Detecting land subsidence using multicriteria decision analysis and machine learning in Skøyen, Oslo

Ronald Mathew Baysa Payabyab

Thesis submitted for the degree of
Master in Geosciences: Geomorphology and Geomatics
60 credits

Department of Geosciences
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2022

Detecting land subsidence using multicriteria decision analysis and machine learning in Skøyen, Oslo

Ronald Mathew Baysa Payabyab

Detecting land subsidence using multicriteria decision analysis and machine learning in Skøyen, Oslo

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Acknowledgements

I would like to express my appreciation to my main supervisor postdoctor Désirée Treichler. She has given me precise and great feedback on the thesis and has consistently allowed this paper to be my own work, but at the same time made sure that I was on the right track.

I would also like to thank professor François Renard and GIS-developer Sean Stonerock from Multiconsult for giving me inspiration to the topic. Sean has especially been quite helpful towards me during the hard times of the pandemic.

Finally, I must express my very profound gratitude to my parents and to my best friend for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Summary/Sammendrag

This master thesis is a quantitative study aiming to map the risk of land subsidence in Skøyen at the ward of Ullern, which is an urbanized area in Oslo. Land subsidence is a type of natural hazard that cause gradual insulation on the subsurface and is a common problem in many cities. Land subsidence can lead to significant financial costs in society and potential damage in buildings. The purpose of this study is therefore to check the possibility of mapping the risk of subsidence based upon 7 criteria. The thesis is divided in two different method. The first method is a multi-criterion analysis that is based on human decision making, prioritizing 7 different criteria that are potentially linked with land subsidence. The second method also use the same 7 criteria, but use machine learning methods such as neural networks and XGBoost to map subsidence based upon finding a pattern in the 7 criteria by using InSAR as a feeding algorithm. The dataset comes from many different sources such as the Norwegian Water and Energy Directorate, the Norwegian Geological Survey, the Mapping Authority and NIBIO. The resulting subsidence maps tend to vary. While multi-criterion analysis and neural networks have classified several areas as high risk for land subsidence with multiple outliers, XGBoost appears to have a lot more defined areas for high-risk subsidence. This should however be taken into more consideration as the data set is quite limited in the study, and there are potentially more criteria that are more suitable than the original 7 criteria chosen.

Denne masteroppgaven er en kvantitativ studie som skal kartlegge risikograden for subsidens på Skøyen i Ullern bydel som er et urbanisert område i Oslo. Subsidens er en form for naturfare som skaper gradvis innsynkning i overflaten og har vært en kjent problem i flere storbyer som følger av bevegelser i grunnvann. Dette kan føre til store økonomiske kostnader i samfunnet og potensielle skader i bebygde områder. Formålet med oppgaven er derfor å sjekke muligheten for å kartlegge risikograden for subsidens på bakgrunn av syv ulike kriterier. Oppgaven bruker to ulike metoder. Den ene metoden er en multikriterieanalyse som er basert på å lage et risikokart for subsidens etter menneskelig beslutningsgrunnlag gjennom vekting og prioritering av 7 ulike kriterier som kan bidra til subsidens. Den andre metoden bruker også de samme 7 kriteriene, men bruker istedenfor maskinlæringsmetodene nevrale nettverk og XGBoost for å kartlegge subsidens gjennom å finne et mønster i de 7 kriteriene ved å bruke InSAR-data som grunnlag. Datagrunnlaget kommer fra flere ulike kilder og tar bakgrunn i kartlag fra blant annet Norges vassdrags- og energidirektorat, Norsk Geologisk Undersøkelse, Kartverket og NIBIO. Resultatene har en tendens til å variere. Mens multikriterieanalysen og nevrale nettverk har flere områder klassifisert som høyrisiko for subsidens med mer støy i risikokartet, så har maskinlæringsmetoden XGBoost stort sett klart å definere tydelige mønstre på utsatte områder i lik grad med InSAR. Det må likevel tas høyde for at datagrunnlaget er begrenset i studiet, og at andre kriterier utenom de 7 kriteriene kan være mer aktuelle.

Contents

Abstract

Slow mass movements such as land subsidence have been quite a big issue with regards to infrastructural projects, which have caused delays in road planning or collapse of buildings in urban areas. The dominant explanation for these incidents is due to hydrogeological conditions such as pumping of groundwater. Previous research has mostly focused on creating susceptibility maps for natural hazards in general rather than specifically for land subsidence. While machine learning methods have been more commonly used, there are very few studies that have attempted to use multicriteria decision analysis (MCDA) to map land subsidence. The intention of this project is to create land subsidence susceptibility map using MCDA and machine learning algorithms such as neural networks and XGBoost to check how differently they perform. The same dataset will be applied for both MCDA and machine learning, containing 7 criteria. Additionally InSAR-data will be used as a feeding algorithm for the machine learning algorithms. The results of this project shows that maps generated from MCDA strongly deviate from the InSAR-reference map. Regression analyses have shown an R^2-score below 0.01 for MCDA even for the maps created even when adjusting the weights or removing some criteria during the sensitivity analysis. On the contrary, both neural networks and XGBoost gained better accuracy with 0.6468 and 0.7128 respectively after tuning their hyperparameters. The neural network did however show some similar pattern as the MCDA when visualizing the sensitivity maps, and had a quite poor AUC-score of 0.65 compared to XGBoost with an AUC-score of 0.82. The findings of this project suggest that the choice of criteria could be improved by choosing factors or criteria more specifically directed at hydrogeological conditions which are limited in the study area. Testing a larger set of hyperparameters is also relevant for the machine learning algorithms to avoid bias. Nevertheless, the machine learning algorithms appear to be more suitable for predicting land subsidence due to the subjective nature of MCDA.

1 Introduction

Detecting slow mass movements are important to avoid disruption of engineering projects as selecting suitable areas are crucial for the development of infrastructures such as roads, tunnels and skyscrapers. It is a long-term investment providing economic growth and speed up city development. Inadequate infrastructure could therefore be detrimental to the society if a project is hindered or delayed (Beiler and Treat, 2014). The rapid expansion of larger and cost-efficient projects has thus raised issues regarding site planning vulnerable to natural hazards such as land subsidence or other slow mass movements. For instances, the Norwegian Road Authorities have experienced and documented gradual risks of land subsidence related to infrastructural projects in Bjørvika, Oslo which has caused precautions when planning newer infrastructures nearby ports or setting up a new tramline in the area (Vegvesen, 2016). In a worst case scenario, such instances of land subsidence may result in the formation of sinkholes, potholes, settlement of structures, and subsidence of roads. Gradual subsidence can potentially make roads collapse into subsurface cavities due to loss of bearing capacity in the ground. This can for example take place when limestone is dissoluted by fluid flow in the surface, which creates voids underground causing the subsidence. Groundwater often plays a central role for causing subsidence due to exploitation. This a global phenomena and has occurred in larger cities around the world such as Mexico City, Mexico (Sowter et al., 2016), Bangkok, Thailand (Zeitoun and Wakshal, 2013), Shanghai, China and Tianjin, China (Yuan et al., 2020).

The study of susceptibility mapping of natural hazards has been the focus of major scientific research, engineering study, and practices throughout the world (Pourghasemi and Saravi, 2019). Susceptibility mapping for land subsidence provides crucial information in spatial mapping for areas vulnerable to different types of natural hazards such as landslide and land subsidence. In recent times, land susceptibility maps have been generated using geological, geomorphological, topographical and hydrological data; which also are the main factors of land subsidence (Hakim et al., 2020a). There are various methods for generating a land subsidence susceptibility map. GIS-MCDA poses as one of the more common methods creating susceptibility maps (Ghorbanzadeh et al., 2018). In brief, GIS-MCDA is a general framework for supporting complex decision-making situations with many and, quite often, conflicting objectives that stakeholders and/or decision-makers value differently. Decision makers in GIS-MCDA procedures make use of a variety of spatial data, taken into account of the expert's opinions where the preferences are set according to rules established in the decision-making process (Malczewski, 1999a). Although GIS-MCDA often yields accurate and satisfactory results, it has been demonstrated that there is a degree of uncertainty associated with the method (Feizizadeh et al., 2013). This could for example occur during the weighting process where the individual's preferences might be inconsistent. Furthermore, since the process of MCDA are often based on the experts' opinions, the results from the susceptibility maps highly depend on each individual's preferences, which could result in multiple susceptibility maps with different outcomes. Analytical hierarchy process (AHP) is therefore sometimes applied in MCDA to organize complex decisions. In short, the goal of the AHP is to provide a rational framework for a needed decision by quantifying its criteria and options, and relating those elements for an overall goal.

Alternatively, risk maps for land subsidence can also be created by using methods from machine learning. The application of artificial intelligence for spatial processing and analysis have recently been under development within the GIS-community. Many machine learning models such as logistic regression, random forests, support vector machines and neural networks has been applied in geographic information systems to create maps for different purposes from predicting susceptibility of gully erosion (Shahabi et al., 2019) to even determining the most profitable location to establish a hotel (Yang, 2015). The main reasons why machine learning models are getting more widely used in the fields of spatial analysis and engineering are the remarkable performance, flexibility and accuracy in modeling and predicting phenomena whereas knowledge based models depend highly on expert's judgment and are associated with uncertainty. On the other hand, rigid systems such machine learning may not always function as the designers have intended since there are aspects of human decision-making that are difficult to automate (Sui, 1994). Therefore an automated strategy that can simulate the experts' learning and reasoning process would be highly highly desirable.

1.1 Goals

This thesis intends to create risk maps of slow mass movements, mainly focusing on land subsidence. Two different approaches will be made. The first approach is to use expert-based decision-making to create a risk map. The same 7 criteria will be used later as input-data when using machine learning techniques to predict subsidence on InSAR-data as output. Its purpose is to observe and compare how a risk map based upon human decision making might differentiate with a more technical approach using machine learning algorithms. Both risk maps generated from these techniques will be further compared with a reference map from InSAR-data gathered by Norwegian Geological Survey (NGU).

The thesis will first introduce the geological background of the study area, and briefly present some of the infrastructure in the urban areas of Oslo. Furthermore the concepts of land subsidence such as major causes, processes and some examples where subsidence have previously appeared will be introduced and. Previous research of land subsidence mapping using multicriteria decision analysis (MCDA) and/or machine learning methods will also be presented. The thesis will then also discuss common practices of monitoring and predicting land subsidence with emphasis to interferometric synthetic aperture radar (InSAR) and its advantages and disadvantages. After presenting the fundamentals of land subsidence and some general practices of subsidence monitoring, a theoretical background of the desired methods to generate the susceptibility maps will be discussed and examples of earlier practices of using such methods will be mentioned as well. Background information of the seven criteria chosen to predict land subsidence in the analysis and their connection to land subsidence are to be introduced at the end of the background chapter.

By using different approaches, the research questions revolves around if expert-based mapping can be used to detect the susceptibility of land subsidence, and maybe even be potentially used in geotechnical planning. The study area will therefore be applied on a heavily trafficked area in the intersection between Drammensveien and Bygdøy allé at Ullern district in Oslo, Norway. This will also include the area of Skøyen where a new subway station are planned to be built. The study area will be briefly presented later on in the background chapter.

Overall, the goal is to create subsidence risk maps by using three different methods, namely: GIS-MCDA with analytical hierarchy process (AHP) and pairwise comparisons, neural networks and XGBoost, which is a machine learning method based on an ensemble of decision trees. The first method will be GIS-MCDA, while the next two methods focus more on predicting land subsidence with neural networks and XGBoost as machine learning techniques. The approach of GIS-MCDA will be based on a spatial multicriteria analysis in ArcGIS Pro. Multiple criteria such as geological, topographic and hydrological conditions will be taken into into consideration based on recommendations and data from Norwegian Road Authorities (Statens Vegvesen), Bane Nor and NGU. All data types used to generate the 7 chosen criteria are stored in a common geodatabase in order to process and create the final product of each criterion. The processing of each individual criterion will be described further in-depth in the methodology-chapter. Additionally, the criteria will also be chosen with respect to causes of groundwater exploitation in tunnels. Each criterion will be ranked based on the algorithms of AHP where the relative importance of each individual criterion is compared with the other criterion. The result of the analysis will represent the weight of each criterion. A sensitivity analysis involving removal of one criterion at a time are made at the end of the GIS-MCDA analysis to test the robustness of the result along with tweaking the weights added to each criterion. To briefly summarize, the aim of this thesis is to check how differently GIS-MCDA, neural networks, and XGBoost predicts suitable areas for infrastructural projects based on the risk of land subsidence in the area, and the discussion in the thesis will thus emphasize on how credible these susceptibility maps based upon the known conditions of the area and the observed land subsidence. In other words, it is of great interest in the thesis to observe how human knowledge-based decision making compares to machine learning when evaluating suitable places to build new infrastructure.

