhus” Projeksjon: “EUREF89 UTM zone 32, 2d + NN2000” Format: FGDB 10.0
-Nedbørsfelt/ REGINE-enhet	https://nedlasting.nve.no/gis/	REGINE enhet | Esri Shapefile EUREF89 (UTM) | UTM-sone 32 “Melhus”
N20 Bygninger	https://kartkatalog.geonorge.no/metadata/n20-bygning/b187c449-04ac-42fe-9eab-83d81bf338bc	Geografisk område: “Melhus” Projeksjon: “EUREF89 UTM zone 32, 2d + NN2000” | Format: FGDB 10.0
FKB Bane	https://kartkatalog.geonorge.no/metadata/fkb-bane/3165138f-1461-44fe-8b10-eac44e08a10a	Geografisk område: “Melhus” Projeksjon: “EUREF89 UTM zone 32, 2d + NN2000” Format: FGDB 10.0
Løsmassekart	http://geo.ngu.no/download/index.jsp	Format: esri shp Projeksjon: UTM zone 32, Euref89
Maringrense	http://geo.ngu.no/download/index.jsp	esri shp | UTM sone 32, Euref89 (Mulig indikasjon på kvikkleire områder)
Stikkrenne/Kulvert	https://vegkart.atlas.vegvesen.no/#kartlag:geodata/@263137,7015251,9/hvor:(kommune~!5028)~	Vegobjtype:"Stikkrenner/Kulvert"|Søkeområde: "Melhus" SOSI | konvertert til shp med SOSI-Shape v 3.2
Bruer	https://vegkart.atlas.vegvesen.no/#kartlag:geodata/@263137,7015251,9/hva:!(id~60)~/hvor:(kommune~!5028)~	Vegobjtype: "Bruer" | Søkeområde: "Melhus" SOSI | konvertert til shp med SOSI-Shape v 3.2

14. Appendix E

14.1: 1_Batch_Clip_Tool

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-02 20:09:44
"""
import arcpy
from sys import argv
def #  NOT  IMPLEMENTED# Function Body not implemented

def BatchClip1(Batch_Input_Features_or_Dataset, Clip_Feature,
Output_Features_or_Dataset=fr"Kaldvella.gdb\Clip_OutFeatureClass_{Name}",
XY_Tolerance="0 Meters"):  # 1_Batch_Clip_Tool

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False


    for Input_Features_or_Dataset in #  NOT
IMPLEMENTED(Batch_Input_Features_or_Dataset):

        # Process: Parse Path (Parse Path) (mb)
        Path, Name, Extension, Workspace_Name =
arcpy.mb.ParsePathExt(in_data_element=Input_Features_or_Dataset, format="FORMAT")

        # Process: Clip (Clip) (analysis)
        if Name:
            arcpy.analysis.Clip(in_features=Input_Features_or_Dataset, clip_features=Clip_Feature,
out_feature_class=Output_Features_or_Dataset, cluster_tolerance=XY_Tolerance)

        # Process: Collect_Values (Collect Values)
        # Collect Values Utility is not implemented

        return Output_Values

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"scrathWorkspace.gdb", workspace=r"Workspace.gdb"):
        BatchClip1(*argv[1:])
```

14.2: 2_Raster_From_LAS

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-04 19:40:03
"""
import arcpy
from sys import argv

def Raster_From_LAS(Input_zLas_files="%Name%", Las_Output_folder="%Name%",
Las_Dataset_name="%Name%.lasd", Sampling_Type="CELLSIZE",
Output_Elevation_raster=f"{arcpy.env.workspace},%Name%",
Output_HillShade_Raster="%Name%"):  # Raster_From_LAS

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("3D")
    arcpy.CheckOutExtension("spatial")

    Class_Codes = ["2", "6", "9", "10", "11", "17"]
    Interpolation_Type = "BINNING IDW LINEAR"
    Output_Data_Type = "FLOAT"
    Sampling_Value = 1

    # Process: Convert LAS (Convert LAS) (conversion)
    arcpy.conversion.ConvertLas(in_las=Input_zLas_files, target_folder=Las_Output_folder,
file_version="SAME_AS_INPUT", point_format="", compression="NO_COMPRESSION",
las_options=["REARRANGE_POINTS", "REMOVE_EXTRA_BYTES"],
out_las_dataset=Las_Dataset_name, define_coordinate_system="ALL_FILES",
in_coordinate_system="PROJCS[\"ETRS_1989_UTM_Zone_32N\",GEOGCS[\"GCS_ETRS_198
9\",DATUM[\"D_ETRS_1989\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101]],PRIMEM[\
"Greenwich\",0.0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Mercator
\"],PARAMETER[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],PARAMETE
R[\"Central_Meridian\",9.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_
Origin\",0.0],UNIT[\"Meter\",1.0]],VERTCS[\"NN2000_height\",VDATUM[\"Norway_Normal_Null_2
000\"],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"Meter\",1.0]]")

    # Process: DEM dataset (Create LAS Dataset) (management)
    Watershed_Las_data = f"{arcpy.env.workspace},%Name%.lasd"
    arcpy.management.CreateLasDataset(input=[Las_Output_folder],
out_las_dataset=Watershed_Las_data, folder_recursion="NO_RECURSION",
in_surface_constraints=[],
spatial_reference="PROJCS[\"ETRS_1989_UTM_Zone_32N\",GEOGCS[\"GCS_ETRS_1989\",D
ATUM[\"D_ETRS_1989\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101]],PRIMEM[\"Gre
enwich\",0.0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Mercator\"],P
ARAMETER[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],PARAMETER[\"
Central_Meridian\",9.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_Origi
n\",0.0],UNIT[\"Meter\",1.0]],VERTCS[\"NN2000 height\",VDATUM[\"Norway Normal Null
2000\"],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"metre\",1.0]]",
compute_stats="COMPUTE_STATS", relative_paths="ABSOLUTE_PATHS",
create_las_prj="NO_FILES")

    # Process: Make LAS Dataset Layer (Make LAS Dataset Layer) (management)
    Las_dataset = f"{arcpy.env.workspace},%Name%"
    arcpy.management.MakeLasDatasetLayer(in_las_dataset=Watershed_Las_data,
out_layer=Las_dataset, class_code=Class_Codes, return_values=["Last Return", "Single
```

```
Return", "First of Many", "Last of Many", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12",
"13", "14", "15"], no_flag="INCLUDE_UNFLAGGED", synthetic="INCLUDE_SYNTHETIC",
keypoint="INCLUDE_KEYPOINT", withheld="EXCLUDE_WITHHELD", surface_constraints=[],
overlap="INCLUDE_OVERLAP")

    # Process: LAS Dataset To Raster (LAS Dataset To Raster) (conversion)
    arcpy.conversion.LasDatasetToRaster(in_las_dataset=Las_dataset,
out_raster=Output_Elevation_raster, value_field="ELEVATION",
interpolation_type=Interpolation_Type, data_type=Output_Data_Type,
sampling_type=Sampling_Type, sampling_value=Sampling_Value, z_factor=1)
    Output_Elevation_raster = arcpy.Raster(Output_Elevation_raster)

    # Process: HillShade (HillShade) (3d)
    arcpy.ddd.HillShade(in_raster=Output_Elevation_raster, out_raster=Output_HillShade_Raster,
azimuth=315, altitude=45, model_shadows="NO_SHADOWS", z_factor=1)
    Output_HillShade_Raster = arcpy.Raster(Output_HillShade_Raster)

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"%scratchWorkspace%", workspace=r"%Workspace%"):
        Raster_From_LAS(*argv[1:])
```

14.3: 3_Remove_Bridges_Tool

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-02 14:46:51
"""
import arcpy
from sys import argv

def RemoveBridgesTool(WaterBoundaryPL="FKB_Vanngrense",
WaterPolygon="FKB_Vannområde", Road_Polygon="FKB_vegområde",
Watershed_Polygon="Nedborsfelt:RegineEnhet", Input_DEM="DEM_05",
HllShd05_mod="HS_05", _3_Remove_Bridges="", Cellsize=0):  # 3_Remove_Bridges_Tool

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("spatial")
    arcpy.CheckOutExtension("3D")

    # Model Environment settings
    with arcpy.EnvManager(scratchWorkspace=r"%scratchWorkspace%",
workspace=r"Workspace"):
        Distance = "25 Centimeters"
        Distance_value_or_field_ = "15 Meters"
        Number_of_Bands = 1

        # Process: Clip (4) (Clip) (analysis)
        Kaldv_VannGrense = "%Name%"
        arcpy.analysis.Clip(in_features=WaterBoundaryPL, clip_features=Watershed_Polygon,
out_feature_class=Kaldv_VannGrense, cluster_tolerance="")

        # Process: Clip (5) (Clip) (analysis)
        Kaldv_VannOmr = "C:Kaldv_VannOmr"
        arcpy.analysis.Clip(in_features=WaterPolygon, clip_features=Watershed_Polygon,
out_feature_class=Kaldv_VannOmr, cluster_tolerance="")

        # Process: Buffer (2) (Buffer) (analysis)
        WB_Buf50_shp_2_ = "WB_Buf50.shp"
        if Kaldv_VannOmr:
            arcpy.analysis.Buffer(in_features=Kaldv_VannGrense,
out_feature_class=WB_Buf50_shp_2_, buffer_distance_or_field=Distance, line_side="FULL",
line_end_type="ROUND", dissolve_option="ALL", dissolve_field=[], method="PLANAR")

        # Process: Merge (Merge) (management)
        AllWater_shp = "AllWater.shp"
        if Kaldv_VannOmr and WB_Buf50_shp_2_:
            arcpy.management.Merge(inputs=[WB_Buf50_shp_2_, Kaldv_VannOmr],
output=AllWater_shp, field_mappings="datafangstdato \"datafangstdato\" true true false 8 Date 0
0,First,#,C:"Outputfields" #Output fields

        # Process: Repair Geometry (Repair Geometry) (management)
        if Kaldv_VannOmr and WB_Buf50_shp_2_:
            Repaired_Input_Features =
arcpy.management.RepairGeometry(in_features=AllWater_shp, delete_null="DELETE_NULL",
validation_method="ESRI")[0]
```

```
    # Process: Dissolve (Dissolve) (management)
    AllWater_Diss_shp = "AllWater_Diss.shp"
    if Kaldv_VannOmr and WB_Buf50_shp_2_:
        arcpy.management.Dissolve(in_features=Repaired_Input_Features,
out_feature_class=AllWater_Diss_shp, dissolve_field=[], statistics_fields=[],
multi_part="SINGLE_PART", unsplit_lines="DISSOLVE_LINES")

    # Process: Extract by Mask (Extract by Mask) (sa)
    Ext_AW = "Ext_AW"
    Extract_by_Mask = Ext_AW
    if AllWater_Diss_shp and Kaldv_VannOmr and WB_Buf50_shp_2_:
        with arcpy.EnvManager(cellSize="MINOF"):
            Ext_AW = arcpy.sa.ExtractByMask(in_raster=Input_DEM,
in_mask_data=AllWater_Diss_shp)
            Ext_AW.save(Extract_by_Mask)


    # Process: Raster to Point (Raster to Point) (conversion)
    Vann_Points_shp = "Vann_Points.shp"
    if AllWater_Diss_shp and Ext_AW and Kaldv_VannOmr and WB_Buf50_shp_2_:
        arcpy.conversion.RasterToPoint(in_raster=Ext_AW,
out_point_features=Vann_Points_shp, raster_field="Value")

    # Process: Calculate Field (2) (Calculate Field) (management)
    if AllWater_Diss_shp and Ext_AW and Kaldv_VannOmr and WB_Buf50_shp_2_:
        Vann_Points_shp_2_ = arcpy.management.CalculateField(in_table=Vann_Points_shp,
field="grid_code", expression="!grid_code! - 0,287", expression_type="PYTHON3",
code_block="", field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Clip (3) (Clip) (analysis)
    Kaldv_Roads = "Kaldv_Roads"
    arcpy.analysis.Clip(in_features=Road_Polygon, clip_features=Watershed_Polygon,
out_feature_class=Kaldv_Roads, cluster_tolerance="")

    # Process: Intersect (Intersect) (analysis)
    BridgeLocations = "BridgeLocations"
    if Kaldv_VannOmr and WB_Buf50_shp_2_:
        arcpy.analysis.Intersect(in_features=[[AllWater_shp, ""], [Kaldv_Roads, ""]],
out_feature_class=BridgeLocations, join_attributes="ALL", cluster_tolerance="",
output_type="INPUT")

    # Process: Buffer (Buffer) (analysis)
    Bridge_Locations = "Bridge_Buff_20.shp"
    if Kaldv_VannOmr and WB_Buf50_shp_2_:
        arcpy.analysis.Buffer(in_features=BridgeLocations, out_feature_class=Bridge_Locations,
buffer_distance_or_field=Distance_value_or_field_, line_side="FULL", line_end_type="ROUND",
dissolve_option="NONE", dissolve_field=[], method="PLANAR")

    # Process: Clip (Clip) (analysis)
    Bridge_Points = "Bridge_Points"
    if AllWater_Diss_shp and Ext_AW and Kaldv_VannOmr and WB_Buf50_shp_2_:
        with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
```

```
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000_height",VDATUM["Norway_Normal_
Null_2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["Meter",1.0]]")
:
            arcpy.analysis.Clip(in_features=Vann_Points_shp_2_, clip_features=Bridge_Locations,
out_feature_class=Bridge_Points, cluster_tolerance="")

      # Process: Add Field (Add Field) (management)
      if AllWater_Diss_shp and Ext_AW and Kaldv_VannOmr and WB_Buf50_shp_2_:
         Bridge_Points_shp_2_ = arcpy.management.AddField(in_table=Bridge_Points,
field_name="Elevation", field_type="FLOAT", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="NON_REQUIRED", field_domain="")[0]

      # Process: Calculate Field (Calculate Field) (management)
      if AllWater_Diss_shp and Ext_AW and Kaldv_VannOmr and WB_Buf50_shp_2_:
         Bridge_Points_shp = arcpy.management.CalculateField(in_table=Bridge_Points_shp_2_,
field="Elevation", expression="!grid_code!", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

      # Process: Erase (Erase) (analysis)
      WP_Erase_shp = "WP_Erase.shp"
      if AllWater_Diss_shp and Bridge_Points_shp and Ext_AW and Kaldv_VannOmr and
WB_Buf50_shp_2_:
         arcpy.analysis.Erase(in_features=Bridge_Points_shp,
erase_features=Vann_Points_shp_2_, out_feature_class=WP_Erase_shp, cluster_tolerance="")

      # Process: IDW (IDW) (sa)
      IDW_Stream = "idw"
      IDW = IDW_Stream
      if AllWater_Diss_shp and Bridge_Points_shp and Ext_AW and Kaldv_VannOmr and
WB_Buf50_shp_2_:
         with arcpy.EnvManager(mask=AllWater_Diss_shp):
            IDW_Stream = arcpy.sa.Idw(in_point_features=WP_Erase_shp,
z_field="GRID_CODE", cell_size="0.5", power=2, search_radius="VARIABLE 12",
in_barrier_polyline_features="")
            IDW_Stream.save(IDW)
      # Process: Mosaic To New Raster (Mosaic To New Raster) (management)
      if AllWater_Diss_shp and Bridge_Points_shp and Ext_AW and Kaldv_VannOmr and
WB_Buf50_shp_2_:
         Output_mosiac_raster_name =
arcpy.management.MosaicToNewRaster(input_rasters=[Input_DEM, IDW_Stream],
output_location=_3_Remove_Bridges, raster_dataset_name_with_extension="dem05uBruVoll",
coordinate_system_for_the_raster="PROJCS[\"ETRS_1989_UTM_Zone_32N\",GEOGCS[\"GCS
_ETRS_1989\",DATUM[\"D_ETRS_1989\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101
]],PRIMEM[\"Greenwich\",0.0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transver
se_Mercator\"],PARAMETER[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],
PARAMETER[\"Central_Meridian\",9.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"L
atitude_Of_Origin\",0.0],UNIT[\"Meter\",1.0]],VERTCS[\"NN2000_height\",VDATUM[\"Norway_No
rmal_Null_2000\"],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"Met
er\",1.0]]", pixel_type="32_BIT_FLOAT", cellsize=Cellsize,
number_of_bands=Number_of_Bands, mosaic_method="LAST",
mosaic_colormap_mode="FIRST")[0]
         Output_mosiac_raster_name = arcpy.Raster(Output_mosiac_raster_name)

      # Process: Hillshade (Hillshade) (sa)
      Hillshade = HllShd05_mod
```

```
        if AllWater_Diss_shp and Bridge_Points_shp and Ext_AW and Kaldv_VannOmr and
WB_Buf50_shp_2_:
            HllShd05_mod = arcpy.sa.HillShade(in_raster=Output_mosiac_raster_name,
azimuth=315, altitude=45, model_shadows="NO_SHADOWS", z_factor=1)
            HllShd05_mod.save(Hillshade)