The next two methods of neural networks and XGBoost do have the same kind of practices where the seven criteria chosen from the the MCDA are used to predict land subsidence based on observed InSAR-data of the study area. The InSAR-data will represent the target value and the machine learning models will attempt to look after patterns in the criteria to predict subsidence/uplift. Multiple hyperparameters of those methods will be tuned to get as optimal result as possible. Unlike MCDA, those two methods will require some processing and file conversion of the data set in order to run the prediction in Python-format. The results/prediction will then be converted back into ArcGIS Pro.

Both methods of GIS-MCDA and machine learning are well known to create susceptibility maps and have been applied on multiple occasions earlier with varying results and different types of criteria around the globe. On the other hand, similar studies has not currently taken place in this particular area of Norway. It is also worth to mention that most articles study those two methods separately rather than comparing them to each other.

2 Background

2.1 Study site

This project will focus on Skøyen and Bygdøy alle from Ullern district in Oslo. The area has recently been of interest by Oslo municipality as it has been decided to further develop the area as a hub for public transportation with regards to building a new metro station as a part of Fornebubanen, which is related to the extension of the metro system in Oslo and parts of Bærum municipality. Additionally the municipality has also set plans to build at least 5000 new houses in the area (*Områderegulering for Skøyen*, 2021). The technical reports from *Områderegulering for Skøyen* (2021) do however further mention that Skøyen is particularly an area where the loose materials in the subsurface contains lot of clay, which makes the area vulnerable to subsidence-related damage in buildings, and there is a high probability for causing more subsidence if the groundwater-level changes in the area.

2.2 General geology

The Oslo-area was completely covered by large ice masses during the ice age for around 20 000 years ago (Klemsdal, 2002). The underlying land surface were pressed down by the enormous ice masses during the late-glacial and post-glacial periods to the extent that the global sea level was lowered. This has resulted in glaciers delivering debris ranging in size from clay-sized rock flour to boulders to the front of the glacier/sea boundary as unconsolidated deposits. Coarse materials accumulated close to the glacier snout while silt and clay particles were further transported towards the sea as they were flocculated into aggregates and settled to the sea floor. Melt water from glaciers continued to deliver fine particles to the sea forming deltas at river mouths where the flocculation process continued as the glaciers slowly retreated. When the ice sheets retreated even farther, local land surface rebounded faster than the sea level rose due to ice melt, lifting more of the old sea bottom above sea level. The surface got exposed to subaerial processes which in turn formed weathered crust in the surface zone. Due to delays in the Earth's crust, it will take some time to retain an equilibrium after the ice masses has melted. Thus, most of Oslo still experience a post-glacial rebound, i.e the landmasses are still slowly heaving upwards. This means that the majority of the study area are now dry land that were once ocean floor during the ice age.

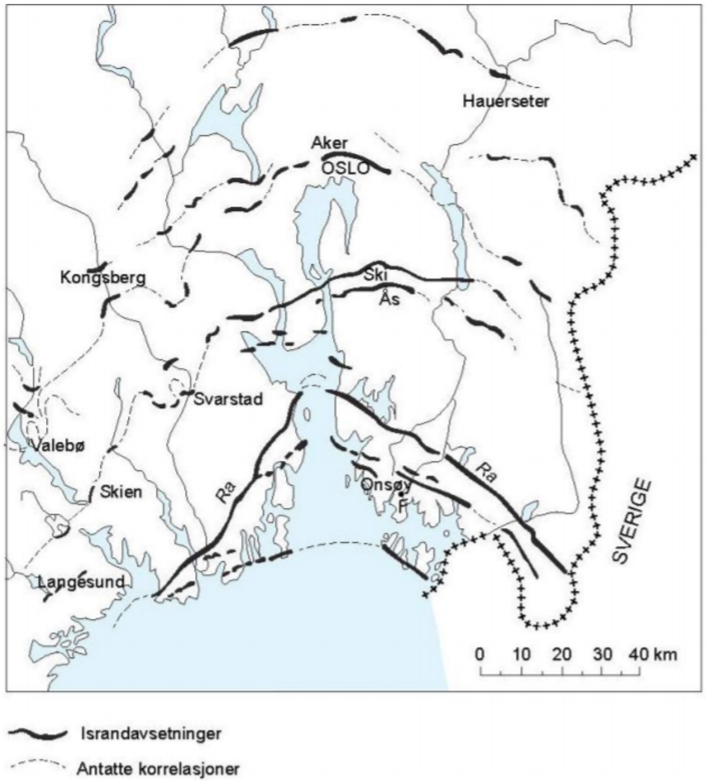


Figure 1: An illustration over unconsolidated material deposits within the Oslo fjord. Source: (Klemsdal, 2002)

As figure 1 illustrates, the area lies in the Oslo rift which were formed by stretching, dilution and cracking of the crust in carbon and perm (Nordgulen et al., 1998). Most of the study area are covered by 540-415 million cambrosilurian bedrock. Although the Oslo rift is affected by the processes during carbonperm, the structures are also to a larger extent influenced by the development under the caledonian orogeny when the area got exposed to compressive forces from northwest. The faults in the area have affected the bedrock during latecaledonian and permian time where the cambrosilurian bedrock folded. This has caused several decimeters of thick clay zones along these faults, but the bending folds have been replaced with cracks and faults causing weakness zones parallel to the strike (Nordgulen et al., 1998). The weakness zones can additionally be partially open and create canals where water can infiltrate the bedrock.

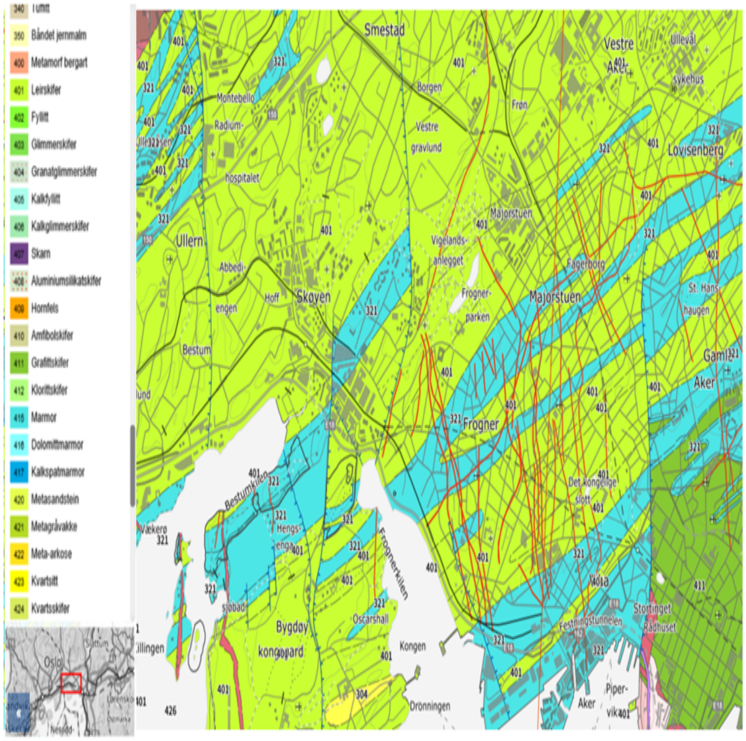


Figure 2: The study area. Most of the study area are based in Skøyen and partially the intersection between Frogner and Bygdøy. (Lat 59.919258, Long 10.683225) Source: (NGU, n.d.)

The study area of Skøyen itself is dominated by limestone nodules (light blue) with layers of slate (light green) as shown from figure 2. Most of the area are highly urbanized with few exposed bedrock on the surface as illustrated in figure 2. A larger North-South fault can be found further east of Skøyen Station, but there are otherwise no other known fault zones in the area. Otherwise, larger parts of the area consists of marine deposits, which potentially can contain quick clay. The thickness of loose materials tend to variate by area from outcrops to more than 50 meters of depth to bedrock. Multiple samples from the municipality indicate loose materials to a larger extent consist of very wet to medium solid clay with little to no sensitive clay. The case of subsidence is not only particular to Skøyen, but also other urbanized areas in Oslo as well such as the Oslo Central railstation and the Barcode area (Eriksson et al., 2021). Significant subsidence deformation of more than 5 mm/year within Oslo has also been observed during the 90s (Lauknes et al., 2006). The study area itself do also show similar activites in figure 3, especially around the trains stations and along the roads.

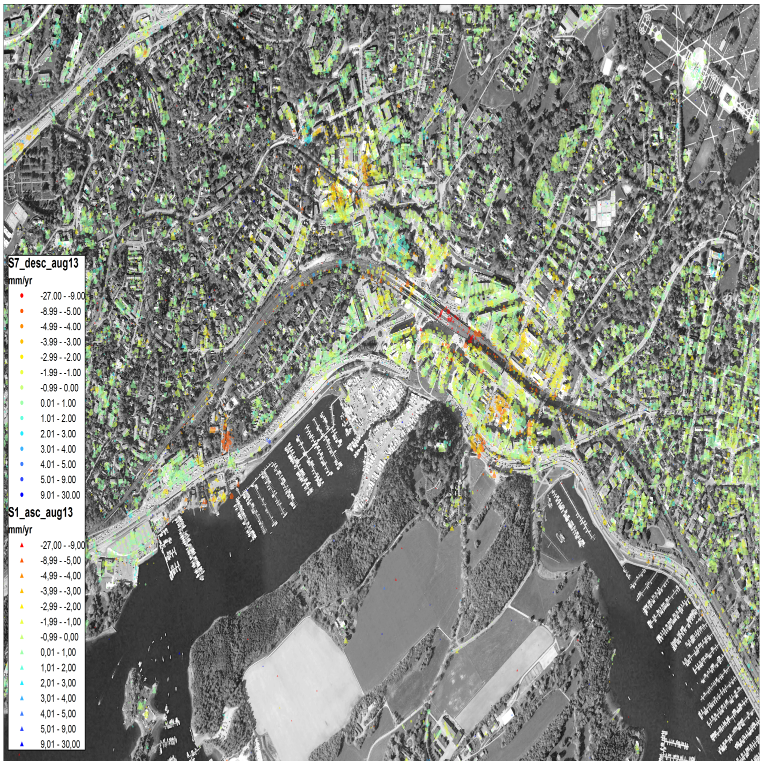


Figure 3: An example of subsidence distribution in the Skøyen-area using InSAR. Some parts of Skøyen has a subsidence rate as high as 27 mm/year, particularly within the train station. Source:(Romsenter, n.d.)

2.3 Theoretical background of land subsidence

Land subsidence is a gradual settling or sudden sinking of the Earth's surface due to removal or displacement of subsurface earth materials, and it is a problem that occurs around the world. For instances, many countries have suffered costly from the damages of land subsidence, especially from major city areas such as Tokyo, Japan; Shanghai, China; and the California region in USA. Causes of land subsidence include compaction of aquifer-systems, drainage of organic soils, underground compaction, natural compaction, sinkholes and thawing of permafrost (Galloway et al., 1999). Exploitation of groundwater appears to be one of the more common occurrences for land subsidence as more than 80 % of identified cases in the US stems from underground water exploitation. Likewise in Shanghai, China, the exploitation in the deeper aquifer during the 1990s caused an aggravation of land subsidence due delayed drainage in a watertight layer (Cui, 2018).

Any kinds of fluid in pore spaces or fractures in rock are under pressure due to the weight of overlying rock. Fluid pressure decreases when withdrawn from the subsurface, removing the support and potentially cause a collapse. Water in the form of groundwater and petroleum from natural gas and oil are the most common fluids that gets withdrawn for human use, and thus subsidence related to withdrawal of fluids are often manmade.

Subsidence can also occur by hydro-compaction. Hydro-compaction is the shrinkage of clay due to the removal or drying of water the clay minerals have absorbed. In relatively dry areas where the ground contains a mix of sand, silt and clay, the dry silt or clay particles behaves like a cement that holds the sand particles in an open matrix with air spaces between them. Water infiltration through intense rainfall, irrigation or broken pipes leaves the open soil structure vulnerable since the sudden influx of water dissolves silt and clay, making the air spaces between the sand particles collapse, and causes the affected area to sink as illustrated in figure 4.