        return Output_mosiac_raster_name

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]"):
        RemoveBridgesTool(*argv[1:]
```

14.4: 4_Add_Culvert_Tool

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-02 14:46:59
"""
import arcpy
from sys import argv
def #  NOT  IMPLEMENTED# Function Body not implemented

def
AddCulverts(Watershed_Polygon="C:\\Users\\tskur\\OneDrive\\Dokumenter\\Master\\Masteroppg
aven\\Melhus_summerjob_2021\\02 Data_Melhus\\12 Model
Input\\Nedborfelt_RegineEnhet.shp", Input_DEM="dem_allmods",
Stikkrenne_Linjer="stikkrenne_kulvertl.shp", Output_DEM_Location="Output.gdb"):  #
Add_Culverts_Tool

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    arcpy.ImportToolbox(r"c:\program files\arcgis\pro\Resources\ArcToolbox\toolboxes\Data
Management Tools.tbx.tbx")
    # Model Environment settings
    with arcpy.EnvManager(scratchWorkspace=r"Scratch", workspace=r"Workspace"):
        Distance_value_or_field_ = "Buff_Dist"

        # Process: Clip (4) (Clip) (analysis)
        Stikk_Clip = "Stikk_Clip"
        arcpy.analysis.Clip(in_features=Stikkrenne_Linjer, clip_features=Watershed_Polygon,
out_feature_class=Stikk_Clip, cluster_tolerance="")

        # Process: Calculate Field (4) (Calculate Field) (management)
        Stikkrenne_Clip_shp_6_ = arcpy.management.CalculateField(in_table=Stikk_Clip,
field="DIAMETERIN", expression="Redef(!DIAMETERIN!, !BREDDEINNV!)",
expression_type="PYTHON3", code_block="""def Redef (diameterin, breddeinnv):
    if diameterin == 0:
        return breddeinnv
    elif diameterin == 0 and breddeinnv == 0:
        return 200
    else:
        return diameterin



""", field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

        # Process: Calculate Field (5) (Calculate Field) (management)
        Stikkrenne_Clip_shp_4_ =
arcpy.management.CalculateField(in_table=Stikkrenne_Clip_shp_6_, field="BREDDEINNV",
expression="!DIAMETERIN!/10", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

        # Process: Add Field (6) (Add Field) (management)
        Stikkrenne_Clip_shp_3_ = arcpy.management.AddField(in_table=Stikkrenne_Clip_shp_4_,
field_name="Buff_Dist", field_type="DOUBLE", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="NON_REQUIRED", field_domain="")[0]
```

```
    # Process: Calculate Field (8) (Calculate Field) (management)
    Stikkrenne_Clip_shp_8_ =
arcpy.management.CalculateField(in_table=Stikkrenne_Clip_shp_3_, field="Buff_Dist",
expression="!DIAMETERIN!/1000", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Buffer (2) (Buffer) (analysis)
    Stikk_Clip_Buff = "Stikk_Clip_Buff"
    arcpy.analysis.Buffer(in_features=Stikkrenne_Clip_shp_8_,
out_feature_class=Stikk_Clip_Buff, buffer_distance_or_field=Distance_value_or_field_,
line_side="FULL", line_end_type="ROUND", dissolve_option="NONE", dissolve_field=[],
method="PLANAR")

    # Process: Int (Int) (ia)
    INT_AllMods = "INT_AllMods"
    Int = INT_AllMods
    INT_AllMods = arcpy.ia.Int(in_raster_or_constant=Input_DEM)
    INT_AllMods.save(Int)


    # Process: Polygon to Raster (Polygon to Raster) (conversion)
    Stikkrenne_Clip_Buffer_PolygonToRaster_tif = "stikk_Clip_Buff_PolygonToRaster.tif"
    arcpy.conversion.PolygonToRaster(in_features=Stikk_Clip_Buff, value_field="OBJECTID",
out_rasterdataset=Stikkrenne_Clip_Buffer_PolygonToRaster_tif,
cell_assignment="CELL_CENTER", priority_field="NONE", cellsize=INT_AllMods,
build_rat="BUILD")

    # Process: If Field Value Is (If Field Value Is) ()
    True_107, False_108 = #  NOT  IMPLEMENTED(in_data=Stikk_Clip,
where_clause="KOORDH > 0", invert_where_clause="", selection_condition="EXISTS", count=0,
count_min=0, count_max=0)

    # Process: Add Geometry Attributes (Add Geometry Attributes) (management)
    if False_108:
        Stikk_Clip_2_ = arcpy.management.AddGeometryAttributes(Input_Features=Stikk_Clip,
Geometry_Properties=["LINE_START_MID_END"], Length_Unit="METERS",
Area_Unit="SQUARE_METERS",
Coordinate_System="PROJCS[\"ETRS_1989_UTM_Zone_32N\",GEOGCS[\"GCS_ETRS_1989\
",DATUM[\"D_ETRS_1989\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101]],PRIMEM[\"
Greenwich\",0.0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Mercator\
"],PARAMETER[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],PARAMETE
R[\"Central_Meridian\",9.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_
Origin\",0.0],UNIT[\"Meter\",1.0]],VERTCS[\"NN2000_height\",VDATUM[\"Norway_Normal_Null_2
000\"],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"Meter\",1.0]]")[0
]

    # Process: XY Table To Point Mid (XY Table To Point) (management)
    Stikk_Mid_shp = "Stikk_Mid"
    if False_108:
        arcpy.management.XYTableToPoint(in_table=Stikk_Clip_2_,
out_feature_class=Stikk_Mid_shp, x_field="MID_X", y_field="MID_Y", z_field="MID_Z",
coordinate_system="PROJCS[\"ETRS_1989_UTM_Zone_32N\",GEOGCS[\"GCS_ETRS_1989\",
DATUM[\"D_ETRS_1989\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101]],PRIMEM[\"Gr
eenwich\",0.0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Mercator\"],
PARAMETER[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],PARAMETER[\
"Central_Meridian\",9.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_Ori
gin\",0.0],UNIT[\"Meter\",1.0]],VERTCS[\"NN2000 height\",VDATUM[\"Norway Normal Null
```

```
2000\"],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"metre\",1.0]];-5120900 -9998100 450445547371518;-10460459235 419430400195312;-100000 10000;1;1;1;IsHighPrecision")

    # Process: Extract Values to Points Mid (Extract Values to Points) (sa)
    Extract_Stikk_Mid_shp = "Extract_Stikk_Mid.shp"
    if False_108:
        arcpy.sa.ExtractValuesToPoints(in_point_features=Stikk_Mid_shp, in_raster=Input_DEM, out_point_features=Extract_Stikk_Mid_shp, interpolate_values="NONE", add_attributes="VALUE_ONLY")

    # Process: Add Z_Mid (Add Field) (management)
    if False_108:
        Extract_Stikk_Mid = arcpy.management.AddField(in_table=Extract_Stikk_Mid_shp, field_name="Z_Mid", field_type="DOUBLE", field_precision=None, field_scale=None, field_length=None, field_alias="", field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field Z_Mid Value (Calculate Field) (management)
    if False_108:
        Extract_Stikk_Mid_shp_2_ = arcpy.management.CalculateField(in_table=Extract_Stikk_Mid, field="Z_Mid", expression="!RASTERVALU!", expression_type="PYTHON3", code_block="", field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: XY Table To Point End (XY Table To Point) (management)
    Stikk_End_shp = "Stikk_End.shp"
    if False_108:
        arcpy.management.XYTableToPoint(in_table=Stikk_Clip_2_, out_feature_class=Stikk_End_shp, x_field="END_X", y_field="END_Y", z_field="END_Z", coordinate_system="PROJCS[\"ETRS_1989_UTM_Zone_32N\",GEOGCS[\"GCS_ETRS_1989\",DATUM[\"D_ETRS_1989\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Mercator\"],PARAMETER[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],PARAMETER[\"Central_Meridian\",9.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_Origin\",0.0],UNIT[\"Meter\",1.0]],VERTCS[\"NN2000 height\",VDATUM[\"Norway Normal Null 2000\"],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"metre\",1.0]];-5120900 -9998100 450445547371518;-10460459235 419430400195312;-100000 10000;1;1;1;IsHighPrecision")

    # Process: Extract Values to Points End (Extract Values to Points) (sa)
    Extract_Stikk_End_shp = "Extract_Stikk_End.shp"
    if False_108:
        arcpy.sa.ExtractValuesToPoints(in_point_features=Stikk_End_shp, in_raster=Input_DEM, out_point_features=Extract_Stikk_End_shp, interpolate_values="NONE", add_attributes="VALUE_ONLY")

    # Process: Add Field Z_End (Add Field) (management)
    if False_108:
        Extract_Stikk_End = arcpy.management.AddField(in_table=Extract_Stikk_End_shp, field_name="Z_End", field_type="DOUBLE", field_precision=None, field_scale=None, field_length=None, field_alias="", field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field Z_End Value (Calculate Field) (management)
    if False_108:
```

```
        Extract_Stikk_End_shp_3_ =
arcpy.management.CalculateField(in_table=Extract_Stikk_End, field="Z_End",
expression="!RASTERVALU!", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Add Join (Add Join) (management)
    if False_108:
        Extract_Stikk_Mid_Layer =
arcpy.management.AddJoin(in_layer_or_view=Extract_Stikk_Mid_shp_2_, in_field="FID",
join_table=Extract_Stikk_End_shp_3_, join_field="FID", join_type="KEEP_ALL",
index_join_fields="NO_INDEX_JOIN_FIELDS")[0]

    # Process: XY Table To Point Start (XY Table To Point) (management)
    Stikk_Start_shp = "Stikk_Start.shp"
    if False_108:
        arcpy.management.XYTableToPoint(in_table=Stikk_Clip_2_,
out_feature_class=Stikk_Start_shp, x_field="START_X", y_field="START_Y",
z_field="START_Z",
coordinate_system="PROJCS[\"ETRS_1989_UTM_Zone_32N\",GEOGCS[\"GCS_ETRS_1989\",
DATUM[\"D_ETRS_1989\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101]],PRIMEM[\"Gr
eenwich\",0.0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Mercator\"],
PARAMETER[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],PARAMETER[\
"Central_Meridian\",9.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_Ori
gin\",0.0],UNIT[\"Meter\",1.0]],VERTCS[\"NN2000 height\",VDATUM[\"Norway Normal Null
2000\"],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"metre\",1.0]];-
5120900 -9998100 450445547371518;-10460459235 419430400195312;-100000
10000;1;1;1;IsHighPrecision")

    # Process: Extract Values to Points Start (Extract Values to Points) (sa)
    Extract_Stikk_Start_shp = "Extract_Stikk_Start.shp"
    if False_108:
        arcpy.sa.ExtractValuesToPoints(in_point_features=Stikk_Start_shp,
in_raster=Input_DEM, out_point_features=Extract_Stikk_Start_shp, interpolate_values="NONE",
add_attributes="VALUE_ONLY")

    # Process: Add Z_Start (Add Field) (management)
    if False_108:
        Extract_Stikk_Start = arcpy.management.AddField(in_table=Extract_Stikk_Start_shp,
field_name="Z_Start", field_type="FLOAT", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field Z_Start Value (Calculate Field) (management)
    if False_108:
        Extract_Stikk_Start_shp_3_ =
arcpy.management.CalculateField(in_table=Extract_Stikk_Start, field="Z_Start",
expression="!RASTERVALU!", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Add Join (2) (Add Join) (management)
    if False_108:
        Extract_Stikk_Mid_Layer_3_ =
arcpy.management.AddJoin(in_layer_or_view=Extract_Stikk_Mid_Layer,
in_field="Extract_Stikk_End.FID", join_table=Extract_Stikk_Start_shp_3_, join_field="FID",
join_type="KEEP_ALL", index_join_fields="NO_INDEX_JOIN_FIELDS")[0]

    # Process: Add Field (5) (Add Field) (management)
```

```
    if False_108:
        Extract_Stikk_Mid_Layer_2_ = 
arcpy.management.AddField(in_table=Extract_Stikk_Mid_Layer_3_, field_name="ReCa_Z_mid", 
field_type="DOUBLE", field_precision=None, field_scale=None, field_length=None, 
field_alias="", field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED", 
field_domain="")[0]

    # Process: Calculate Field (7) (Calculate Field) (management)
    if False_108:
        Extract_Stikk_Mid_Layer_4_ = 
arcpy.management.CalculateField(in_table=Extract_Stikk_Mid_Layer_2_, field="ReCa_Z_mid", 
expression="(!Extract_Stikk_End.Z_End! + !Extract_Stikk_Start.Z_Start!) / 2", 
expression_type="PYTHON3", code_block="", field_type="TEXT", 
enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Add Join (3) (Add Join) (management)
    if False_108:
        Stikkrenne_Clip_Buffer_Polyg = 
arcpy.management.AddJoin(in_layer_or_view=Stikkrenne_Clip_Buffer_PolygonToRaster_tif, 
in_field="OID", join_table=Extract_Stikk_Mid_Layer_4_, join_field="Extract_Stikk_Mid.shp.FID", 
join_type="KEEP_COMMON", index_join_fields="NO_INDEX_JOIN_FIELDS")[0]

    # Process: Lookup (Lookup) (sa)
    RasRe_Mid_Z = "RasRe_Mid_Z"
    Lookup = RasRe_Mid_Z
    if False_108:
        RasRe_Mid_Z = arcpy.sa.Lookup(in_raster=Stikkrenne_Clip_Buffer_Polyg, 
lookup_field="Extract_Stikk_Mid.ReCa_Z_mid")
        RasRe_Mid_Z.save(Lookup)


    # Process: Lookup (2) (Lookup) (sa)
    LU_KOORDH_tif = "LU_KOORDH.tif"
    Lookup_2_ = LU_KOORDH_tif
    if True_107:
        LU_KOORDH_tif = 
arcpy.sa.Lookup(in_raster=Stikkrenne_Clip_Buffer_PolygonToRaster_tif, 
lookup_field="KOORDH")
        LU_KOORDH_tif.save(Lookup_2_)


    # Process: Mosaic To New Raster (Mosaic To New Raster) (management)
    if False_108 and True_107:
        Output_DEM = arcpy.management.MosaicToNewRaster(input_rasters=[Input_DEM, 
RasRe_Mid_Z, LU_KOORDH_tif], output_location=Output_DEM_Location, 
raster_dataset_name_with_extension="UbruVollstikk", 
coordinate_system_for_the_raster="PROJCS[\"ETRS_1989_UTM_Zone_32N\",GEOGCS[\"GCS
_ETRS_1989\",DATUM[\"D_ETRS_1989\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101
]],PRIMEM[\"Greenwich\",0.0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transver
se_Mercator\"],PARAMETER[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],
PARAMETER[\"Central_Meridian\",9.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"L
atitude_Of_Origin\",0.0],UNIT[\"Meter\",1.0]],VERTCS[\"NN2000 height\",VDATUM[\"Norway 
Normal Null 
2000\"],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"metre\",1.0]]", 
pixel_type="32_BIT_FLOAT", cellsize=0, number_of_bands=1, mosaic_method="LAST", 
mosaic_colormap_mode="FIRST")[0]
        Output_DEM = arcpy.Raster(Output_DEM)
```

```
        return Output_DEM

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]"):
        AddCulverts(*argv[1:])
```

14.5: 5_Build_Levee

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-02 14:47:10
"""
import arcpy
from sys import argv

def BuildLevee(Flomvoll_punkter="Flomvoll_punkter.shp", Distance_value_or_field_="5 Meters",
Interpolert_flomvoll="C:DemManipulationTool.gdb\\IDW", Output_Folder="workspace.gdb",
Pixel_Type="32_BIT_FLOAT", Cellsize=0.5, DEM_allmods="DEM_allmods"):  #
5_Build_Levee_Tool

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("3D")
    arcpy.CheckOutExtension("spatial")

    arcpy.ImportToolbox(r"c:\program files\arcgis\pro\Resources\ArcToolbox\toolboxes\Data
Management Tools.tbx.tbx")
    # Model Environment settings
    with arcpy.EnvManager(snapRaster="dem_05"):

        # Process: Points To Line (Points To Line) (management)
        Flomvoll_Linje = %Name%_Line.shp"
        arcpy.management.PointsToLine(Input_Features=Flomvoll_punkter,
Output_Feature_Class=Flomvoll_Linje, Line_Field="", Sort_Field="", Close_Line="NO_CLOSE")

        # Process: Buffer (Buffer) (analysis)
        Flomvoll_Buffer_Maske = "Buffer"
        arcpy.analysis.Buffer(in_features=Flomvoll_Linje,
out_feature_class=Flomvoll_Buffer_Maske, buffer_distance_or_field=Distance_value_or_field_,
line_side="FULL", line_end_type="ROUND", dissolve_option="ALL", dissolve_field=[],
method="PLANAR")

        # Process: IDW (IDW) (sa)
        IDW = Interpolert_flomvoll
        with arcpy.EnvManager(mask=Flomvoll_Buffer_Maske):
            Interpolert_flomvoll = arcpy.sa.Idw(in_point_features=Flomvoll_punkter, z_field="Z",
cell_size=DEM_allmods, power=2, search_radius="VARIABLE 4",
in_barrier_polyline_features="")
            Interpolert_flomvoll.save(IDW)


        # Process: Mosaic To New Raster (Mosaic To New Raster) (management)
        Modified_Output_Raster =
arcpy.management.MosaicToNewRaster(input_rasters=[Interpolert_flomvoll, DEM_allmods],
output_location=Output_Folder, raster_dataset_name_with_extension="dem_05_Levee",
coordinate_system_for_the_raster="PROJCS[\"ETRS_1989_UTM_Zone_32N\",GEOGCS[\"GCS
_ETRS_1989\",DATUM[\"D_ETRS_1989\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101
]],PRIMEM[\"Greenwich\",0.0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transver
se_Mercator\"],PARAMETER[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],
PARAMETER[\"Central_Meridian\",9.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"L
atitude_Of_Origin\",0.0],UNIT[\"Meter\",1.0]],VERTCS[\"NN2000 height\",VDATUM[\"Norway
Normal Null
```

```
2000\"],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"metre\",1.0]]",
pixel_type=Pixel_Type, cellsize=Cellsize, number_of_bands=1, mosaic_method="LAST",
mosaic_colormap_mode="FIRST")[0]
        Modified_Output_Raster = arcpy.Raster(Modified_Output_Raster)

        return Modified_Output_Raster

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"scratchWorkspace", workspace=r"Workspace"):
        BuildLevee(*argv[1:])
```

14.6: 6_Generate_Blockage_Tool

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 18:59:51
"""
import arcpy
from ToolTest.Iterate_Feature_2_Raster import Iterate_Feature_2_Raster
from sys import argv

def Generate_Blockage(CP_Kaldvella="CP_Kaldvella",
Clip_fkb_vann_omrade_shp="Clip_fkb_vann_omrade.shp", dem_allmods="dem_allmods",
Fill_Rasters_gdb="Fill_Rasters.gdb", _Name_="%Name%"):  # Generate_Blockage

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("spatial")
    arcpy.CheckOutExtension("ImageExt")
    arcpy.CheckOutExtension("ImageAnalyst")

    arcpy.ImportToolbox(r"c:\program files\arcgis\pro\Resources\ArcToolbox\toolboxes\Analysis
Tools.tbx.tbx")
    # Model Environment settings
    with arcpy.EnvManager(scratchWorkspace=r"scratchWorkspace", workspace=r"Workspace"):
        Fill100_gdb_2_ = "Fill100.gdb"

        # Process: Buffer (4) (Buffer) (analysis)
        CP_Buf10 = "CP_Buf10"
        arcpy.analysis.Buffer(in_features=CP_Kaldvella, out_feature_class=CP_Buf10,
buffer_distance_or_field="10 Meters", line_side="FULL", line_end_type="ROUND",
dissolve_option="NONE", dissolve_field=[], method="PLANAR")

        # Process: Clip (4) (Clip) (analysis)
        CP_Crit =
"C:\\Users\\tskur\\OneDrive\\Dokumenter\\Master\\Masteroppgaven\\Melhus_summerjob_2021\\0
3 GIS maps and projects\\Tool_Test\\CP_Fill_Scratch.gdb\\CP_Crit"
        arcpy.analysis.Clip(in_features=CP_Buf10, clip_features=Clip_fkb_vann_omrade_shp,
out_feature_class=CP_Crit, cluster_tolerance="")

        # Process: Buffer (5) (Buffer) (analysis)
        BF_05 = "BF_05"
        arcpy.analysis.Buffer(in_features=CP_Crit, out_feature_class=BF_05,
buffer_distance_or_field="1 Meters", line_side="FULL", line_end_type="ROUND",
dissolve_option="NONE", dissolve_field=[], method="PLANAR")

        # Process: Zonal Statistics as Table (Zonal Statistics as Table) (ia)
        ZS = "ZS"
        arcpy.ia.ZonalStatisticsAsTable(in_zone_data=BF_05, zone_field="ORIG_FID",
in_value_raster=dem_allmods, out_table=ZS, ignore_nodata="DATA", statistics_type="ALL",
process_as_multidimensional="CURRENT_SLICE", percentile_values=90,
percentile_interpolation_type="AUTO_DETECT")
        .save(Zonal_Statistics_as_Table)


        # Process: Add Join (Add Join) (management)
```

```
    B_ZST = arcpy.management.AddJoin(in_layer_or_view=BF_05, in_field="OBJECTID",
join_table=ZS, join_field="ORIG_FID", join_type="KEEP_ALL",
index_join_fields="INDEX_JOIN_FIELDS")[0]

    # Process: Add Field (2) (Add Field) (management)
    B_ZST_50 = arcpy.management.AddField(in_table=B_ZST, field_name="F50",
field_type="DOUBLE", field_precision=3, field_scale=None, field_length=None, field_alias="",
field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field (2) (Calculate Field) (management)
    B_ZST_50C = arcpy.management.CalculateField(in_table=B_ZST_50, field="F50",
expression="!ZS.RANGE!*0.50+!ZS.MIN!", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Add Field (4) (Add Field) (management)
    B_ZST_1 = arcpy.management.AddField(in_table=B_ZST_50C, field_name="F100",
field_type="DOUBLE", field_precision=3, field_scale=None, field_length=None, field_alias="",
field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field (4) (Calculate Field) (management)
    B_ZST_1C = arcpy.management.CalculateField(in_table=B_ZST_1, field="F100",
expression="!ZS.RANGE!*1+!ZS.MIN!", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Select (4) (Select) (analysis)
    Select_100 = "Select_100"
    arcpy.analysis.Select(in_features=B_ZST_1C, out_feature_class=Select_100,
where_clause="BF_05.F100 IS NOT NULL")

    # Process: Split By Attributes (4) (Split By Attributes) (analysis)
    Fill100_gdb = arcpy.analysis.SplitByAttributes(Input_Table=Select_100,
Target_Workspace=Fill100_gdb_2_, Split_Fields=["BF_05_ORIG_FID"])[0]

    # Process: Iterate_Feature_2_Raster (3) (Iterate_Feature_2_Raster) (ToolTest)
    Iterate_Feature_2_Raster(Output_Raster=_Name_, Fill100_gdb=Fill100_gdb,
Field="BF_05_F100", Fill_Rasters_gdb=Fill_Rasters_gdb)
    _Name_ = arcpy.Raster(_Name_)

if __name__ == '__main__':
  # Global Environment settings
  with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]"):
    Generate_Blockage(*argv[1:])
```

14.6.1: 6_1_Iterate_Feature_Class_2_Raster

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:03:43
"""
import arcpy
import os

from sys import argv
def FeatureClassGenerator(workspace, wild_card, feature_type, recursive) :
  with arcpy.EnvManager(workspace = workspace):

    dataset_list = [""]
    if recursive:
      datasets = arcpy.ListDatasets()
      dataset_list.extend(datasets)

    for dataset in dataset_list:
      featureclasses = arcpy.ListFeatureClasses(wild_card, feature_type, dataset)
      for fc in featureclasses:
        yield os.path.join(workspace, dataset, fc), fc


def Iterate_Feature_2_Raster(Output_Raster=fr"Raster_{Name}", Fill100_gdb="Fill100.gdb",
Field="BF_05_F100", Fill_Rasters_gdb="Fill_Rasters.gdb"):  # Iterate_Feature_2_Raster

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False


    for FeatureClass, Name in FeatureClassGenerator(Fill100_gdb, "", "", "NOT_RECURSIVE"):

        # Process: Feature to Raster (Feature to Raster) (conversion)
        with arcpy.EnvManager(workspace=Fill_Rasters_gdb):
            arcpy.conversion.FeatureToRaster(in_features=FeatureClass, field=Field,
out_raster=Output_Raster, cell_size="0.5")
            Output_Raster = arcpy.Raster(Output_Raster)

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"scratchWorkspace", workspace=r"Workspace"):
        Iterate_Feature_2_Raster(*argv[1:])
```

14.7: 7_Hydro_Tool

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 20:47:44
"""
import arcpy
from ToolTest.Hydro_No_Block import Hydro_No_Block
from ToolTest.Iterate_Con import Iterate_Con
from ToolTest.Iterate_Erase import Iterate_Erase
from ToolTest.Iterate_Fill import Iterate_Fill
from ToolTest.Iterate_FlowAcc import Iterate_FlowAcc
from ToolTest.Iterate_FlowDir import Iterate_FlowDir
from ToolTest.Iterate_Int import Iterate_Int
from ToolTest.Iterate_Layername import Iterate_Layername
from ToolTest.Iterate_Mosaic import Iterate_Mosaic
from ToolTest.Iterate_Ras_2_PL import Iterate_Ras_2_PL
from ToolTest.Iterate_areaRecalc import Iterate_areaRecalc
from sys import argv

def Iterate_Hydro(dem_allmods="dem_allmods", Fill_Rasters_gdb="Fill_Rasters.gdb",
Flow_direction_type="D8", Input_flow_direction_type="D8"):  # Iterate_Hydro

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("spatial")
    arcpy.CheckOutExtension("3D")
    arcpy.CheckOutExtension("ImageAnalyst")

    Iterate_Bluespots_gdb = "Iterate_Bluespots.gdb"
    Ras_2_PL_IT_gdb = "Ras_2_PL_IT.gdb"
    Erase_IT_gdb = "Erase_IT.gdb"
    Clip_fkb_vann_omrade_shp = "Clip_fkb_vann_omrade.shp"
    Int_IT_gdb = "Int_IT.gdb"
    Con_IT_gdb = "Con_IT.gdb"
    Flow_Acc_IT_gdb = "Flow_Acc_IT.gdb"
    Flow_Dir_IT_gdb = "Flow_Dir_IT.gdb"
    Fill_IT_gdb = "CFill_IT.gdb"
    Output_FCP_gdb = "Output_FCP.gdb"

    # Process: Iterate_Mosaic (Iterate_Mosaic) (ToolTest)
    _Name_ = Iterate_Mosaic(Output_FCP_gdb=Output_FCP_gdb,
Input_base_raster=dem_allmods, Workspace=Fill_Rasters_gdb,
Raster_Dataset_Name_with_Extension="%Name%")[0]
    _Name_ = arcpy.Raster(_Name_)

    # Process: Iterate_Fill (Iterate_Fill) (ToolTest)
    _Name_2_ = "Fill_IT.gdb\\%Name%"
    if _Name_:
        Iterate_Fill(Workspace=Output_FCP_gdb, Fill_IT_gdb=Fill_IT_gdb,
Fill__Name_=_Name_2_)
        _Name_2_ = arcpy.Raster(_Name_2_)

    # Process: Iterate_FlowDir (Iterate_FlowDir) (ToolTest)
    _Name_3_ = "Flow_Dir_IT.gdb\\%Name%"
    if _Name_ and _Name_2_:
```

```
        Iterate_FlowDir(Workspace=Fill_IT_gdb, Flow_Dir_IT_gdb=Flow_Dir_IT_gdb,
FD__Name_=_Name_3_, Flow_direction_type=Flow_direction_type)
        _Name_3_ = arcpy.Raster(_Name_3_)

    # Process: Iterate_FlowAcc (Iterate_FlowAcc) (ToolTest)
    _Name_4_ = "Flow_Acc_IT.gdb\\%Name%"
    if _Name_ and _Name_2_ and _Name_3_:
        Iterate_FlowAcc(Workspace=Flow_Dir_IT_gdb, Flow_Acc_IT_gdb=Flow_Acc_IT_gdb,
FA_Name_=_Name_4_, Input_flow_direction_type=Input_flow_direction_type,
Input_weight_raster="")
        _Name_4_ = arcpy.Raster(_Name_4_)

    # Process: Iterate_Con (Iterate_Con) (ToolTest)
    _Name_5_ = "Con_IT.gdb\\%Name%"
    if _Name_ and _Name_2_ and _Name_3_ and _Name_4_:
        Iterate_Con(Workspace=Flow_Acc_IT_gdb, Con_IT_gdb=Con_IT_gdb,
Expression="VALUE >= 50000", C__Name_=_Name_5_)
        _Name_5_ = arcpy.Raster(_Name_5_)

    # Process: Iterate_Int (Iterate_Int) (ToolTest)
    _Name_6_ = "CInt_IT.gdb\\%Name%"
    if _Name_ and _Name_2_ and _Name_3_ and _Name_4_ and _Name_5_:
        Iterate_Int(Workspace=Con_IT_gdb, Int_IT_gdb=Int_IT_gdb, Int__Name_=_Name_6_)
        _Name_6_ = arcpy.Raster(_Name_6_)

    # Process: Iterate_Ras_2_PL (Iterate_Ras_2_PL) (ToolTest)
    _Name_7_ = "Ras_2_PL_IT.gdb\\%Name%"
    if _Name_ and _Name_2_ and _Name_3_ and _Name_4_ and _Name_5_ and _Name_6_:
        Iterate_Ras_2_PL(Ras_2_PL_IT_gdb=Ras_2_PL_IT_gdb, Int_IT_gdb=Int_IT_gdb,
R2PL__Name_=_Name_7_)

    # Process: Hydro_No_Block (Hydro_No_Block) (ToolTest)
    Waterline = "Erase_IT.gdb\\Waterline"
    if _Name_ and _Name_2_ and _Name_3_ and _Name_4_ and _Name_5_ and _Name_6_
and _Name_7_:
        Hydro_No_Block(DEM_allmods=dem_allmods, Flow_direction_type="D8",
Input_flow_direction_type="D8", Waterline=Waterline)

    # Process: Iterate_areaRecalc (Iterate_areaRecalc) (ToolTest)
    if _Name_ and _Name_2_ and _Name_3_ and _Name_4_ and _Name_5_ and _Name_6_
and _Name_7_:
        R2PL_Int_C_FAFD_F_MRaster_No_Block = Iterate_areaRecalc()[0]

    # Process: Iterate_Erase (Iterate_Erase) (ToolTest)
    _Name_8_ = "Erase_IT.gdb\\%Name%"
    if R2PL_Int_C_FAFD_F_MRaster_No_Block and Waterline and _Name_ and _Name_2_ and
_Name_3_ and _Name_4_ and _Name_5_ and _Name_6_ and _Name_7_:
        Iterate_Erase(Ras_2_PL_IT_gdb=Ras_2_PL_IT_gdb, Erase_IT_gdb=Erase_IT_gdb,
Clip_fkb_vann_omrade_shp=Clip_fkb_vann_omrade_shp, _Name_=_Name_8_,
NB_Waterline__2_=Waterline)

    # Process: Iterate_Layername (Iterate_Layername) (ToolTest)
    if R2PL_Int_C_FAFD_F_MRaster_No_Block and Waterline and _Name_ and _Name_2_ and
_Name_3_ and _Name_4_ and _Name_5_ and _Name_6_ and _Name_7_ and _Name_8_:
        T1_2_ = Iterate_Layername(Iterate_Bluespots_gdb=Iterate_Bluespots_gdb)[0]

    # Process: Copy Features (Copy Features) (management)
```

```
    NoFill = "Erase_IT.gdb\\NoFill"
    if _Name_ and _Name_2_ and _Name_3_ and _Name_4_ and _Name_5_ and _Name_6_
and _Name_7_:
        arcpy.management.CopyFeatures(in_features=Waterline, out_feature_class=NoFill,
config_keyword="", spatial_grid_1=None, spatial_grid_2=None, spatial_grid_3=None)

    return T1_2_

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_Hydro(*argv[1:])
```