Processes that lowers the water table can also potentially trigger land subsidence. The water table defines the level below in which the ground is saturated with water. Anything beneath the water table, i.e the saturated zone, are fully saturated with water, and the pore spaces are occupied. It is possible that the water level are exposed at the surface such as in streams, lakes and swamps. The level of the water table can be changed either by recharge; where water is introduced in the groundwater system such as when intense rainfall infiltrates the surface or by discharge where groundwater leaves the system through surface bodies of water, springs and wells.

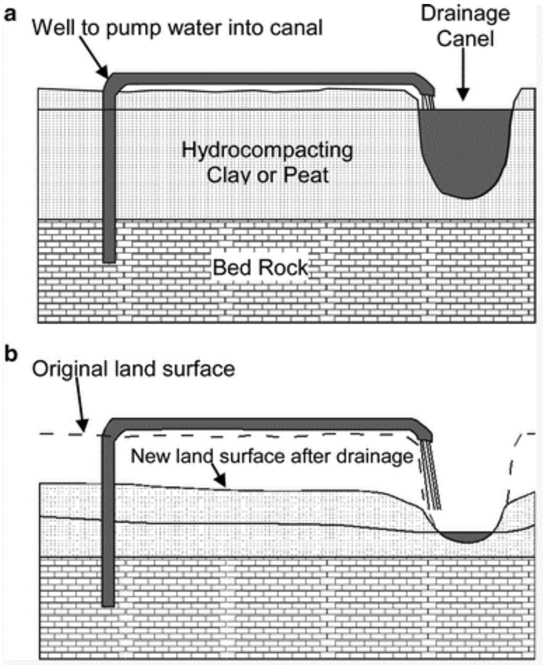


Figure 4: An illustration of the stages of subsidence due to hydro-compaction. Pumping groundwater leads to gradual subsidence, lowering the depth to water table. Source: Zeitoun and Wakshal (2013)

Houses and other buildings are also contributing to land settlements during a consolidation process. Along with hydro-compaction, consolidation applies stress to the soil reducing the bulk volume. The water from the saturated soil will be squeezed out, and when stress is removed from the soil, it will regain some of the volume it has lost in the process. Occurrences of natural hazards such as landslides and land subsidence are expected to rise since climate change affect the stability of natural and engineered slope as a result of increasing rainfall intensity and higher temperatures (Gariano and Guzzetti, 2016). Additionally, groundwater exploitation from porous sediments and fine-grained material is also proven to enhance land subsidence globally (Huang et al., 2012). Along with climatic parameters, the demand for water supply in plains and cities, leads to excessive extraction of groundwater which in turn increase the dangers of land subsidence in these areas. The sudden and gradual movement in clay acquifers has the potential to cause demolition of buildings and rising pipes from the ground (Pourghasemi and Saravi, 2019). Groundwater activity is also an issue in some places in Norway as well. For instances, planned site projects are potentially threatened by groundwater leakage in the tunnels due to the area lying below groundwater level (Braathen, 2018) in the municipality of Fredrikstad. It was assumed that the groundwater potential varies significantly from the area, but the most susceptible area contain little to no bedrock in the subsurface with an abundance of soft sediments.

Groundwater exploitation in city areas is however not the sole cause of subsidence as man-made infrastructure and large-scale engineering construction also contributed to ground displacements such as building loads, pit excavation and dewatering, shield tunnelling and tunnel leakage (Yang and Gong, 2010). Subsidence in urbanized areas is in fact an issue since leakage of human-made sewer pipes or water mains might cause erosion of underground soil and/or rock leading to sudden collapse. Cities built on unconsolidated sediments such as clay, silt, peat and sand are particularly vulnerable to subsidence. This applies to cities located in delta areas, where rivers fare into the sea, along floodplains adjacent to rivers, and in coastal marsh lands (Zeitoun and Wakshal, 2013). Subsidence is a natural process in such settings. Sediments deposited by rivers and oceans gets buried, and the weights of the overlying, newly-deposited sediments, compacts the sediment and the material subsides. There are several reasons building cities in such areas contributes to subsidence. One of them being that the construction of buildings and streets adds weight to the region and further adds more stress to the soil underneath. For instances, there have been some instances of large and unforeseen subsidence events around Oslo Central Station relating to building projects in the neighborhood area of Bjørvika (Vegvesen, 2016). It was concluded in a technical report from Norwegian Road Authorities that adding fill material from asphalting made the area more vulnerable to subsidence along with the reduction of pore pressure due to causing drainage effects when performing drill samples. Areas often have to be drained to be occupied, resulting in lowering the groundwater table and leads to hydro-compaction and consolidation. Another man-made structures that could potentially are dams and levees built to prevent or control flooding. Usually sedimentation from floods helps replenish the sediments that subside, decreasing the overall rate of subsidence. When the sediment supply is suddenly cut off, the replenishment does not occur and the subsidence rate enhances.

2.4 InSAR and the practices of predicting ground subsidence

Monitoring elevation-change are fundamental to track land subsidence. Measurements by continuous GPS (CGPS), campaign global positioning system (GPS), spirit-leveling, and use of interferometric synthetic aperture radar (InSAR) are among the most common methods for measuring land subsidence. Compaction of aquifer-systems can also be measured by extensometers, which is an instrument that are often used to measure deformation of materials under stress.

InSAR provides more cost-efficient and higher resolution subsidence than those provided by benchmark releveling, extensometers and GPS surveys, providing

DEMs with meter accuracy and terrain deformation with millimeter accuracy. In principle, DEMs and deformation patterns can be estimated in a very dense grid at low cost compared with any other traditional methods. The rationale behind a multi-image approach to estimate surface displacements stems from the interferogram. Interferograms are maps of relative ground-surface change that are constructed from InSAR data. It requires at least two images taken at intervals in time to determine if there has been any shift in land surface levels. If the ground has moved away (subsidence) or towards (uplift) the satellite between the times of the two SAR-images, a different portion of the wavelength is reflected back to the satellite resulting in a phase shift. That is when surface movement has been detected, measured and recorded via an interferogram. Figure 5 shows the acquisition process. Two radar images are collected from different times from similar points in space that can be compared against each other. Any movements are measured and portrayed as a picture.

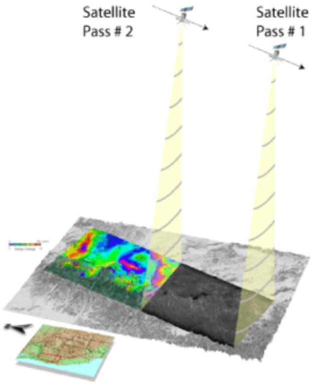


Figure 5: An example of how InSAR is recorded. A satellite passes over an area and records data about it. Two or nore passes are required to create InSAR images. Source: (USGS, n.d.a)

A pixel value in a SAR-image changes phase depending on the relative position of the satellite scatterer, temporal changes of the target and atmospheric variations. If we have $K + 1$ SAR images of the same area as the reference master acquisition m, the phase difference of pixel $\mathbf{x} = \left[\begin{smallmatrix}\xi \\ \mu\end{smallmatrix}\right]$ (where ξ and μ representing the azimuth and slant range coordinates respectively) between the generic slave image k with respect to the master image will be represented as:

$$\begin{aligned}\psi_k(\mathbf{x}) =& \psi_k(\mathbf{x}) - \psi_m(\mathbf{x}) \\ =& \frac{4\pi}{\lambda}[r_k(\mathbf{x}) - r_m(\mathbf{x})] + [\sigma_k(\mathbf{x}) - \sigma_m(\mathbf{x})] \\ &+ [a_k(\mathbf{x}) - a_m(\mathbf{x})]\end{aligned} \tag{1}$$

where r is the distance of the satellite target, σ is the scatterer reflectivity phase, and a is the atmospheric phase contribution Ferretti et al. (2001). In repeat-pass interferometry, it is also possible to represent r_k as:

$$r_k = r_m + \triangle r_k + \triangle \mu_k \tag{2}$$

$\triangle r_k$ is the range of variation due to different satellite positions, and $\triangle \mu_k$ is the possible target motion in the direction of the satellite line-of-sight that appears during the time interval between the two acquisitions.

Although InSAR often gives precise measurement of subsidence, use of traditional methods such as extensometers have been utilized to constrain the measurements from InSAR. For instances, Buckley et al. (2003) used extensometers to measure ground subsidence around local areas in Houston, Texas giving a root-mean squared error of less than 2.5 mm. It was proven that radar interferometry can benefit from complementary use of ground-based subsidence measurements since measurements from InSAR could be disturbed from atmospheric artefacts, decorrelation or heavy vegetation. Temporal decorrelation makes InSAR-measurements unfeasible over vegetated areas and where electromagnetic profiles and/or the positions of the scatterers change with time within the resolution cell. Furthermore, geometrical decorrelation also limits the number of image pairs suitable for interferometric applications and prevents one from fully exploring the data sets available.

2.5 GIS-MCDA and application of machine learning for generating susceptibility maps

2.5.1 Basic concepts of GIS-MCDA

In the past decades, several studies of subsidence susceptibility and zonation has been carried out across the globe with a diverse range of methods included such as the applications of neural network (Lee et al., 2012), interferometric synthetic aperture radar (Dehghani et al., 2014), and AHP (Ghorbanzadeh et al., 2018). Since it takes time for land subsidence to instigate, where it often takes a couple of years before deformation begin to appear at the surface, a geospatial approach to land subsidence susceptibility mapping (LSSM) could therefore be an effective technique to improve the understanding of susceptibility to such kind of hazards. LSSM attempts to discover areas with high susceptibility of land subsidence and also reduce the severe adverse effects of this type of natural hazards. By implementing geospatial analysis such as GIS-MCDA in the process, a prediction of potential land subsidence occurrences can be zoned and mapped, which can contribute to reduce negative effects during site planning. GIS-MCDA provides a powerful method for overlaying maps that can be sufficiently used for LSSM due GIS' efficiency of data collection, analysis and validation (Gaspar et al., 2004).

In order to start to start a multicriteria decision analysis (MCDA), it is first important to have a clearly defined goal in the analysis. This could for instances be: Where is the most ideal place to live in Norway as a student? Which areas are prone to landslide? Where is to most suitable place to build a shopping mall? Such objectives are often quite difficult to answer as it often depends on what factors the decision-maker wants prioritize. The decision-maker can be an individual, a group of individuals, or an organization (Malczewski, 1999a). It is therefore important to have decision alternatives evaluated on the basis of a set of criteria, that includes both attributes and objectives. Both individual criterion and a set of criteria should posses properties to adequately represent the nature of the decision situation. For instances, if the criterion housing price is included in the first question, it must be measurable and comprehensible. According to Malczewski and Rinner (2015b), the properties of a set of criteria must fit the following description:

1. Complete (Cover all aspects of a decision problem)
2. Operational (Meaningfully used in the analysis)
3. Decomposable (The set of criteria can be broken into parts and simplified)
4. Non-redundant (Avoid problem of double counting)
5. Minimal (The number of criteria should be as small as possible)

Additionally a criterion can be spatially explicit or implicit. A spatially explicit criteria involves spatial characteristics of decision alternatives such as shape and size, while a spatially implicit criterion indicates that data are needed to compute the level of achievement of criterion such as distance to roads and equity of income distribution.

Furthermore, a criterion is a generic term including both the concepts of objective and attribute (Malczewski, 1999a). An objective is a statement about the desired state of a system under consideration, while an attribute is a property of an element of a real-world geographical system.

Each criterion needs to be given a certain weight that indicates how important the criterion is in the decision-making process. It is worth to mention that the methods of assessing criterion weights are context-related, but there are some desired properties that the criterion weights should have. That is, the criterion weights $w_1, w_2 ... w_k$ are typically assumed to meet the conditions: $0 \leq w_k \leq 1$ and $\sum_{k=1}^{n} = 1$. The greater the weight, the more important is the criterion in the overall value. There are also some additional ways to improve the weighting the criteria such as the analytical hierarchy process and pairwise comparison. This will be discussed in more detail in the chapter of methodology.

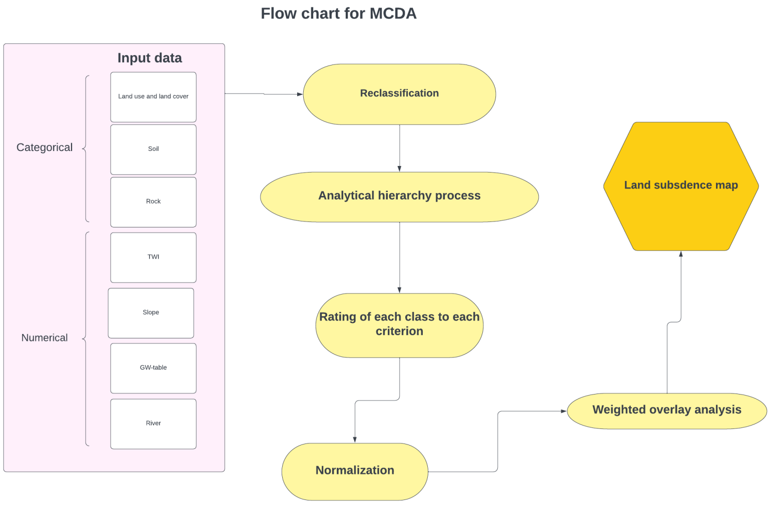


Figure 6: An example flow chart for the project. All 7 criteria go through several processes before the fina product.

Figure 6 shows a simplified process of the MCDA workflow. Not all criteria for the project contain numerical values. Thus, the criteria containing categorical values such as the "land use and land cover" contain classes such as "Urban fabric" and "Arable" land must be reclassified into numeric values before the overlay analysis. The step of analytical hierarchy process could come at any step, but must be before weighted overlay analysis as the process itself define the exact weights for each criterion based on individual pairwise comparison between each criterion. During the reclassification, it is important to rate each class to each individual criterion on a scale from 0 to 10 in order to create a normalized map. After the normalization, the weights that resulted from AHP are applied in the weighted overlay analysis as described from the last paragraph.

2.5.2 Studies of MCDA vs machine learning methods

There is a wide variety of implementing GIS multicriteria decision analysis (MCDA) to evaluate ground susceptibility and site selection. De Luca et al. (2012) constructed a value-focused MCDA approach for developing three hypothetical corridors in order to create a high-speed train track between Palermo, Italy and Berlin, Germany. Skilodimou et al. (2019) on the other hand used MCDA to delineate suitable areas for urban planning with emphasis on past landslide and flood events. Similarly, Dai et al. (2001) made a geo-environmental evaluation for urban land-use planning in northwest China. By applying topography, surface and bedrock geology, groundwater condition and historic geologic hazards as factors in the analysis, suitability maps for different types of land-use and projects such as high-rise buildings and natural conservation was created. Using a more traditional approach for geo-environmental planning by manual mapping in the area would otherwise be more laborious, time-consuming and less cost-efficient.

One of the critical aspects of applying spatial multicriteria analysis is that it involves evaluation of geographic events based on chosen criteria and the decision maker's preferences with respect to the given set of evaluation criteria (Malczewski, 1999b). There are two considerations that are crucial for successfully applying MCDA with spatial data. One of them is the GIS' capabilities of data acquisition, storage, retrieval, manipulation and analysis. When attempting to acquire and process them to gather information for decision making, the complexity of the problem may exceed the decision maker's cognitive abilities. Although there are no standard limitations for selecting the number of criteria included in the analysis, the efficiency of MCDA-application depends on the number of considered criteria. Studies conducted by Saaty and Ozdemir (2003) implies that the number of elements for making judgements through MCDA-techniques, such as the analytical hierarchy process (AHP), should not be more than seven. The reasoning is founded in the consistency of information derived from the relations among the elements. As the number of elements surpass seven, the increase in inconsistencies will become too small for the mind to single out the element. Another factor is the MCDA capabilities for aggregating geographical data and decision maker's preferences into unidimensional values of alternative decisions. This is also related to the number of criteria and the performance of sensitivity analysis. Karlsson et al. (2017) for instances made a susceptibility analysis of natural hazards with regards to road planning in Sweden. The results indicates that the use of spatial MCDA needs further development regarding decision rules and criteria. This is due to the uncertainties subjected in the susceptibility assessment. The experts involved in the project made more or less subjective, and partially inconsistent judgement of the criteria used to determine natural hazards. In spite of the current limitations of MCDA, it is still a useful method for spatially identifying potentially natural hazard susceptible areas, but it should be followed up with in-situ investigations and more detailed modeling for use as decision support.

Since the processes of natural hazards are quite complex in nature and can be affected by multiple factors, it requires huge amount of data to represent each conditional criteria properly. Thus, data mining techniques has gradually been implemented in decision-making process for GIS-environments. In general, data mining is described as an analysis tool for a large amount of data sets. Data mining techniques such as ANN (Artificial Neural Networks), SVM (Support Vector Machine), LR (Logistic regression) etc., are considered to be efficient ways for analyzing large chunks of data and for producing landslide susceptibility, hazard and risk maps (Yilmaz and Ercanoglu, 2019). For instances, Lee et al. (2004) utilized ANN to weight criteria of landslide susceptibility in Yongin, South-Korea. The area was suffering to landslide damage due to heavy rainfall in 1991. Determining the weights of the landslide criteria through ANN were therefore investigated in order to compare the performance between data mining and the ordinary statistical approach in the multicriteria analysis. Lee et al. (2012) has also applied ANN for measuring ground subsidence susceptibility (GSS) in areas nearby an abandoned coal mine. Their procedure was based on eight physiographic and structural factors as input data. The back-propagation algorithm was then applied to calculate the weights between the input and hidden layers and between the hidden and output layers by using a MATLAB software package. Another example of using data mining in GIS is the application of logistic regression by using equal proportions of the binary pixels 1 ("landslide") and 0 ("no landslide") to represent landslide susceptibility. Ayalew and Yamagishi (2005) got the results by arranging all classes of their chosen parameters according to corresponding landslide densities. The regression was performed among independent variables.

Traditional methods of overlay and multicriteria evaluation are capable to replicate and perhaps even be replaced by neural networks. The concept behind the ANN-model is derived from the simulation of the human brain. Each basic computational unit in the network represents a neuron, which performs a simple weighted summation and nonlinear mapping (Zhou and Civco, 1996). An ANN-model are composed of highly interconnected processing elements arranged in a way that are analogous the neurons. Furthermore, ANN also contains a sequence of layers and slabs with full or random connection between successive layers. The connection is usually described as input layer, where data/criteria are presented to the networks, and output layers holds the response of the given input to the network. Otherwise, the layers distinct from input and output are considered as hidden layers. These types of intermediate layers usually captures low-level features such as the weighted average of several inputs of criteria or the presence of simple patterns within one criterion. By using such approaches for geospatial analysis, ANN-modeling is capable to automatically determine weights of each factor considered for the MCDA based on representative training sets in an objective and nonlinear manner in comparison to overlay analysis which subjectively assigns weights. This was proven in a case study by Sui (1994) where he used six criteria as input for ANN in order to make a suitability map for site planning. Land parcels were first extracted as overlay and by polygon selection capabilities. Some of the data were then used as training sets to find the easiest path to solve a particular problem. The analysis also includes a learning rate that controls how much of the errors between network output and desired output propagates back to preceding nodes in the network. The artificial neural networks learns by adjusting weights between the neurons in response to the errors between the actual output values and the target output values. The generated output will then produce the thematic suitability map. Overall, this technique provides a promising approach to handle uncertainties in input data. Problems occurring from traditional cartographic modeling techniques such as weight determination and inability to handle noisy and/or missing data can be resolved through learning process and hidden layers in ANN.

2.6 Machine learning algorithms used in this thesis

2.6.1 Neural networks

In its simplistic form, a neural network is either a two-stage regression or classification model, that is typically represented as a network diagram. It consist of a large number of interconnected processing elements called neurons which are working together to resolve a specific problem.

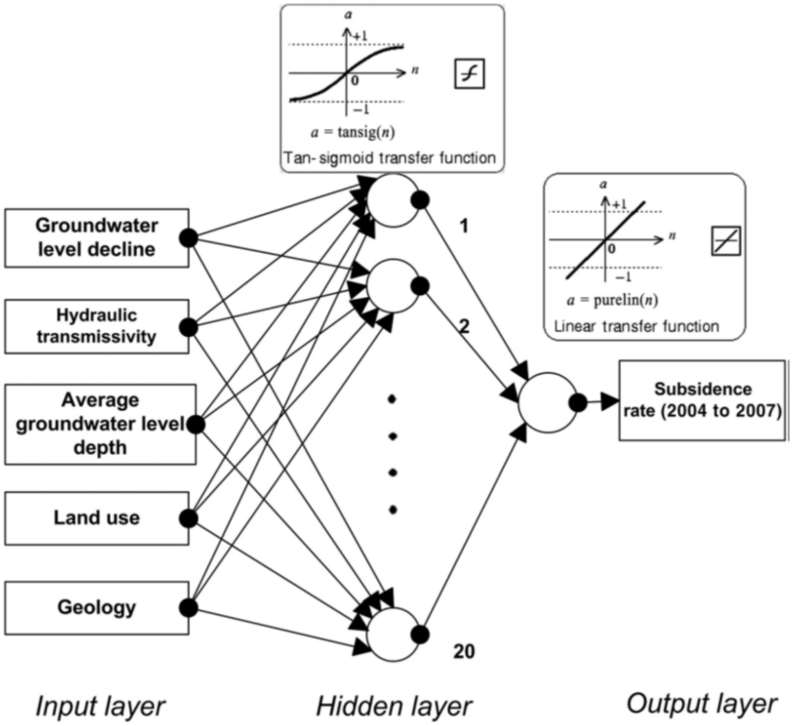


Figure 7: An example of the neural network architecture for predicting land subsidence using 5 criteria that potentially could influence the risk. Source: Bagheri et al. (2019a)

Figure 5 shows an example of how a neural network can be used to predict land subsidence. The network is built upon three types of layers, namely: an input layer, a hidden layer, and an output layer. The input layer in this example shows the criteria and data that has been used to predict land subsidence. The last two layers process the input they receive from the input layer by multiplying each input by a corresponding weight, summing the product, and then processing the sum using a non-linear transfer function (activation function) to produce the result. The neurons in these models act like nodes where data and computations flow. They receive one or more input signals, and these signals can either stem from a raw data set or from neurons positioned at a previous layer of the neural net. Each connection between neurons are represented by weights. Once a neuron receives its inputs from the neurons of the preceding layer of the model, it sums up each signal multiplied by its corresponding weight and passes them further to an activation function. The general aim of the neural networks is to tune the weights of every connection to properly estimate an output or reaction with a given a set of inputs. In this project, we will look closer to the Feed Forward Neural Network (FFNN) algorithm. The paragraphs below will dive a little deeper into the algorithms used to create the neural network algorithms for this project.

The activation functions The biological neuron is simulated in a neural by an activation function. In a classification task, an activation function has to have the switch-on characteristic, which mean that the output should change state. This simulates the "turning on" of a biological neuron. In a regression problem on the other hand, the final layer of the neural network will have one neuron and the value it returns is a continuous numerical value. The expression "z" represent the input data.

A common activation function for both classification and regression problem is the sigmoid function:

$$f(z) = \frac{1}{1+e^{-z}} \tag{3}$$

The sigmoid function moves from 0 to 1 when the input z is greater than a certain value. It is not a stepwise function, and the output does not change instantaneously. This activation function is often utilized due to its smooth behavior where the function goes from 0 to 1 in a well-behaved way. In a binary case, the output class depends on the value of the activation function used in the last layer. If the $f(z^l) \geq 0.5$, the output will be defined as class type 0, and class type 1 otherwise.

Another activation function is the rectified linear unit (ReLU):

$$f(z) = \begin{cases} 0, & \text{for } z < 0. \\ z, & \text{for } z \geq 0. \end{cases}$$

ReLU is linear for all positive values and 0 for all negative values. This makes it computationally cheap compared to sigmoid as the model takes less time to train or run. Additionally, it also converge faster since it does not have the vanishing gradients, and it is sparsely activated. Sparsity results in concise models often leads to better predictive powers and less overfitting/noise. Unlike sigmoid, ReLU has no upper limit of the value it can take.