14.7.1: 7_1_Iterate mosaic

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:25:33
"""
import arcpy
from sys import argv
def #  NOT  IMPLEMENTED# Function Body not implemented

def Iterate_Mosaic(Output_FCP_gdb="Output_FCP.gdb", Input_base_raster="dem_allmods", Workspace="Fill_Rasters.gdb", Raster_Dataset_Name_with_Extension=f"M{Name}"):  # Iterate_Mosaic

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False


    for Raster, Name in #  NOT  IMPLEMENTED(Workspace, "", "", "NOT_RECURSIVE"):

        # Process: Mosaic To New Raster (Mosaic To New Raster) (management)
        Mosaic_Name_ = 
arcpy.management.MosaicToNewRaster(input_rasters=[Input_base_raster, Raster], output_location=Output_FCP_gdb, raster_dataset_name_with_extension=Raster_Dataset_Name_with_Extension, coordinate_system_for_the_raster="PROJCS[\"ETRS_1989_UTM_Zone_32N\",GEOGCS[\"GCS_ETRS_1989\",DATUM[\"D_ETRS_1989\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0.0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Mercator\"],PARAMETER[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],PARAMETER[\"Central_Meridian\",9.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_Origin\",0.0],UNIT[\"Meter\",1.0]],VERTCS[\"NN2000_height\",VDATUM[\"Norway_Normal_Null_2000\"],PARAMETER[\"Vertical_Shift\",0.0],PARAMETER[\"Direction\",1.0],UNIT[\"Meter\",1.0]]", pixel_type="32_BIT_FLOAT", cellsize=0.5, number_of_bands=1, mosaic_method="LAST", mosaic_colormap_mode="FIRST")[0]
        Mosaic_Name_ = arcpy.Raster(Mosaic_Name_)

        return Mosaic_Name_

if __name__ == '__main__':
    # Global Environment settings
    with 
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOGCS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.257222101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latitude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]", scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_Mosaic(*argv[1:])
```

14.7.2: 7_2_Iterate_Fill

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:23:13
"""
import arcpy
from sys import argv
def #  NOT  IMPLEMENTED# Function Body not implemented

def Iterate_Fill(Workspace="Output_FCP.gdb", Fill_IT_gdb="Fill_IT.gdb",
Fill_Name_=fr"Fill_{Name}"):  # Iterate_Fill

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("spatial")

    for Raster, Name in #  NOT  IMPLEMENTED(Workspace, "", "", "NOT_RECURSIVE"):

        # Process: Fill (Fill) (sa)
        Fill = Fill_Name_
        with arcpy.EnvManager(workspace=Fill_IT_gdb):
            Fill_Name_ = arcpy.sa.Fill(in_surface_raster=Raster, z_limit=None)
            Fill_Name_.save(Fill)

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_Fill(*argv[1:])
```

14.7.3: 7_3_Iterate_FlowDir

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:23:38
"""
import arcpy
from sys import argv
def #  NOT  IMPLEMENTED# Function Body not implemented
def Iterate_FlowDir(Workspace="Fill_IT.gdb", Flow_Dir_IT_gdb="Flow_Dir_IT.gdb",
FD_Name_=fr"FD_{Name}", Flow_direction_type="D8"):  # Iterate_FlowDir

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("spatial")

    for Raster, Name in #  NOT  IMPLEMENTED(Workspace, "", "", "NOT_RECURSIVE"):
        # Process: Flow Direction (Flow Direction) (sa)
        Flow_Direction = FD_Name_
        Output_drop_raster = ""
        with arcpy.EnvManager(workspace=Flow_Dir_IT_gdb):
            FD_Name_ = arcpy.sa.FlowDirection(in_surface_raster=Raster, force_flow="NORMAL",
out_drop_raster=Output_drop_raster, flow_direction_type=Flow_direction_type)
            FD_Name_.save(Flow_Direction)

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_FlowDir(*argv[1:])
```

14.7.4: 7_4_Iterate_FlowAcc

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:23:27
"""
import arcpy
from sys import argv
def #  NOT  IMPLEMENTED# Function Body not implemented

def Iterate_FlowAcc(Workspace="Flow_Dir_IT.gdb", Flow_Acc_IT_gdb="Flow_Acc_IT.gdb",
FA_Name_=fr"\FA{Name}", Input_flow_direction_type="D8", Input_weight_raster):  #
Iterate_FlowAcc

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False
    # Check out any necessary licenses.
    arcpy.CheckOutExtension("spatial")
    for Raster, Name in #  NOT  IMPLEMENTED(Workspace, "", "", "NOT_RECURSIVE"):
        # Process: Flow Accumulation (Flow Accumulation) (sa)
        Flow_Accumulation = FA_Name_
        with arcpy.EnvManager(workspace=Flow_Acc_IT_gdb):
            FA_Name_ = arcpy.sa.FlowAccumulation(in_flow_direction_raster=Raster,
in_weight_raster=Input_weight_raster, data_type="FLOAT",
flow_direction_type=Input_flow_direction_type)
            FA_Name_.save(Flow_Accumulation)
if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_FlowAcc(*argv[1:])
```

14.7.5: 7_5_Iterate_Con

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:22:39
"""
import arcpy
from sys import argv
def #  NOT  IMPLEMENTED# Function Body not implemented

def Iterate_Con(Workspace="Flow_Acc_IT.gdb", Con_IT_gdb="Con_IT.gdb",
Expression="VALUE >= 25000", C_Name_=fr"Con_IT.gdb\C_{Name}"):  # Iterate_Con

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("spatial")
    arcpy.CheckOutExtension("ImageAnalyst")

    Input_false_raster_or_constant_value = 0

    for Raster, Name in #  NOT  IMPLEMENTED(Workspace, "", "", "NOT_RECURSIVE"):

        # Process: Con (Con) (ia)
        Con = C_Name_
        with arcpy.EnvManager(workspace=Con_IT_gdb):
            C_Name_ = arcpy.ia.Con(in_conditional_raster=Raster,
in_true_raster_or_constant=Raster,
in_false_raster_or_constant=Input_false_raster_or_constant_value, where_clause=Expression)
            C_Name_.save(Con)


if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_Con(*argv[1:])
```

14.7.6: 7_6_Iterate_Int

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:23:50
"""
import arcpy
from sys import argv
def #  NOT  IMPLEMENTED# Function Body not implemented

def Iterate_Int(Workspace="ReClass_IT.gdb", Int_IT_gdb="Int_IT.gdb",
Int_Name_=fr"Int_{Name}"):  # Iterate_Int

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("3D")
    arcpy.CheckOutExtension("spatial")
    arcpy.CheckOutExtension("ImageAnalyst")

    for Raster, Name in #  NOT  IMPLEMENTED(Workspace, "", "", "NOT_RECURSIVE"):

        # Process: Int (Int) (ia)
        Int = Int_Name_
        with arcpy.EnvManager(workspace=Int_IT_gdb):
            Int_Name_ = arcpy.ia.Int(in_raster_or_constant=Raster)
            Int_Name_.save(Int)

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_Int(*argv[1:])
```

14.7.7: 7_7_Iterate_Ras_2_PL

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:25:44
"""
import arcpy
from sys import argv
def #  NOT  IMPLEMENTED# Function Body not implemented

def Iterate_Ras_2_PL(Ras_2_PL_IT_gdb="Ras_2_PL_IT.gdb", Int_IT_gdb="Int_IT.gdb",
R2PL_Name_=fr"R2PL_{Name}"):  # Iterate_Ras_2_PL

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False


    for Raster, Name in #  NOT  IMPLEMENTED(Int_IT_gdb, "", "", "NOT_RECURSIVE"):

        # Process: Raster to Polyline (Raster to Polyline) (conversion)
        with arcpy.EnvManager(outputMFlag="Disabled", outputZFlag="Disabled",
workspace=Ras_2_PL_IT_gdb):
            arcpy.conversion.RasterToPolyline(in_raster=Raster,
out_polyline_features=R2PL_Name_, background_value="ZERO", minimum_dangle_length=0,
simplify="SIMPLIFY", raster_field="Value")

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_Ras_2_PL(*argv[1:])
```

14.7.8: 7_8_Iterate_Erase

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:22:59
"""
import arcpy
import os

from sys import argv
def FeatureClassGenerator(workspace, wild_card, feature_type, recursive) :
  with arcpy.EnvManager(workspace = workspace):

    dataset_list = [""]
    if recursive:
      datasets = arcpy.ListDatasets()
      dataset_list.extend(datasets)

    for dataset in dataset_list:
      featureclasses = arcpy.ListFeatureClasses(wild_card, feature_type, dataset)
      for fc in featureclasses:
        yield os.path.join(workspace, dataset, fc), fc


def Iterate_Erase(Ras_2_PL_IT_gdb="Ras_2_PL_IT.gdb", Erase_IT_gdb="Erase_IT.gdb",
Clip_fkb_vann_omrade_shp="Clip_fkb_vann_omrade.shp", _Name_=fr"Erase_IT.gdb\{Name}",
NB_Waterline_2_="NB_Waterline"):  # Iterate_Erase

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False


    for FeatureClass, Name in FeatureClassGenerator(Ras_2_PL_IT_gdb, "", "",
"NOT_RECURSIVE"):

        # Process: Buffer (Buffer) (analysis)
        Drain_NoBlock_Buf50 = "Drain_NoBlock_Buf50"
        with arcpy.EnvManager(workspace=Erase_IT_gdb):
            arcpy.analysis.Buffer(in_features=NB_Waterline_2_,
out_feature_class=Drain_NoBlock_Buf50, buffer_distance_or_field="0.5 Meters",
line_side="FULL", line_end_type="ROUND", dissolve_option="ALL", dissolve_field=[],
method="PLANAR")

        # Process: Erase (Erase) (analysis)
        Name = "T12"
        _Name_Erase_Scratch = fr"Scratch_Gen_Block.gdb\{Name}Erase_Scratch"
        if Drain_NoBlock_Buf50 and FeatureClass:
            arcpy.analysis.Erase(in_features=FeatureClass, erase_features=Drain_NoBlock_Buf50,
out_feature_class=_Name_Erase_Scratch, cluster_tolerance="")

        # Process: Erase (2) (Erase) (analysis)
        if Drain_NoBlock_Buf50 and FeatureClass and _Name_Erase_Scratch:
            arcpy.analysis.Erase(in_features=_Name_Erase_Scratch,
erase_features=Clip_fkb_vann_omrade_shp, out_feature_class=_Name_, cluster_tolerance="")

if __name__ == '__main__':
    # Global Environment settings
```

```
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOGCS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.257222101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latitude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]", scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_Erase(*argv[1:])
```

14.7.9: 7_9_Iterate_Layername

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:24:36
"""
import arcpy
import os

from sys import argv
def FeatureClassGenerator(workspace, wild_card, feature_type, recursive) :
  with arcpy.EnvManager(workspace = workspace):

    dataset_list = [""]
    if recursive:
      datasets = arcpy.ListDatasets()
      dataset_list.extend(datasets)

    for dataset in dataset_list:
      featureclasses = arcpy.ListFeatureClasses(wild_card, feature_type, dataset)
      for fc in featureclasses:
        yield os.path.join(workspace, dataset, fc), fc


def Iterate_Layername(Iterate_Bluespots_gdb="Iterate_Bluespots.gdb"):  # Iterate_Layername