Another variant of ReLU is the leaky ReLU. Leaky ReLU has a small slope for negative values, instead of 0 altogether. Leaky ReLU may for instances have $y = 0.01z$ when $z < 0$. An advantage of leaky ReLU over ordinary ReLU is that we can worry less about the initialization of the neural network. In the case of ReLU, it is possible to end up with a neural network that never learns if the neurons are not activated in the beginning. The network may have many "dead" ReLU, e.g ReLU always gives values under 0, without even noticing.

In a classification problem, one might also consider softmax as an activation function for the output layer. Softmax assigns decimal probabilities to each class in a multi-class problem. Those decimal probabilities must add up to 1.0. This additional constraint helps training converge more quickly. Softmax is implemented through a neural network layer just before the output layer. The softmax layer must have the same number of nodes as the output layer. The equation of softmax is as follows:

$$f(z)_i = \frac{\exp z_i}{\sum_{j=1}^{n} \exp z_j} \tag{4}$$

The equation above shows the expression of softmax. z represents the inpt vector which consist of n elements for n classes. z_i is the i-th element and can be any real value. The activation function calculates a probability for every possible class. This form of activation function is usually computationally cheap when the number of classes are small but becomes more expensive when the number of classes increases (Géron, 2017). For binary classification, using softmax should give the same result as sigmoid, since softmax is a generalization of sigmoid for a larger number of classes.

The basic notations Before diving in to the algorithm, it is essential to explain some basic notation first. For the upcoming equations, each weight is identified with $w_{ij}^{(l)}$. The i represents the node number of the connection in layer $l + 1$, while j refers to the node number of the connection in the l-th layer.For instances, the connection between node 1 in layer 1 and node 2 in layer 2 will gain the weight notation $w_{21}^{(1)}$. The notation of the bias weight is considered as $b_i^{(l)}$ where i is the node number just like the weight notation. Note that the bias generally has no input value since it is not a true node with an activation function. Both the values $w_{ji}^{(}1)$ and $b_i^{(l)}$ must be calculated during the training phase of the neural network. The ANN also consist of an output notation $h_j^{(l)}$, where j is the node number in layer l of the network.

Feed Forward propagation In a feed forward process, a new variable $z_i^{(l)}$ is introduced, which represents the sum of inputs into node i of layer l, with the bias term included. For example, in the case of the first node in layer 2, z is equal to:

$$z_1^{(2)} = w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2 + w_{13}^{(1)}x_3 + b_1^{(1)} = \sum_{j=1}^{n} w_{ij}^{(1)}x_i + b_i^{(1)} \tag{5}$$

It is possible to forward propagate the calculation through any given number of layers in the neural network. The generalized term is then:

$$z^{(l+1)} = W^{(l)}h^{(l)} + b^{(l)} \tag{6}$$

where

$$h^{(l+1)} = f(z^{(l+1)}) \tag{7}$$

The output of layer l becomes the input of layer $l+1$. $h^{(1)}$ is simply considered as input layer x, while the last layer is the output layer.

Cost function It is always important iteratively minimize the error of the output the neural network by varying weights and gradient descent. This makes it possible to prevent overfitting of the data, and the optimization revolves around minimizing the cost function. In neural networks , the equivalent cost function of a single training pair (x^z, y^z) is expressed as:

$$\begin{aligned} J(w,b,x,y) &= \frac{1}{2}\left|\left|y^z - h^{n_1}(x^z)\right|\right|^2 \\ &= \frac{1}{2}\left|\left|y^z - y_{pred}(x^z)\right|\right|^2 \end{aligned} \tag{8}$$

The equation above shows the cost function of the z-th training sample where h^{n_1} represents the final layer of the neural network, the output layer in other words. The error is represented in the L^2 norm, which is a common way of representing the error of machine learning algorithms. Instead of taking the absolute value between the predicted value and the actual value, the square of error is calculated.

The downside of using a quadratic function in equation (8) is that there's a risk for getting a learning slowdown. This means that when the neuron's output is getting closer to 1, the curve flattens, and the derivative of the cost function gets very small, which in turn leads to $\frac{\partial C}{\partial w}$ and $\frac{\partial C}{\partial b}$ being very small as well (Nielsen, 2015). To address the issue of learning slowdown, the cost function can be replaced with a different cost function called the cross-entropy function:

$$J(w,b,x,y) = -\sum[y\ln(h^{n_1}(x^z)) + (1-y)\ln(1-h^{n_1}(x^z))] \tag{9}$$

The cross-entropy is positive and tends toward zero as the neuron gets better at computing the desired output, y, for all training inputs. If we replace $h^{n_1}(x^z)$ with $\sigma(z)$, the partial derivative of the cost function can be described as:

$$\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum \frac{\sigma'(z) x_j}{\sigma(z)(1-\sigma(z))} (\sigma(z) - y) \tag{10}$$

The terms $\sigma'(z)$ and $\sigma(z)(1-\sigma(z))$ cancel in the equation and gets simplified to become:

$$\frac{\partial C}{\partial w_j} = \frac{1}{n} \sum x_j (\sigma(z) - y). \tag{11}$$

where x_j are the input variables and n is the total number of items in the training data. The equation above indicates that the learning rate of the weights are controlled by the error in output. The larger the error, the faster the neuron will learn. In particular, it avoids learning slowdown caused by the term $\sigma'(z)$ as it is cancelled out using cross-entropy. $\sigma(z)$ in this equation is the probability of an input data point being class type 1, while the term $1 - \sigma(z)$ represents the probability of class 0. In a multiclass case, the softmax function are used. One-hot encoding will be utilized, and in a binary case, the classes may for instances be represented as $(1, 0) = 0$ and $(0, 1) = 1$.

Backpropagation In equation (5) as discussed for the feed forward algorithm, we have defined the simple foundational equation of the neural network by using three layers as an example. One of the purposes of the backpropagation process is to find out how much a change in the weight $w_{12}^{(2)}$ has on the cost function J in order to evaluate the chain function:

$$\frac{\partial J}{\partial w_{12}^{(2)}} = \frac{\partial J}{\partial h_1^{(3)}} \frac{\partial h_1^{(3)}}{\partial z_1^{(2)}} \frac{\partial z_1^{(2)}}{\partial w_{12}^{(2)}} \tag{12}$$

For the weights connecting to the output layer, the cost function can be directly calculated by comparing the output layer to the training data set (Nielsen, 2015). The output of the hidden nodes does on the other hand have no such direct reference. They are connected to the cost function only through mediating weights and other layers of nodes. This is where backpropagation will do its work. The term that needs to propagate back through the network is denoted as $\delta_i^{(n_1)}$. This term represents the neural network's ultimate connection to the cost function, and it contributes through the weight $w_{ij}^{(2)}$. The output layer δ is communicated through the hidden node by the weight of the connection. The hidden layer δ can in a vectorized form be described as:

$$\delta_j^l = \sum_{i=1} (\delta_1^{(l+1)} w_{ij}^{(l)}) f'(z_j)^l \tag{13}$$

It is then possible to calculate the gradient descent for the weights and biases:

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b, x, y) = \delta_i^{(l)} h_j^{(l-1)} \tag{14}$$

$$\frac{\partial}{\partial b_j^{(l)}} J(W, b, x, y) = \delta_i^{(l)} \tag{15}$$

for all layers l. The derivatives in the activation function enters in equation (13), behaving like a weight to error used in the gradients. This might cause a vanishing gradient when using the sigmoid function. If the weighted inputs to the of the sigmoid function is substantially large such that $|z| >> 0$, then its derivatives will be close to zero. This will, as a consequence, wind up with very small gradients for the weights and biases, and it will take a lot of iterations to optimize the weights and biases for the stochastic gradient descent. This will become less of an issue when using ReLU as an activation function. Since the output of its derivative is either 1 or 0, the vanishing gradient will not cause too much of an issue since not all of the weighted inputs become less than zero.

Error metrics Measuring the performance by estimating the error metric is important to check how well a model predicts data. In a classification problem this could for instances be the accuracy score:

$$Accuracy = \frac{\sum_{i=1}^{n} I(t_i = y_i)}{n} \tag{16}$$

The accuracy is measured by the number of correctly guessed targets t_i divided by the n total of targets. I is the indicator which will give the value of 1.0 if $t_i = y_i$ and 0 otherwise in a binary case. This method is easy to implement, but could be biased if the data set is biased as well.

An alternative error metric method for machine learning is the AUC-score, which stands for the area under the ROC-curve. AUC provides an aggregate measure of performance across all positive classification threshold. An interpretation of AUC is that a random positive example are ranked more highly than a random negative in a machine learning model.

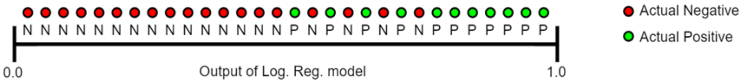

N N N N N N N N N N N N N N N P N P N P N P N P P P P P P P
0.0
Output of Log. Reg. model
1.0
Actual Negative
Actual Positive

Figure 8: A sample illustration of predicted data using logistic regression. Source: (GoogleDev, 2020)

In the figure above, AUC represents the probability that a random positive (i.e green) example is placed to the right of random negative (red) example. AUC ranges in value from 0 to 1. If a model predicted all samples wrong, then AUC would have a value of 0.0, and a model with 100% correct prediction would have an AUC-score of 1.0. The reason AUC is included as an error metric is because it is scale-invariant. It measures how good predictions are ranked rather than their absolute values. Furthermore, AUC also measures the quality of the predictions of a model regardless of what classification threshold is chosen. However, scale invariancy is not always desirable because it is sometimes necessary to have well calibrated probability outputs, which AUC would not take into considerations compared to accuracy.

AUC and accuracy will thus be used as error metrics when performing the land subsidence predictions for both neural networks and XGBoost. These would be the key factors when deciding the most optimal parameters to use.

2.6.2 XGBoost

Just like neural networks, XGBoost use training data x_i to predict a target variable y_i. XGBoost stands for "extreme gradient boosting", and it is an open-source implementation of the gradient boosted trees algorithm, which inspired from decision trees. Decision trees create a model that predicts a label by evaluating a tree of if-then-else statements and true/false questions. The goal of a decision tree is to estimate the minimum number of questions needed to assess the probability of making a correct prediction of category in a classification problem.

The difference between an ordinary decision tree and XGBoost is that XGBoost uses an ensemble of decision trees. The ensemble consist of a set of classification and regression trees.

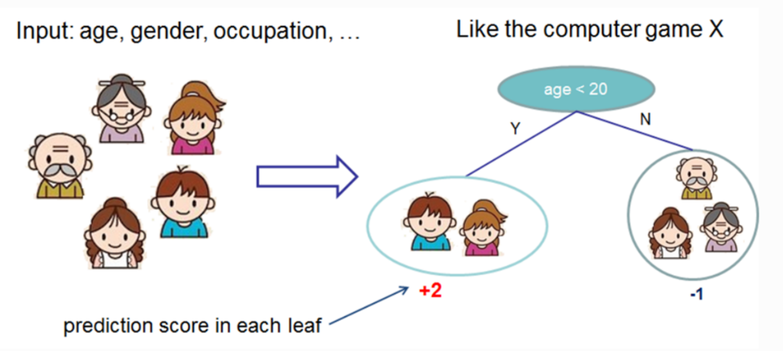

Input: age, gender, occupation, ...
Like the computer game X
age < 20
Y
N
+2
-1
prediction score in each leaf

Figure 9: A simple illustration of a decision tree ensemble using family members as an example.Source: (XGBoostDev, 2020).

For instances, the figure above classify family members into different leaves, and gives them a score on the corresponding leaf, which in this case are based on age and the preference of a video game. The real score is associated with each of the leaves, which gives a richer interpretation of that go beyond classification.

A single tree is not strong enough, and the ensemble model is thus used to sum the prediction of multiple trees together.

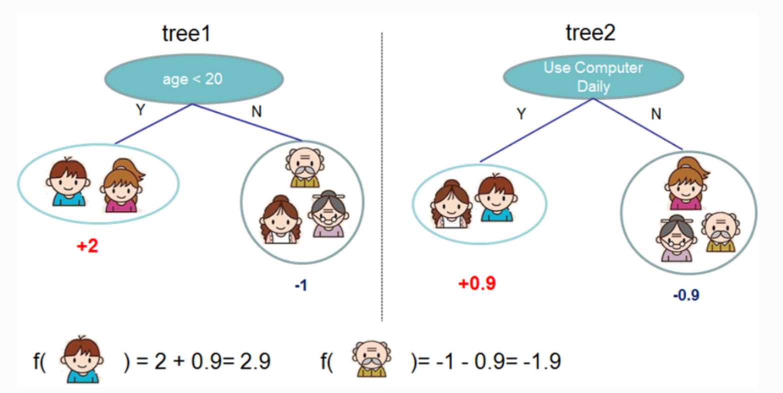


Figure 10: The model are splitted into two trees: One for age and one for daily computer usage, creating an ensemble. (XGBoostDev, 2020)

The figure above shows a tree ensemble of two trees. The prediction scores of each individual tree are summed up to get the final score. It can be mathematically be expressed as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \Upsilon \tag{17}$$

where K is the number of trees, f_k is a function in the functional space Υ, which is the set of all possible classification and regression trees (Chen and Guestrin, 2016).

In short, XGBoost learns a model by taking a weighted sum of a suitable number of base learners. The base learners are a part of an ensemble. This differs from ordinary machine learning which try to learn one hypothesis from training data, while ensemble methods construct a set hypotheses and combine them to use.

Moreover, a regularized objective is minimized in order to learn the set of function used in the model by the expression:

$$\mathcal{L}(\phi) = \sum_i l\left(\hat{y}_i, y_i\right) + \sum_k \Omega\left(f_k\right) \tag{18}$$

where $\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$. In this context, l represents a convex loss function that represents the difference between the prediction $\hat{y}$ and the target y. The second term Ω on the other hand penalizes the complexity of the model. It helps to smooth the final learnt weights to avoid overfitting.

The tree ensemble model is trained in an additive manner. Several base learners or functions are explored, and an objective function is picked for minimizing the loss, having the expression:

$$L^{<t>} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{\langle t-1 \rangle} + f_t\left(x_i\right)\right) + \Omega\left(f_t\right) \tag{19}$$

$\hat{y}_i^{(t)}$ is the prediction og the i-th instance at the t-th iteration, where f_t is added to to minimize the objective to improve the model. The purpose of the training loss function is to measure how predictive a model is with respect to training data.

2.7 Criteria for land subsidence analysis

In general, the choice of criteria for land subsidence and main details of geographical information are not expressing land subsidence susceptibility assessment based on statistical methods, but rather as a decision-making process defined by multiple experts, as they have more in-depth knowledge of local movements in specific locations (Gigović et al., 2019, Rikalovic et al., 2014, Karlsson et al., 2017). This chapter will briefly explain the 7 criteria chosen for this project. Several of the criteria have some connection with other natural hazards, especially with landslide, and some of the articles referenced are multicriteria analyses which emphasizes on creating general natural hazards susceptibility maps. There however still some criteria that are particularly relevant to subsidence such as altitude to groundwater table.

2.7.1 Slope

Slope stability remain one of the most important factors for evaluating risk of natural hazards. This is because slope stability affects numerous factors relating to land subsidence including terrain, underground water level, mechanical parameters, and unit weights of geomaterials (Wang, 2019). Traditionally, most MCDA relating to natural hazards uses slope as a criteria where higher degrees of slope are considered to increase the susceptibility. For example, when evaluating "safe" areas for portals in mountain tunnels or railways, it is often preferred to consider areas with flatter terrain due to difficulties in construction, maintenance and usage for slopes higher than 1.5 degrees as anything higher could increase the risk of landslides (Karlsson, 2016).The case for land subsidence is on the other hand quite different. Pourghasemi and Saravi (2019) for instances concluded in his creation of land subsidence inventory that land subsidence occur on flat slopes and smooth land as the increase in slope percentage decreases the subsidence occurrence. Despite preferring a flatter terrain when evaluating natural hazards, lower slopes are more susceptible to land subsidence. This is due to the speed of surface runoff is less, giving more sufficient time for water to penetrate deeper into the subsurface and form dissolution cavities (Yeh et al., 2016). Determining areas safe to natural hazards are therefore highly dependent on the slope, and deciding the most suitable slope degrees are contextual.

2.7.2 Soil typr

Soil type has been used in multiple multicriteria analyses for natural hazards (Gigović et al., 2019, Kanungo et al., 2006, Rikalovic et al., 2014). Variation in structure and composition of different rock and soil types, also determines the strength of the material. Stronger rocks give more resistance to driving forces compared to weaker rocks and are hence less prone to susceptibility of natural hazards. The composition of soil must thus be emphasized due to the fact that the study area are mostly situated below the marine limit, where numerous instances of marine clay mixed with other deposits pose a threat to ground stability due to the risk of quick clay slides (Andersen et al., 2019).

Soil compaction are often caused by the structures of rocks and extraction of groundwater from them, especially from highly porous sediments (Petersen-Perlman et al., 2017). This quite a crucial factor since land subsidence itself are affected by groundwater drawdown associated with afforestation and water exploitation activities (Rahmati et al., 2019). The lithological condition in an area would thus determine the rate of infiltration and indirectly influence the occurrences of land subsidence.

2.7.3 Distance to rivers

The evolution of alluvial plains in the area during the Holocene period due to eustatic sea level rise has resulted in occurrences of marine deposits nearby the rivers of the study area in this project. Studies integrating machine learning methods for land subsidence susceptibility indicates that incidents of subsidence movement is closely correlated to the distance from a river network (Pourghasemi and Saravi, 2019, Wang et al., 2019). In such studies, the highest weights of subsidence occurrence are often assigned at a distance of <50 m from a river. It is therefore often implied that lower distances from rivers have the greatest effect on land subsidence.

2.7.4 Altitude of groundwater table

Among all possible causes of land subsidence, withdrawal of groundwater is particularly troublesome, primarily because it is able to produce noticeable settlements with rapid rates. Recent alluvial, marine or lacustrine deposits with altered coarse-grained water bearing strata and fine-grained compressible layers has been recognized to be more vulnerable to subsidence induced by groundwater withdrawal (Modoni et al., 2013). It should also be noticed that excessive groundwater pumping often weakens the soil structure in the subsurface and often controls the subsidence rate. For instances, field data from Shanghai measured 2-3m of subsidence in the central area ever since land surveys took place in 1921, while the subsidence curve flattened when the pumping activity was limited (Shen et al., 2004). This is mainly related to the consolidation mechanism caused by pore water pressure drawdown. Similar behavior were also observed in the Marand Plain, Iran, where reduction of total volume in an aquifer for agricultural purposes was considered to contribute to land subsidence in a multicriteria analysis (Ghorbanzadeh et al., 2018). Massive exploitation of groundwater resources was regarded as the main reason for a severe decline in groundwater levels in the last three decades in the area.

The hydrological system, the geotechnical system, and the building system are often correlated to each other when evaluating the impact of groundwater on land subsidence. Withdrawal of groundwater from deep aquifers causes hydrological effects in the aquifers and the semi-pervious layer. Changes in the groundwater table elevation along with the drop of the piezometric head impacts the geotechnical system. Within the geotechnical system, processes such as oxidation, shrinkage and compaction plays a crucial role for the degree of land subsidence. Oxidation as a result of dewatering, can for instances increase the subsidence in soil with high content of organic material. Additionally, lowering the piezometric head and groundwater table may cause the pore pressure in compressible Holocene layers to decrease and increase the effective stress in these layers, resulting in compaction and settlement of soil. Damages to buildings is a result between the calculated hydrological effects, and the geotechnical system which determines the settlement.

2.7.5 Land use and land cover

Aside from different environmental factors on the spatial distribution of natural hazards, land use and land cover dynamics are important for assessing susceptibility of natural hazards since vegetation types and covering dimensions determines ground stability to some extent (Meneses et al., 2019). Certain changes in land use such as deforestation and slope ruptures to roads increases the number of unstable slopes in an area, which in turn promotes the propensity of occurences in natural harzards. The land use variable itself is rather dynamic over time and is often influenced by climate-driven changes and anthropogenic impacts (Promper et al., 2014). Land use and land cover maps can thus be directly implemented in hazard models in order to evaluate different scenarios of hazard susceptibility and at the same generate consistent susceptibility maps. This was also evident from Rahmati et al. (2019) where anthropogenic influence on the surface have significant effects on subsidence activity. In more detail, Orhan (2021) conducted a study researching the connection between land cover and subsidence activity in Konya, Turkey. He concluded that different types of land cover such as agricultural areas, vegetation and urbanization did all positively impact the subsidence rate of the local area. It happened each time major changes of land cover occurred, and the increase of subsidence were related to declination of groundwater levels, but each type of land cover did however decline the groundwater levels at a different pace.

Utilizing remote sensing is quite common to generate an land use map layer for analysis of natural hazards. Chen et al. (2019) for instances managed to extract a landslide susceptibility map using landslide-related factors from optical remote sensing imagery with predictive models using decision trees. The DEM's in their study were generated from ASTER satellite imagery where the classes are divided into grass land, forest land, farmland, residential areas and water bodies. Such classes are also commonly used from other authors such as (Chen et al., 2019) or (Karlsson et al., 2017). Water bodies are however quite irrelevant in the case for analyzing land subsidence activities. Vegetation could also be analyzed further using Normalized Vegetation Index (NDVI) which quantifies greenness and density of vegetation. NDVI commonly occurs along with land cover when studying land subsidence for machine learning (Rafiei Sardooi et al., 2021), but this will however not be included further in the project as a criterion since most of the study area is heavily urbanized (Nordgulen et al., 1998) and using it as criterion could otherwise just increase the value of all pixels in the map layer when performing overlay analysis.

2.7.6 Topographic wetness index (TWI)

The topographic wetness index (TWI) quantifies the topographic control on hydrological processes, and was first developed within the runoff model TOPMODEL (Sørensen et al., 2006). TWI is commonly defined as $\ln(\frac{\alpha}{\tan\beta})$ where α represents the local upslope area draining through a certain point in the watershed per unit contour length and $\tan\beta$ is the local slope. TWI has often been used as a criteria in multicriteria analysis where the study site is more vulnerable to floods and heavy rainfall (Chen et al., 2019, Karlsson, 2016). This is because topography controls spatial distribution of groundwater level and soil moisture which can trigger land subsidence events caused by saturated conditions within the soil. TWI is also used as a criterion because it is a secondary topographical vector defining the degree of deposition of water at a given position (Arabameri et al., 2020) which in turn also contributes to the degree of groundwater drawdown. However, TWI does not account for soil type and land cover, both of which are affecting the water saturation in the area. This must therefore be taken into consideration for the overlay analysis in GIS.

2.7.7 Rock type

A prediction of potential subsidence and its magnitude are of great interest in land reclamation or drainage projects. Compression/compaction, consolidation, shrinkage, and oxidation are the four main causes of contributing downward movement of the ground surface. Unlike other processes, subsidence caused by consolidation can only occur on clays or soils of low permeability. The amount of subsidence brought by these processes is a function of the pore space of the original material, the effectiveness of the compacting mechanism, and the thickness of the deposit undergoing compaction (Glopper and Ritzema, 2006). Geology and soil type contain properties of permeability based on grain size, but they are only sufficient for predicting subsidence caused by compression/-compaction which only occurs on soil with at least some sand. Oh et al. (2019) did for instance include both soil type and permeability when they conducted a study to predict land subsidence in abandoned coal mines in Taebaek, South Korea. In underground coal mining subsidence are often caused by sinkholes occurring due to sudden collapse of overburden into underground voids, which in turn are due to the decline of water levels in the soil as a result of pumping. Despite a wide range for similar lithologies is indicated, it appears that underground voids holding considerable amounts of water tend to be stored in sediments of low permeability that commonly constitutes the bulk of aquifer-aquitard systems (Domenico and Mifflin, 1965). This is especially relevant to take the rocks permeability and its vulnerability of weathering in consideration when the study area in this project is situated in an urban environment with lots of construction activity underground. This has experience-wise proven to be one of the main causes of anthropogenic subsidence such as Zarazoga city, Spain Sevil et al. (2017) and Napoli, Italy Scotto di Santolo et al. (2016).

3 Methodology

3.1 Evaluation criteria

Creating a land subsidence susceptibility map requires data from multiple types of criteria, originating from different sources, data types and resolution. During the MCDA-process, satellite images, geological maps, point data of wells, and

DEM's were applied to generate a susceptibility map from the study area. As mentioned earlier, seven criteria were taken into consideration to study the susceptibility in the area, including: slope, distance to streams, land use and land cover, rock type, topographic wetness index, and water table altitude. This chapter will give an overview of the criteria chosen for the study.

Table 1: The data used in the susceptibility assessment, the data sources, and the associated factor classes for the landslide susceptibility mapping in the study area.