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False


    for T1_3_, Name in FeatureClassGenerator(Iterate_Bluespots_gdb, "", "POLYGON",
"NOT_RECURSIVE"):

        # Process: Add Field (Add Field) (management)
        T1 = arcpy.management.AddField(in_table=T1_3_, field_name="LayerName",
field_type="TEXT", field_precision=None, field_scale=None, field_length=None, field_alias="",
field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED", field_domain="")[0]

        # Process: Calculate Field (Calculate Field) (management)
        T1_2_ = arcpy.management.CalculateField(in_table=T1, field="LayerName",
expression=f"\"{Name}\"", expression_type="PYTHON3", code_block="", field_type="TEXT",
enforce_domains="NO_ENFORCE_DOMAINS")[0]

        return T1_2_

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_Layername(*argv[1:])
```

14.7.10: 7_10_Hydro_No_Block

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:09:21
"""
import arcpy
from sys import argv

def Hydro_No_Block(DEM_allmods="DEM_allmods", Flow_direction_type="D8",
Input_flow_direction_type="D8", Waterline="Waterline"):  # Hydro_No_Block

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("spatial")
    arcpy.CheckOutExtension("ImageAnalyst")
    arcpy.CheckOutExtension("3D")

    Input_false_raster_or_constant_value = 0

    # Process: Fill (Fill) (sa)
    Fill_NB = "Fill_NB"
    Fill = Fill_NB
    Fill_NB = arcpy.sa.Fill(in_surface_raster=DEM_allmods, z_limit=None)
    Fill_NB.save(Fill)


    # Process: Flow Direction (Flow Direction) (sa)
    FlowDir_NB = "FlowDir_NB"
    Flow_Direction = FlowDir_NB
    Output_drop_raster = ""
    FlowDir_NB = arcpy.sa.FlowDirection(in_surface_raster=Fill_NB, force_flow="NORMAL",
out_drop_raster=Output_drop_raster, flow_direction_type=Flow_direction_type)
    FlowDir_NB.save(Flow_Direction)


    # Process: Flow Accumulation (Flow Accumulation) (sa)
    Flow_Acc_NB =
"C:\\Users\\tskur\\OneDrive\\Dokumenter\\Master\\Masteroppgaven\\Melhus_summerjob_2021\\0
3 GIS maps and projects\\Tool_Test\\DEM_no_block.gdb\\Flow_Acc_NB"
    Flow_Accumulation = Flow_Acc_NB
    Flow_Acc_NB = arcpy.sa.FlowAccumulation(in_flow_direction_raster=FlowDir_NB,
in_weight_raster="", data_type="FLOAT", flow_direction_type=Input_flow_direction_type)
    Flow_Acc_NB.save(Flow_Accumulation)


    # Process: Con (Con) (ia)
    Con_Flow = "Con_Flow"
    Con = Con_Flow
    Con_Flow = arcpy.ia.Con(in_conditional_raster=Flow_Acc_NB,
in_true_raster_or_constant=Flow_Acc_NB,
in_false_raster_or_constant=Input_false_raster_or_constant_value, where_clause="VALUE >
25000")
    Con_Flow.save(Con)
```

```
    # Process: Int (Int) (ia)
    Output_raster_3_ = "Int_Con_Flow1"
    Int = Output_raster_3_
    Output_raster_3_ = arcpy.ia.Int(in_raster_or_constant=Con_Flow)
    Output_raster_3_.save(Int)


    # Process: Raster to Polyline (Raster to Polyline) (conversion)
    NB_Waterline = "NB_Waterline"
    with arcpy.EnvManager(outputMFlag="Disabled", outputZFlag="Disabled"):
        arcpy.conversion.RasterToPolyline(in_raster=Output_raster_3_,
out_polyline_features=NB_Waterline, background_value="ZERO", minimum_dangle_length=0,
simplify="SIMPLIFY", raster_field="VALUE")

    # Process: Add Field (Add Field) (management)
    Add_Area_Recalc = arcpy.management.AddField(in_table=NB_Waterline,
field_name="Area_Recalc", field_type="DOUBLE", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field (Calculate Field) (management)
    Area_Recalc = arcpy.management.CalculateField(in_table=Add_Area_Recalc,
field="Area_Recalc", expression="!grid_code!*0.25", expression_type="PYTHON3",
code_block="", field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Select (Select) (analysis)
    arcpy.analysis.Select(in_features=Area_Recalc, out_feature_class=Waterline,
where_clause="Area_Recalc > 25000")

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"Scratch.gdb", workspace=r"workspace.gdb"):
        Hydro_No_Block(*argv[1:])
```

14.8: 8_Iterate_Damage_Points

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:54:40
"""
import arcpy
import os

from sys import argv
def FeatureClassGenerator(workspace, wild_card, feature_type, recursive) :
  with arcpy.EnvManager(workspace = workspace):

    dataset_list = [""]
    if recursive:
      datasets = arcpy.ListDatasets()
      dataset_list.extend(datasets)

    for dataset in dataset_list:
      featureclasses = arcpy.ListFeatureClasses(wild_card, feature_type, dataset)
      for fc in featureclasses:
        yield os.path.join(workspace, dataset, fc), fc

def #  NOT  IMPLEMENTED# Function Body not implemented
def #  NOT  IMPLEMENTED# Function Body not implemented
def #  NOT  IMPLEMENTED# Function Body not implemented

def Iterate_Damage_Points(Input_Buildings="Clip_fkb_bygning_omrade.shp",
Input_Roadnet="Road_Culvert_Bridge_Erased.shp",
Input_Railnet="Clip_banenettverk_banelenke.shp", Erase_IT_gdb="Erase_IT.gdb",
Rail_Damage_gdb="Rail_Damage.gdb", Road_Damage_gdb="Road_Damage.gdb",
Build_Damage_gdb="Build_Damage.gdb", Damage_Points_Rail=fr"Rail_Damage.gdb\{Name}",
Damage_Points_Road=fr"Road_Damage.gdb\{Name}",
Damage_Points_Buildings=fr"Build_Damage.gdb\{Name}"):
# Iterate_Damage_Points

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("spatial")


    for FeatureClass, Name in FeatureClassGenerator(Erase_IT_gdb, "", "",
"NOT_RECURSIVE"):

        # Process: Intersect (3) (Intersect) (analysis)
        Name = "Waterline"
        Rail_X_Water = fr"Rail_Water_{Name}"
        arcpy.analysis.Intersect(in_features=[[Input_Railnet, ""], [FeatureClass, ""]],
out_feature_class=Rail_X_Water, join_attributes="ALL", cluster_tolerance="",
output_type="POINT")

        # Process: If Row Count Is (If Row Count Is) ()
        True_113, False_114 = #  NOT  IMPLEMENTED(in_layer_or_view=Rail_X_Water,
count_condition="IS_LESS_THAN", count=1, count_min=0, count_max=0)

        # Process: Delete Features (Delete Features) (management)
```

```
        if True_113:
            Output_Feature_Class =
arcpy.management.DeleteFeatures(in_features=Rail_X_Water)[0]

        # Process: Intersect (2) (Intersect) (analysis)
        Road_X_Water = fr"Damage_Points_Scratch.gdb\Int_Road_{Name}"
        arcpy.analysis.Intersect(in_features=[[Input_Roadnet, ""], [FeatureClass, ""]],
out_feature_class=Road_X_Water, join_attributes="ALL", cluster_tolerance="",
output_type="POINT")

        # Process: If Row Count Is (2) (If Row Count Is) ()
        True_2_, False_2_ = #  NOT  IMPLEMENTED(in_layer_or_view=Road_X_Water,
count_condition="IS_LESS_THAN", count=1, count_min=0, count_max=0)

        # Process: Delete Features (2) (Delete Features) (management)
        if True_2_:
            Output_Feature_Class_2_ =
arcpy.management.DeleteFeatures(in_features=Road_X_Water)[0]

        # Process: Buffer (Buffer) (analysis)
        Buf_Name_ = fr"Damage_Points_Scratch.gdb\Buf{Name}"
        arcpy.analysis.Buffer(in_features=FeatureClass, out_feature_class=Buf_Name_,
buffer_distance_or_field="2 Meters", line_side="FULL", line_end_type="ROUND",
dissolve_option="NONE", dissolve_field=[], method="PLANAR")

        # Process: Dissolve (Dissolve) (management)
        dis_Name_ = fr"Damage_Points_Scratch.gdb\dis{Name}"
        arcpy.management.Dissolve(in_features=Buf_Name_, out_feature_class=dis_Name_,
dissolve_field=[], statistics_fields=[], multi_part="MULTI_PART",
unsplit_lines="DISSOLVE_LINES")

        # Process: Intersect (Intersect) (analysis)
        Building_X_Water = fr"Damage_Points_Scratch.gdb\Int_Build_{Name}"
        arcpy.analysis.Intersect(in_features=[[Input_Buildings, ""], [dis_Name_, ""]],
out_feature_class=Building_X_Water, join_attributes="ALL", cluster_tolerance="",
output_type="INPUT")

        # Process: If Row Count Is (3) (If Row Count Is) ()
        True_3_, False_3_ = #  NOT  IMPLEMENTED(in_layer_or_view=Building_X_Water,
count_condition="IS_LESS_THAN", count=1, count_min=0, count_max=0)

        # Process: Delete Features (3) (Delete Features) (management)
        if True_3_:
            Output_Feature_Class_3_ =
arcpy.management.DeleteFeatures(in_features=Building_X_Water)[0]

        # Process: Copy Features (2) (Copy Features) (management)
        if False_2_:
            with arcpy.EnvManager(workspace=Road_Damage_gdb):
                arcpy.management.CopyFeatures(in_features=Road_X_Water,
out_feature_class=Damage_Poinnts_Road, config_keyword="", spatial_grid_1=None,
spatial_grid_2=None, spatial_grid_3=None)

        # Process: Copy Features (Copy Features) (management)
        Copy_Rail_Name_ = fr"Damage_Points_Scratch.gdb\Copy_Rail_{Name}"
        if False_114:
```

```
        arcpy.management.CopyFeatures(in_features=Rail_X_Water,
out_feature_class=Copy_Rail_Name_, config_keyword="", spatial_grid_1=None,
spatial_grid_2=None, spatial_grid_3=None)

    # Process: Mean Center (Mean Center) (stats)
    if False_114:
        with arcpy.EnvManager(workspace=Rail_Damage_gdb):
            arcpy.stats.MeanCenter(Input_Feature_Class=Copy_Rail_Name_,
Output_Feature_Class=Damage_Points_Rail, Weight_Field="", Case_Field="",
Dimension_Field="")

    # Process: Copy Features (3) (Copy Features) (management)
    Copy_Build_Name_ = fr"Damage_Points_Scratch.gdb\Copy_Build_{Name}"
    if False_3_:
        arcpy.management.CopyFeatures(in_features=Building_X_Water,
out_feature_class=Copy_Build_Name_, config_keyword="", spatial_grid_1=None,
spatial_grid_2=None, spatial_grid_3=None)

    # Process: Feature To Point (2) (Feature To Point) (management)
    Point_Build_Name_ = fr"Damage_Points_Scratch.gdb\Point_Build_{Name}"
    if False_3_:
        arcpy.management.FeatureToPoint(in_features=Copy_Build_Name_,
out_feature_class=Point_Build_Name_, point_location="INSIDE")

    # Process: Polygon to Raster (Polygon to Raster) (conversion)
    Ras_Name_ = fr"Damage_Points_Scratch.gdb\Ras_{Name}"
    with arcpy.EnvManager(snapRaster="DEM_allmods"):
        arcpy.conversion.PolygonToRaster(in_features=Buf_Name_, value_field="Area_Recalc",
out_rasterdataset=Ras_Name_, cell_assignment="CELL_CENTER", priority_field="NONE",
cellsize="0.20", build_rat="BUILD")

    # Process: Extract Values to Points (Extract Values to Points) (sa)
    _Name_3_ = fr"Build_Damage.gdb\{Name}"
    if False_3_:
        with arcpy.EnvManager(workspace=Build_Damage_gdb):
            arcpy.sa.ExtractValuesToPoints(in_point_features=Point_Build_Name_,
in_raster=Ras_Name_, out_point_features=_Name_3_, interpolate_values="INTERPOLATE",
add_attributes="ALL")

    # Process: Add Field (Add Field) (management)
    if False_3_:
        _Name_4_ = arcpy.management.AddField(in_table=_Name_3_,
field_name="Area_Recalc", field_type="FLOAT", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field (Calculate Field) (management)
    if False_3_:
        Damage_Points_Buildings = arcpy.management.CalculateField(in_table=_Name_4_,
field="Area_Recalc", expression="!RASTERVALU!", expression_type="PYTHON3",
code_block="", field_type="FLOAT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

if __name__ == '__main__':
  # Global Environment settings
  with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
```

```
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_Damage_Points(*argv[1:])
```

14.9: 9_ Iterate_Damageclass

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-09 19:55:43
"""
import arcpy
import os

from sys import argv
def FeatureClassGenerator(workspace, wild_card, feature_type, recursive) :
  with arcpy.EnvManager(workspace = workspace):

    dataset_list = [""]
    if recursive:
      datasets = arcpy.ListDatasets()
      dataset_list.extend(datasets)

    for dataset in dataset_list:
      featureclasses = arcpy.ListFeatureClasses(wild_card, feature_type, dataset)
      for fc in featureclasses:
        yield os.path.join(workspace, dataset, fc), fc


def Iterate_damage_class(Damage_Points="Damage.gdb"):  # Iterate_damage_class

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False


    for Input_points, Name in FeatureClassGenerator(Damage_Points, "", "POINT", 
"NOT_RECURSIVE"):

        # Process: Add Field (3) (Add Field) (management)
        Layername_added = arcpy.management.AddField(in_table=Input_points, 
field_name="LayerName", field_type="TEXT", field_precision=None, field_scale=None, 
field_length=None, field_alias="", field_is_nullable="NULLABLE", 
field_is_required="NON_REQUIRED", field_domain="")[0]

        # Process: Calculate Field (3) (Calculate Field) (management)
        Layername_calculated = arcpy.management.CalculateField(in_table=Layername_added, 
field="LayerName", expression=f"\"{Name}\"", expression_type="PYTHON3", code_block="", 
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

        # Process: Add Field (Add Field) (management)
        Acc_Area_Added = arcpy.management.AddField(in_table=Layername_calculated, 
field_name="Acc_Area", field_type="FLOAT", field_precision=None, field_scale=None, 
field_length=None, field_alias="", field_is_nullable="NULLABLE", 
field_is_required="NON_REQUIRED", field_domain="")[0]

        # Process: Calculate Field (Calculate Field) (management)
        Acc_Area_Calculated = arcpy.management.CalculateField(in_table=Acc_Area_Added, 
field="Acc_Area", expression="!Area_Recalc!", expression_type="PYTHON3", code_block="", 
field_type="FLOAT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

        # Process: Add Field (2) (Add Field) (management)
```

```
        Damageclass_added = arcpy.management.AddField(in_table=Acc_Area_Calculated,
field_name="Damage_Class", field_type="FLOAT", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="NON_REQUIRED", field_domain="")[0]

        # Process: Calculate Field (2) (Calculate Field) (management)
        Damageclass_calculated =
arcpy.management.CalculateField(in_table=Damageclass_added, field="Damage_Class",
expression="Reclass(!ACC_AREA!)", expression_type="PYTHON3", code_block="""def
Reclass(Acc_Area):
    if (Acc_Area < 50000):
        return 1
    elif Acc_Area > 50000 and Acc_Area <= 250000:
        return 2
    elif (Acc_Area > 250000 and Acc_Area <= 1000000):
        return 3
    elif (Acc_Area > 1000000 and Acc_Area <= 5000000):
        return 4
    elif (Acc_Area > 5000000 and Acc_Area <= 25000000):
        return 5
    elif (Acc_Area > 25000000):
        return 6

    """, field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

        return Acc_Area_Calculated

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"ScratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_damage_class(*argv[1:])
```

14.10: Bluespot_Mapping

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-02 11:43:11
"""
import arcpy
from sys import argv

def Bluespot_Mapping(Bluespot_Lyr_3_="Bluespot_Lyr", Rasterized_River="river_ras_ext",
FKB_Buildings="Clip_fkb_bygning_omrade.shp", DEM_All_Sinks_Filled="All_Mods",
DEM_Small_Sinks_Filled="All_Mods005"):  # Bluespot_Mapping

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False

    # Check out any necessary licenses.
    arcpy.CheckOutExtension("spatial")
    arcpy.CheckOutExtension("3D")
    arcpy.CheckOutExtension("ImageAnalyst")
    arcpy.CheckOutExtension("ImageExt")

    # Model Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"scratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        _1 = 1
        _0 = 0
        Bluespot_Main_gdb = "Bluespot_Main.gdb"

        # Process: Flow Direction (Flow Direction) (sa)
        FlowDir_as = "Wporkspace.gdb\\FlowDir_as"
        Flow_Direction = FlowDir_as
        Drop_AS = "Workspace.gdb\\Drop_AS"
        FlowDir_as = arcpy.sa.FlowDirection(in_surface_raster=DEM_All_Sinks_Filled,
force_flow="NORMAL", out_drop_raster=Drop_AS, flow_direction_type="D8")
        FlowDir_as.save(Flow_Direction)

        Drop_AS = arcpy.Raster(Drop_AS)

        # Process: Minus (Minus) (sa)
        BS_Locations = "Workspace.gdb\\BS_Locations"
        Minus = BS_Locations
        BS_Locations = arcpy.sa.Minus(in_raster_or_constant1=DEM_Small_Sinks_Filled,
in_raster_or_constant2=DEM_All_Sinks_Filled)
        BS_Locations.save(Minus)
```

```
    # Process: Con (Con) (sa)
    Con_BS = "Workspace.gdb\\Con_BS"
    Con = Con_BS
    if BS_Locations:
        Con_BS = arcpy.sa.Con(in_conditional_raster=BS_Locations, in_true_raster_or_constant=_1, in_false_raster_or_constant=_0, where_clause="")
        Con_BS.save(Con)


    # Process: Dissolve (2) (Dissolve) (management)
    Dis_Build = "Workspace.gdb\\Dis_Build"
    arcpy.management.Dissolve(in_features=FKB_Buildings, out_feature_class=Dis_Build, dissolve_field=["FID"], statistics_fields=[], multi_part="MULTI_PART", unsplit_lines="DISSOLVE_LINES")

    # Process: Polygon to Raster (Polygon to Raster) (conversion)
    Ras_Build = "Workspace.gdb\\Ras_Build"
    arcpy.conversion.PolygonToRaster(in_features=Dis_Build, value_field="OBJECTID", out_rasterdataset=Ras_Build, cell_assignment="CELL_CENTER", priority_field="NONE", cellsize="0.5", build_rat="BUILD")

    # Process: Reclassify (2) (Reclassify) (sa)
    Reclas_Build = "Workspace.gdb\\Reclas_Build"
    Reclassify_2_ = Reclas_Build
    Reclas_Build = arcpy.sa.Reclassify(in_raster=Ras_Build, reclass_field="VALUE", remap="0 1;NODATA 0", missing_values="DATA")
    Reclas_Build.save(Reclassify_2_)


    # Process: Reclassify (Reclassify) (sa)
    BS_Val_1 = "Workspace.gdb\\BS_Val_1"
    Reclassify = BS_Val_1
    if BS_Locations and Reclas_Build:
        BS_Val_1 = arcpy.sa.Reclassify(in_raster=Con_BS, reclass_field="VALUE", remap="0 0;1 1;NODATA 0", missing_values="DATA")
        BS_Val_1.save(Reclassify)


    # Process: Minus (2) (Minus) (sa)
    BS_Min_Build = "Workspace.gdb\\BS_Min_Build"
    Minus_2_ = BS_Min_Build
    if BS_Locations and Reclas_Build:
        BS_Min_Build = arcpy.sa.Minus(in_raster_or_constant1=BS_Val_1, in_raster_or_constant2=Reclas_Build)
        BS_Min_Build.save(Minus_2_)


    # Process: Reclassify (3) (Reclassify) (sa)
    Reclas_Build_3_ = "Workspace.gdb\\Reclas_Build"
    Reclassify_3_ = Reclas_Build_3_
    if BS_Locations and Reclas_Build:
        Reclas_Build_3_ = arcpy.sa.Reclassify(in_raster=BS_Min_Build, reclass_field="VALUE", remap="-1 NODATA;0 NODATA;1 1", missing_values="DATA")
        Reclas_Build_3_.save(Reclassify_3_)
```

```
    # Process: Reclassify (5) (Reclassify) (sa)
    water_ras_0_1 = "Workspace.gdb\\water_ras_0_1"
    Reclassify_5_ = water_ras_0_1
    water_ras_0_1 = arcpy.sa.Reclassify(in_raster=Rasterized_River, reclass_field="VALUE",
remap="1 1;NODATA 0", missing_values="DATA")
    water_ras_0_1.save(Reclassify_5_)


    # Process: Minus (3) (Minus) (sa)
    BS_Min_Wat_Build = "Workspace.gdb\\BS_Min_Wat_Build"
    Minus_3_ = BS_Min_Wat_Build
    if BS_Locations and Reclas_Build:
        BS_Min_Wat_Build = arcpy.sa.Minus(in_raster_or_constant1=Reclas_Build_3_,
in_raster_or_constant2=water_ras_0_1)
        BS_Min_Wat_Build.save(Minus_3_)


    # Process: Reclassify (4) (Reclassify) (sa)
    BS_0_1 = "Workspace.gdb\\BS_0_1"
    Reclassify_4_ = BS_0_1
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        BS_0_1 = arcpy.sa.Reclassify(in_raster=BS_Min_Wat_Build, reclass_field="VALUE",
remap="0 NODATA;1 1", missing_values="DATA")
        BS_0_1.save(Reclassify_4_)


    # Process: Region Group (Region Group) (sa)
    BS_Region = "Workspace.gdb\\BS_Region"
    Region_Group = BS_Region
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        BS_Region = arcpy.sa.RegionGroup(in_raster=BS_0_1, number_neighbors="EIGHT",
zone_connectivity="WITHIN", add_link="ADD_LINK", excluded_value=0)
        BS_Region.save(Region_Group)


    # Process: Add Field (Add Field) (management)
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        BlueSpotRegion_Volume = arcpy.management.AddField(in_table=BS_Region,
field_name="Volume", field_type="DOUBLE", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="REQUIRED", field_domain="")[0]

    # Process: Zonal Statistics as Table (Zonal Statistics as Table) (sa)
    SumUnfilledDepth = "Workspace.gdb\\SumUnfilledDepth"
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        arcpy.sa.ZonalStatisticsAsTable(in_zone_data=BlueSpotRegion_Volume,
zone_field="VALUE", in_value_raster=DEM_Small_Sinks_Filled, out_table=SumUnfilledDepth,
ignore_nodata="DATA", statistics_type="SUM",
process_as_multidimensional="CURRENT_SLICE", percentile_values=90,
percentile_interpolation_type="AUTO_DETECT")
        .save(Zonal_Statistics_as_Table)


    # Process: Add Field (4) (Add Field) (management)
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
```

```
        SumUnfilledDepth_4_ = arcpy.management.AddField(in_table=SumUnfilledDepth,
field_name="Sum_SS", field_type="LONG", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field (4) (Calculate Field) (management)
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        SumUnfilledDepth_3_ =
arcpy.management.CalculateField(in_table=SumUnfilledDepth_4_, field="Sum_SS",
expression="!SUM!", expression_type="PYTHON3", code_block="", field_type="TEXT",
enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Join Field (2) (Join Field) (management)
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        BlueSpotRegion_SS = arcpy.management.JoinField(in_data=BlueSpotRegion_Volume,
in_field="VALUE", join_table=SumUnfilledDepth_3_, join_field="VALUE", fields=["Sum_SS"])[0]