Data layer	Source data	Data type	Scale/Original resolution
Slope	DEM obtained from Kartverket	Raster	50 m
Distance from major streams	River data from NVE	Line	1:50 000
Land cover and land use	LANDSAT-8 OLI	Raster	50 m
Risk of deep weathering	NGU and Norwegian Road Authorities	Polygon	1: 150 000
Soil type	NGU	Polygon	1:50 000
Topographic Wetness Index	DEM obtained from Kartverket	Raster	50 m
Water table altitude	GRANADA database	Point	1:50 000

The methodological hierarchy behind generating the susceptibility maps is based on the GIS-multicriteria decision analysis (GIS-MCDA) structure. This approach uses the capabilities of GIS for management of geospatial data and the flexibility of MCDA to combine factual pieces of information from the criteria with value-based information such as expert opinions, meaning that the integration of GIS and MCDA supplement each other.

3.2 Spatial database construction, environment set-up and data preparation

All data used in the current study were georeferenced to Universal Transversal Mercator (UTM), ETRS_1989_UTM, zone 32. Data sources originally in vector data format were transformed into raster-data with a 50x50m cell size. This is because not all criteria layers had finer resolutions available.

The data of the criteria were stored in a geodatabase within ArcGIS Pro 2.4.

The program is one of the latest GIS application for Esri, which opens up opportunities to explore, visualize, and analyze data. Some of the criteria comes in forms of vector data such as points, lines or polygons to describe the geographical phenomena. Since the project requires querying attribute data and involves reclassifying a subset of data within each criteria, all types of data were converted into map layers and stored in multiple separate geodatabases: One containing original data, one exclusively containing map layers under processing, and one containing the final prodcts. The geospatial database management system (DBMS) within GIS are designed to store and manage large amount of data to handle ordering, sorting and retrieval. This is to ensure that data are quickly available while still maintaining the integrity. This is to protect data against deletion and corruption, and to facilitate the addition, removal and updating of data when necessary. Geographical data are quite special in way that each data for database modelling contains both spatial and attribute information associated with it (McDonnell et al., 2015). A relational database were integrated in the GIS-analysis which stores data in tuples. All records have identification codes to use as unique keys to identify records in each file. Relational databases often have the advantage of being very flexible in terms of meeting the demands of all queries that can be expressed as Boolean logic or mathematical operations.

ArcGIS provides a plethora of tools for processing data. The tools are grouped into toolboxes by the type of actions they perform and each toolbox contain toolsets that further defines the tools by their functionality. For instances, the Analysis toolbox is grouped into Extract, Overlay, and Statistics. The toolset Proximity then consists of tools involving calculations about the nearby surroundings of points, lines and/or polygons such as buffer analysis and polygon neighbors.

Both the MCDA-analysis using AHP and machine learning algorithms creating the susceptibility maps for land subsidence requires use of multiple tools and toolsets such as spatial analysis, data management, and geostatistical analysis. To optimize the workflow of the project, the programming language Python was integrated with the GIS-environment to run and automate most of the tools used for processing the data in the analysis. Geoprocessing scripts begin by importing the arcpy-package in Python. This package ensures all of the functionalities in ArcGIS are available in Python. The arcpy-package has an object-oriented design which can defined using object-oriented terms (Tateosian, 2015). The semi-automated script used to prepare data sets for MCDA-analysis can be found on appendix A. The environment was set up on Jupyter.

3.2.1 DEM-based criteria: Slope and TWI

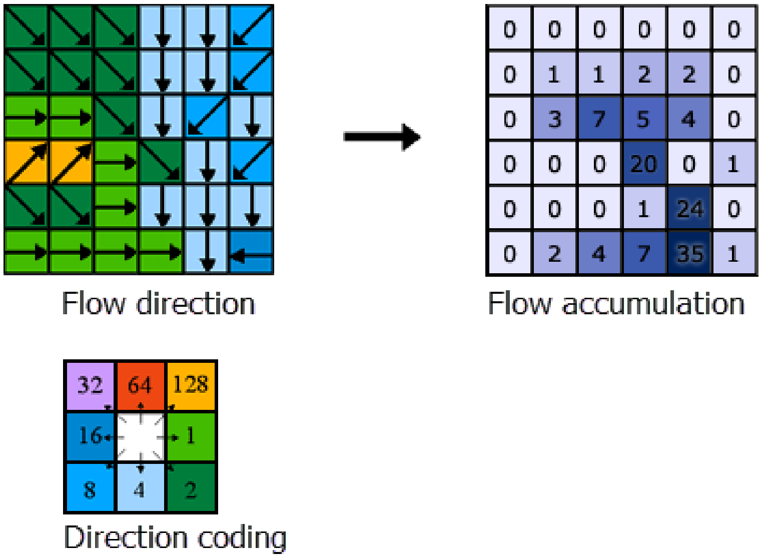
the digital elevation model of the area was extracted from the Norwegian Mapping and Cadastre Authority (Statens Kartverk) with a spatial resolution of 50x50 meters. The DEM was produced by the authorities through planes and/or helicopter with mounted laser scanners. The data-sets have a point density with 2 points per square meter. The slope of the area was calculated through the DEM where the pixel values give an estimation of the elevation in meters above sea level. In the analysis, the slope is defined in degrees. The calculation are performed by using a 3 by 3 cell neighborhood with using a planar method as an algorithm for calculating the slope. For each cell, the slope-tool in ArcGIS calculates the maximum rate of change in value from a given cell compared to its neighboring cells. In other words, the maximum change in elevation is defined as the distance between a given cell and its eight cell neighbors, which also identifies the steepest downhill descent from the cell. Mathematically, this is expressed as:

$$slope = \arctan(\sqrt{\frac{\partial z}{\partial x}^2 + \frac{\partial z}{\partial y}^2} * \frac{180}{\pi}) \tag{20}$$

The equation shows that the rate of change of the surface in the horizontal ($\frac{\partial z}{\partial x}$) and vertical ($\frac{\partial z}{\partial y}$) direction from the cell center determines the slope, while $\frac{180}{\pi}$ converts the value from radians to degrees.

Besides from slope, the criterion for TWI was also based on the DEM from Statens Kartverk. Generating a TWI for the study area required using several hydrological tools within ArcGIS. After gathering the DEM of the study area, the resulting raster-layer was used further to calculate the flow direction. This is the direction the water will flow out from an area that is represented by a specific raster cell. The flow direction is decided from one cell by estimating the gradient for each neighboring cell (ESRI, n.d.). The direction with the steepest gradient will determine the flow direction. The D8-method was used for giving an approximation of the flow direction, which measures one direction of flow out of eight possible directions because of 8 neighboring cells. When direction is decided, the relevant cell in the drainage raster is coded with a value that represents the specified direction.

Furthermore, the resulting layer from flow direction are processed by a tool that calculates flow accumulation. This tool provides information about how many cells that drains to a certain point. In other words, the tool calculates the accumulated flow as the accumulated weight of all cells flowing into each downslope cell in the output raster. The number of cells is given as a float-number accordingly to the D8-method as illustrated in figure 11.



0	0	0	0	0	0
0	1	1	2	2	0
0	3	7	5	4	0
0	0	0	20	0	1
0	0	0	1	24	0
0	2	4	7	35	1

Figure 11: Each raster cells are first assigned a direction of travel in the flow direction tool, then a value is given during flow accumulation processing. The top right image shows the number of cells that flows into each cell. Source: (ESRI, n.d.)

After undergoing flow direction and flow accumulation, the slope-raster needs to converted to radians. ArcGIS in general works with trigonometrical functions based on radians, and it is thus necessary to calculate the local slope tangent in order to get the TWI. This was performed in the raster calculator tool that builds and executes map algebraic expressions. The original slope-raster was used as input, and each cell within the raster-grid were multiplied by a factor of $\frac{\pi}{180^o}$. The raster calculator was then used to calculate the local slope (β), which is based on an if-then-statement. The local upslope area draining through a certain point per unit contour length α on the other hand are based on the earlier flow calculations. In raster calculator, the statements would be the following:

```
tan (beta) = Con(slope>0,tan(slope),0.001)
alpha = (flow_acc + 1)*cell_size
TWI = ln(alpha/beta)
```

The local slope is based on a condition stating that only cells with a slope-value higher than 0 will be calculated in the tangent function. Otherwise the cells that does not satisfy the condition are automatically assigned as 0.001 in the output-raster. The α-value was rescaled by adding a factor of 1 to the raster-layer of the flow accumulation, and then multiplied by 50 in order to scale with the cell size used for the general MCDA-analysis. The TWI-raster layer is finally generated by the natural logarithm of the upslope area and the tangent of the local slope.

It should be notified that water can flow from one point to an infinite number of direction rather than just eight directions as defined for the D8-method Rød (2015). The flow direction in this part is hence restricted to to eight direction, which represents a surface flow that are less distributed than in a real life scenario.

3.2.2 Criteria from vector-data (soil and rock type)

The soil type layer contains information about the general distribution of unconsolidated materials covering the surface of the mountain. The data shows only which soil type that are dominating the first upper meters of the terrain. Thicker and thinner layer from other types of soil may occur further down the soil profile. The data for soil type is based on the content of different Quaternary geological maps in different scales (from 1:20000 to 1:250000). The soil and rock data from Norwegian Geological Survey (NGU) were converted into raster format, reprojected to the proper reference system, UTM ETRS 1989, and masked in order to generate a layer only containing the study area. The maps are converted in digital format by scanning and vectorizing the Quaternary maps. There are in total 11 different classes in the original soil map layer, consisting of: bare ground, fluvial deposits, anthropogenic material, sea- and fjord deposits (inconsistent cover), fjord- and sea deposits (consistent cover), humus cover, marine beach deposit, moraine material, rand-moraine, organic material, and wind deposits. Once the soil type layer has been clipped to only contain the study area, only 4 classes remain. The original data for rock types did also had much more classes in its original form and did contain all rock compositions existing in Norway. The classes however are much more limited in the study area, and it was therefore decided to only include the six classes that covered the study area. Additionally, the area is primarily dominated by slate and limestone. Most of the classes are thus different degrees of a combination between those rocks.

River Generating map layer for proximity to main river streams were based upon calculating the euclidean distance from the lines of river streams based on data from the Norwegian Water Resources and Energy Directorate (NVE). In this case, the Euclidean distance tools within ArcGIS Pro describes the cell's relationship based on the line distance from the river dataset from NVE. The river streams functions as a source which identifies the objects of interest. The distance is then calculated from the center of the source cell to the center of each of the surrounding cells. The output values for the Euclidean distance raster are described as floating-point distance values, and contains the measured distance from every cell to the nearest source.

Groundwater table To create the map layer for altitude of water table, a combination of point data representing different type of wells were acquired NGU's national groundwater database (GRANADA). Those wells consisted mainly of densely distributed measurements along the urban area of Skøyen, with registered groundwater levels. The well data are based on several geotechnical ground measurements such as cone penetration testing and oedometer tests. The tests are further investigated in to acquire knowledge about, the density, water content, plasticity boundaries, organic content and strength parameters of the soil. There are in total 223 points with measured groundwater level from the database.

The dataset from GRANADA gives a national overview of groundwater wells, energy wells, and natural groundwater resources, which covers the entire mainland Norway (NGU, 2017). The groundwater wells in GRANADA are defined as points where groundwater was extracted for the purpose of water supply, surveillance of groundwater parameters over time, energy wells for heat pumping, insights for natural water resources, and for research purposes. All data points registered from GRANADA are given the map projection ETRS UTM zone 32 with varying resolutions. Not all data points from GRANADA within the study area are used in the analysis. In fact, only 113 wells from GRANADA were taken further in the analysis. This is due to the fact that most wells in the area did not measure groundwater level and lacks proper date for measuring the groundwater level, giving them an automatic point value of 0 meters. A simple SQL-query in GIS are therefore applied to only include point data with given groundwater values above 0 meters.