    # Process: Zonal Statistics as Table (2) (Zonal Statistics as Table) (sa)
    SumFilledDepth =
"C:\\Users\\tskur\\OneDrive\\Dokumenter\\Master\\Masteroppgaven\\Melhus_summerjob_2021\\0
3 GIS maps and projects\\Tool_Test\\Bluespot_Scratch.gdb\\SumFilledDepth"
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        arcpy.sa.ZonalStatisticsAsTable(in_zone_data=BlueSpotRegion_Volume,
zone_field="VALUE", in_value_raster=DEM_All_Sinks_Filled, out_table=SumFilledDepth,
ignore_nodata="DATA", statistics_type="SUM",
process_as_multidimensional="CURRENT_SLICE", percentile_values=90,
percentile_interpolation_type="AUTO_DETECT")
        .save(Zonal_Statistics_as_Table_2_)


    # Process: Add Field (5) (Add Field) (management)
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        SumFilledDepth_3_ = arcpy.management.AddField(in_table=SumFilledDepth,
field_name="Sum_AS", field_type="LONG", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field (5) (Calculate Field) (management)
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        SumFilledDepth_2_ = arcpy.management.CalculateField(in_table=SumFilledDepth_3_,
field="Sum_AS", expression="!SUM!", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Join Field (Join Field) (management)
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        BlueSpotRegion_SS_AS = arcpy.management.JoinField(in_data=BlueSpotRegion_SS,
in_field="Value", join_table=SumFilledDepth_2_, join_field="Value", fields=["Sum_AS"])[0]

    # Process: Calculate Field (Calculate Field) (management)
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        BlueSpotRegion_VolCalc =
arcpy.management.CalculateField(in_table=BlueSpotRegion_SS_AS, field="Volume",
expression="(0.5 * 0.5) * (!Sum_AS!-!Sum_SS!)", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Watershed (Watershed) (sa)
```

```
    Wtrshd_SF_Flow = "Workspace.gdb\\Wtrshd_SF_Flow"
    Watershed = Wtrshd_SF_Flow
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        Wtrshd_SF_Flow = arcpy.sa.Watershed(in_flow_direction_raster=FlowDir_as,
in_pour_point_data=BlueSpotRegion_VolCalc, pour_point_field="Volume")
        Wtrshd_SF_Flow.save(Watershed)


    # Process: Build Raster Attribute Table (Build Raster Attribute Table) (management)
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        Updated_Input_Raster =
arcpy.management.BuildRasterAttributeTable(in_raster=Wtrshd_SF_Flow, overwrite="NONE",
convert_colormap="NONE")[0]

    # Process: Add Field (2) (Add Field) (management)
    if BS_Locations and BS_Min_Wat_Build and Reclas_Build:
        Watershed_SF_Flow_2_ =
arcpy.management.AddField(in_table=Updated_Input_Raster, field_name="Watershed_Area",
field_type="FLOAT", field_precision=None, field_scale=None, field_length=None, field_alias="",
field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field (2) (Calculate Field) (management)
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and Reclas_Build:
        Wtrshd_SF_Flow_2_ =
arcpy.management.CalculateField(in_table=Watershed_SF_Flow_2_, field="Watershed_Area",
expression="(0.5 * 0.5) * !Count!", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Join Field (3) (Join Field) (management)
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and Reclas_Build
and Wtrshd_SF_Flow_2_:
        BlueSpotRegion_2_ = arcpy.management.JoinField(in_data=BlueSpotRegion_VolCalc,
in_field="Value", join_table=Wtrshd_SF_Flow_2_, join_field="Value", fields=[])[0]

    # Process: Add Field (3) (Add Field) (management)
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and Reclas_Build
and Wtrshd_SF_Flow_2_:
        BlueSpotRegion_3_ = arcpy.management.AddField(in_table=BlueSpotRegion_2_,
field_name="FillUp", field_type="FLOAT", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field (3) (Calculate Field) (management)
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and Reclas_Build
and Wtrshd_SF_Flow_2_:
        BlueSpotRegion_4_ = arcpy.management.CalculateField(in_table=BlueSpotRegion_3_,
field="FillUp", expression="(!Volume!*1000)/!Watershed_Area!", expression_type="PYTHON3",
code_block="", field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Lookup (Lookup) (sa)
    Lookup_FillUP = "Workspace.gdb\\Lookup_FillUP"
    Lookup = Lookup_FillUP
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and Reclas_Build
and Wtrshd_SF_Flow_2_:
        Lookup_FillUP = arcpy.sa.Lookup(in_raster=BlueSpotRegion_4_, lookup_field="FillUp")
        Lookup_FillUP.save(Lookup)
```

```
    # Process: Int (Int) (sa)
    Int_FillUP = "Workspace.gdb\\Int_FillUP"
    Int = Int_FillUP
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and Reclas_Build
and Wrtshd_SF_Flow_2_:
        Int_FillUP = arcpy.sa.Int(in_raster_or_constant=Lookup_FillUP)
        Int_FillUP.save(Int)


    # Process: Raster to Polygon (Raster to Polygon) (conversion)
    FillUP_Poly = "Workspace.gdb\\FillUP_Poly"
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and Reclas_Build
and Wrtshd_SF_Flow_2_:
        with arcpy.EnvManager(outputMFlag="Disabled", outputZFlag="Disabled"):
            arcpy.conversion.RasterToPolygon(in_raster=Int_FillUP,
out_polygon_features=FillUP_Poly, simplify="SIMPLIFY", raster_field="VALUE",
create_multipart_features="SINGLE_OUTER_PART", max_vertices_per_feature=None)

    # Process: Dissolve (Dissolve) (management)
    FillUp_Poly_Dis = "Workspace.gdb\\FillUp_Poly_Dis"
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and Reclas_Build
and Wrtshd_SF_Flow_2_:
        arcpy.management.Dissolve(in_features=FillUP_Poly,
out_feature_class=FillUp_Poly_Dis, dissolve_field=["GRIDCODE"], statistics_fields=[],
multi_part="MULTI_PART", unsplit_lines="DISSOLVE_LINES")

    # Process: Join Field (4) (Join Field) (management)
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and
FillUp_Poly_Dis and Reclas_Build and Wrtshd_SF_Flow_2_:
        Ras2Poly_BlueSpot_Dissolve_2_ =
arcpy.management.JoinField(in_data=FillUp_Poly_Dis, in_field="gridcode",
join_table=BlueSpotRegion_4_, join_field="Value", fields=["FillUp", "Volume"])[0]

    # Process: Make Feature Layer (Make Feature Layer) (management)
    Bluespot_Lyr = "Bluespot_Lyr"
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and
FillUp_Poly_Dis and Reclas_Build and Wrtshd_SF_Flow_2_:
        arcpy.management.MakeFeatureLayer(in_features=Ras2Poly_BlueSpot_Dissolve_2_,
out_layer=Bluespot_Lyr, where_clause="", workspace=Bluespot_Main_gdb,
field_info="GRIDCODE GRIDCODE VISIBLE NONE;FillUp FillUp VISIBLE NONE;Volume
Volume VISIBLE NONE")

    # Process: Add Field (6) (Add Field) (management)
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and
FillUp_Poly_Dis and Reclas_Build and Wrtshd_SF_Flow_2_:
        Bluespot_Lyr_2_ = arcpy.management.AddField(in_table=Bluespot_Lyr,
field_name="Fill_Up_20", field_type="FLOAT", field_precision=None, field_scale=None,
field_length=None, field_alias="", field_is_nullable="NULLABLE",
field_is_required="NON_REQUIRED", field_domain="")[0]

    # Process: Calculate Field (6) (Calculate Field) (management)
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and
FillUp_Poly_Dis and Reclas_Build and Wrtshd_SF_Flow_2_:
```

```
        Bluespot_Lyr_3_ = arcpy.management.CalculateField(in_table=Bluespot_Lyr_2_,
field="Fill_Up_20", expression="!FillUp! + 20", expression_type="PYTHON3", code_block="",
field_type="TEXT", enforce_domains="NO_ENFORCE_DOMAINS")[0]

    # Process: Raster to Polygon (2) (Raster to Polygon) (conversion)
    Output_polygon_features = ".Workspace.gdb\\RasterT_BS_Regi1"
    if BS_Locations and BS_Min_Wat_Build and BlueSpotRegion_VolCalc and Reclas_Build
and Wtrshd_SF_Flow_2_:
        arcpy.conversion.RasterToPolygon(in_raster=BlueSpotRegion_4_,
out_polygon_features=Output_polygon_features, simplify="SIMPLIFY", raster_field="VALUE",
create_multipart_features="SINGLE_OUTER_PART", max_vertices_per_feature=None)

if __name__ == '__main__':
    Bluespot_Mapping(*argv[1:])
```

14.11: Iterate_Bluespot_ID

```
# -*- coding: utf-8 -*-
"""
Generated by ArcGIS ModelBuilder on : 2022-05-02 12:03:48
"""
import arcpy
import os

from sys import argv
def FeatureClassGenerator(workspace, wild_card, feature_type, recursive) :
  with arcpy.EnvManager(workspace = workspace):

    dataset_list = [""]
    if recursive:
      datasets = arcpy.ListDatasets()
      dataset_list.extend(datasets)

    for dataset in dataset_list:
      featureclasses = arcpy.ListFeatureClasses(wild_card, feature_type, dataset)
      for fc in featureclasses:
        yield os.path.join(workspace, dataset, fc), fc

def Iterate_Bluespot_ID(Input_Floodpaths="Iterate_Bluespots.gdb\BS_{Name}",
BS_Point_Name_=fr"Iterate_Bluespots.gdb\BS_Point_{Name}",
Clip_fkb_bygning_omrade_shp="Bluespot_Main.gdb\\Depressions"):
# Iterate_Bluespot_ID

    # To allow overwriting outputs change overwriteOutput option to True.
    arcpy.env.overwriteOutput = False


    for FeatureClass, Name in FeatureClassGenerator(Input_Floodpaths, "", "",
"NOT_RECURSIVE"):

        # Process: Select Layer By Location (Select Layer By Location) (management)
        Layer_With_Selection, Output_Layer_Names, Count =
arcpy.management.SelectLayerByLocation(in_layer=[Input_Bluespots],
overlap_type="INTERSECT", select_features=FeatureClass, search_distance="",
selection_type="NEW_SELECTION", invert_spatial_relationship="NOT_INVERT")

        # Process: Select (Select) (analysis)
        Big_BS_Name_ = fr"Iterate_Bluespots.gdb\Big_BS_{Name}"
        arcpy.analysis.Select(in_features=Layer_With_Selection,
out_feature_class=Big_BS_Name_, where_clause="Shape_Length <= Shape_Area")

        # Process: Select Layer By Location (2) (Select Layer By Location) (management)
        Layer_With_Selection_2_, Output_Layer_Names_2_, Count_2_ =
arcpy.management.SelectLayerByLocation(in_layer=[Clip_fkb_bygning_omrade_shp,
Big_BS_Name_], overlap_type="INTERSECT", select_features=Big_BS_Name_,
search_distance="", selection_type="NEW_SELECTION",
invert_spatial_relationship="NOT_INVERT")

        # Process: Feature To Point (Feature To Point) (management)
```

```
        arcpy.management.FeatureToPoint(in_features=Layer_With_Selection_2_,
out_feature_class=BS_Point_Name_, point_location="CENTROID")

        # Process: Copy Features (Copy Features) (management)
        arcpy.management.CopyFeatures(in_features=Big_BS_Name_,
out_feature_class=BS_Name_, config_keyword="", spatial_grid_1=None, spatial_grid_2=None,
spatial_grid_3=None)

        return Name

if __name__ == '__main__':
    # Global Environment settings
    with
arcpy.EnvManager(outputCoordinateSystem="PROJCS["ETRS_1989_UTM_Zone_32N",GEOG
CS["GCS_ETRS_1989",DATUM["D_ETRS_1989",SPHEROID["GRS_1980",6378137.0,298.2572
22101]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Trans
verse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",9.0],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latit
ude_Of_Origin",0.0],UNIT["Meter",1.0]],VERTCS["NN2000 height",VDATUM["Norway Normal
Null 2000"],PARAMETER["Vertical_Shift",0.0],PARAMETER["Direction",1.0],UNIT["metre",1.0]]",
scratchWorkspace=r"scratchWorkspace.gdb", workspace=r"Workspace.gdb"):
        Iterate_Bluespot_ID(*argv[1:])
```

Torgeir Skurdal

Developing semi-automated GIS methods for indicating downstream flood damage potential of critical points

NTNU

Norwegian University of
Science and Technology