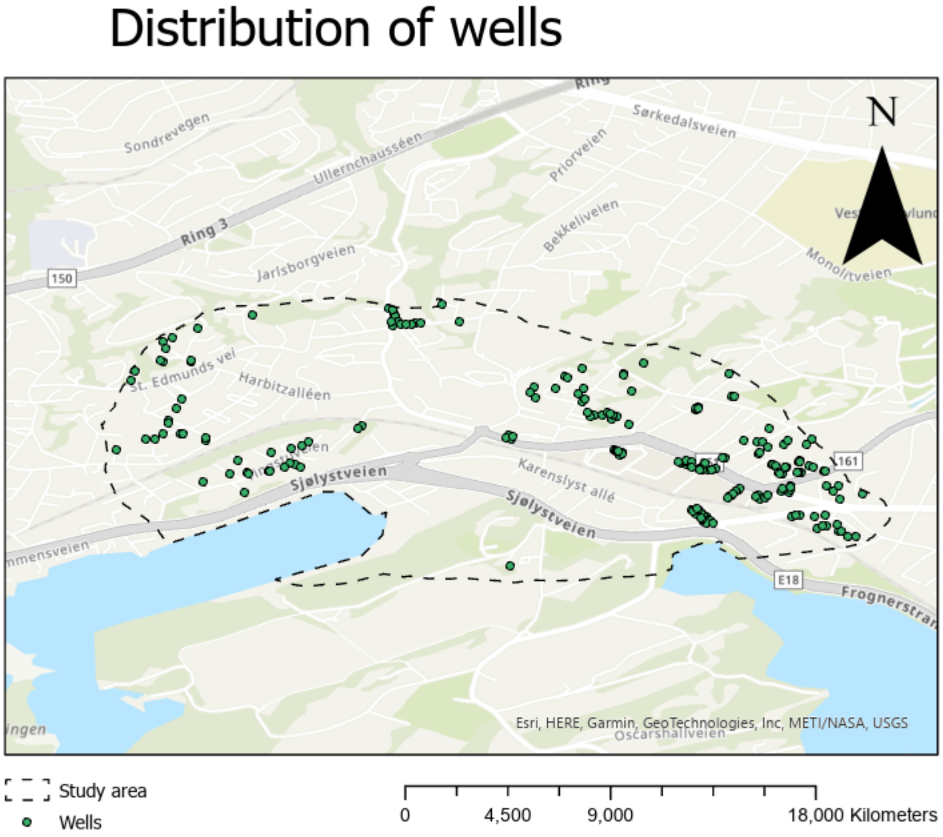


Figure 12: A map illustrating the distribution of well data in the study area. The point data was gathered from NGU's GRANDA database. There in total 215 wells within the study area with verified water level.

Data points from the wells were merged together. A new attribute table was then created by subtracting the groundwater level with the altitude, which is represented by the same 10m-resolution DEM for calculating the TWI of the area. The newly created point values, containing the groundwater level elevation, are then interpolated by using the principles of inverse distance weighting (IDW). IDW gives more weight to nearby points than to distant points. The analytical expression of the surface is stated as:

$$f(x,y) = \frac{\sum_{j=1...N} w(d_j)v_j}{\sum_{j=1...N} w(d_j)} \tag{21}$$

where N is the number of wells, v_j represents the point j value, d_j is the Euclidean distance with point j, and $w(d)$ is the weighting function, which has the formula:

$$w(d) = \begin{cases} \frac{1}{d_{min}^2}, & d \leq d_{min} \\ \frac{1}{d}, & d_{min} < d < d_{max} \\ 0, & d > d_{max} \end{cases} \tag{22}$$

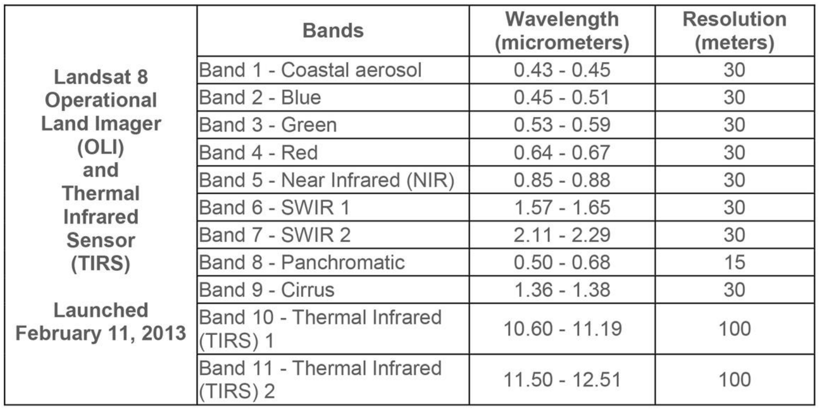
where d_{min} is the minimum distance and d_{max} is the maximum distance. The index d_{min} prevents infinite weight values for $d = 0$, while d_{max} avoids using too distant points in the weighting process (Caruso and Quarta, 1998).

3.2.3 CORINE Land Cover

The land cover map comes from CORINE Land Cover (CLC). It consists of an inventory of land cover with 44 classes, using a minimum mapping unit of 25 hectares for areal phenomena and a minimum width of 100 m for linear phenomena. The data set for the map layer are from 2018 and were retrieved from NIBIO. Nothing much were edited in the data layer before reclassifying. Only 4 land cover classes exist within the study area, excluding water.

3.2.4 Attempts of remote sensing for classifying land cover

There was an earlier attempt to classify land cover through remote sensing in order to get land cover maps newer than 2018, but this was later scratched from the project as the chosen map layer had more accurate coverage. Classifying land use and land cover were based on spectral signatures of different objects in satellite images from LANDSAT 8 OLI, which is defined by the radiance emitted from a material with respect to certain wavelengths. Landsat satellites have multiple sensors with different bands and resolution. Each band cover a specific wavelength, but some of the bands also overlaps other bands in the same wavelength area, making them compatible for change detection analysis Rød (2015).

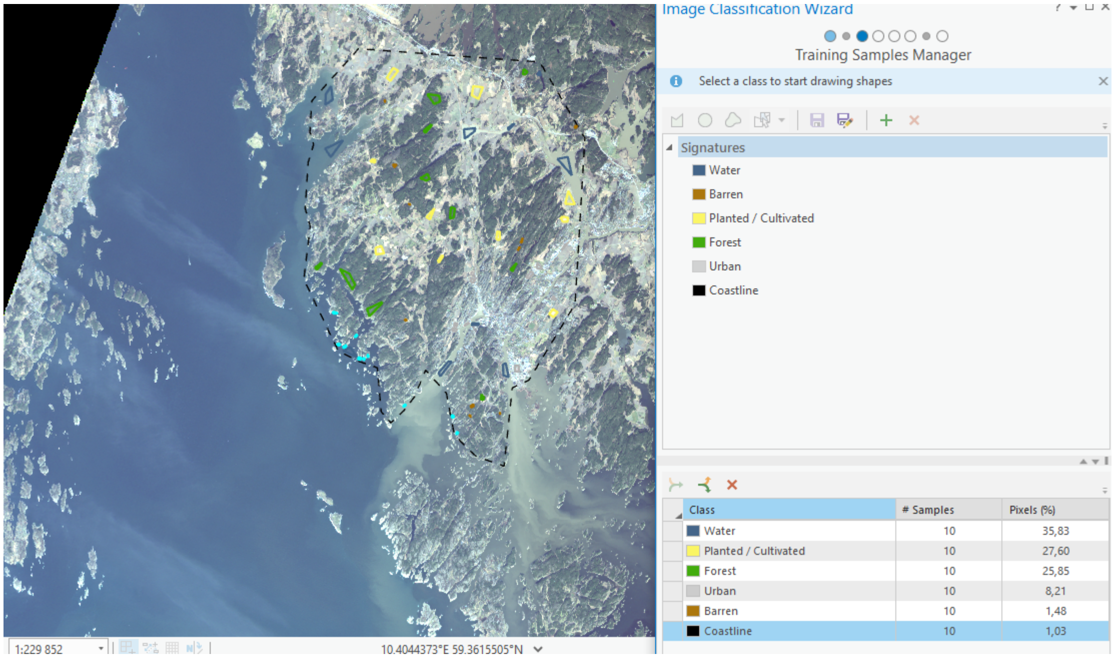


	Bands	Wavelength (micrometers)	Resolution (meters)
Landsat 8 Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS) Launched February 11, 2013	Band 1 - Coastal aerosol	0.43 - 0.45	30
	Band 2 - Blue	0.45 - 0.51	30
	Band 3 - Green	0.53 - 0.59	30
	Band 4 - Red	0.64 - 0.67	30
	Band 5 - Near Infrared (NIR)	0.85 - 0.88	30
	Band 6 - SWIR 1	1.57 - 1.65	30
	Band 7 - SWIR 2	2.11 - 2.29	30
	Band 8 - Panchromatic	0.50 - 0.68	15
	Band 9 - Cirrus	1.36 - 1.38	30
	Band 10 - Thermal Infrared (TIRS) 1	10.60 - 11.19	100
	Band 11 - Thermal Infrared (TIRS) 2	11.50 - 12.51	100

Figure 13: A table consisting of Landat 8 OLI's wavelength bands with defined resolutions. Source: (USGS, n.d.b)

The bands in Landsat 8 OLI are stored as a single band raster dataset. A single raster dataset was created as a combination of all the separate band images to make a raster dataset containing information about the entire wavelength spectrum of the image in order to assign training polygons. The maximum likelihood classification (MLC) tool in ArcGIS is one of the more common approach to classify land cover pixels, and was considered to be used as a classifier in the project. However, using MLC requires a large training area and the data must be normally distributed (Binti Saiful Bahari et al., 2014),which is not the case for the study area. A support vector machine was therefore used as the chosen classifier within ArcGIS Pro's classification wizard. With SVM, misclassification is also minimized since the classifier finds optimal separating hyperplanes between classes by focusing on the training cases (support vectors) that lie at the edge of class distributions, while excluding other training classes.

To assign training polygons the band combination was set to its natural color composite (4,3,2). The classification process is supervised, with a set of classes that are manually established. The parameters of the classes represents a signature, which is defined by a statistical distribution of how frequent the colors blue, green and red appear in the class. The signatures of the different classes is generated by using training samples in ArcGIS. A number of polygons is established for each class, creating a mean-value of how frequent the complementary colors appear within the polygons of the classes they represent. In this case, 6 land cover classes were identified which each of them having at least 10 training polygons, i.e water, forest, bare ground, urban and coastline. The average color combination in the classes were then stored as a signature file to distribute the pixels of the satellite image to the class where the color combination matches the most.



Class	# Samples	Pixels (%)
Water	10	35,83
Planted / Cultivated	10	27,60
Forest	10	25,85
Urban	10	8,21
Barren	10	1,48
Coastline	10	1,03

Figure 14: An overview of the workspace environment in ArcGIS Pro for classifying land-cover. 10 training polygons for each class are drawn within the study area. The Landsat 8 OLI image in the background is the subject for gathering spectral information.

Figure 14 shows the process of creating a land cover map in Fredrikstad as a prototype project. The idea was later scratched off because it had a tendency to misclassify gray rocks as urban areas as well as the coastline. Furthermore, the CORINE land cover map had more detailed classes such as distinguishing between continuous and discontinuous urban fabric which would otherwise be nearly impossible to do manually.

3.3 Normalization of criteria

Before the criteria went into the weighting process, the values within each criterion must first be standardized to get abstract spatial data/criteria into numerical input model such that arithmetic operations are possible to perform. For instances, the soil type layer contains nominal text values that would otherwise be impossible to interpret the relative importance without giving it interval values. The standardization process were performed by reclassifying each criterion into several classes, which were individually normalized in a scale between 0-10. This subchapter will briefly explain the choice of normalization for each class.

Table 2: The ranking process of each classes for each criterion.

Criterion	Class name	Scale
Slope	0 - 3	10
	3 - 6	8
	6 - 9	6
	9 - 12	3
	12 <	0
Distance to streams	0-110	10
	110 - 250	7
	250 - 410	3
	410 <	0
Land use and land cover	Water	Irrelevant (NODATA)
	Continuous urban fabric	10
	Discontinous urban fabric	8
	Green urban areas	4
	Arable land	0
Soil type	Anthropogenic	5
	Sea deposits and marine clay	10
	Marine deposits	0
	weathering materials	7
TWI	5 - 6.9	0
	6.9 - 8.1	3
	8.1 - 9.7	6
	9.7 - 12.3	8
	12.3 <	10
GW-table elevation	0 - 6.5	10
	6.5 - 12.7	8
	12.7 - 18.6	6
	18.6 - 25.3	4
	25.3 - 38.6	0
Rock type	Slate	0
	Slate w. lenses of limestone	4
	Knollekalk w. slate	6
	Knollekalk	8
	Limestone w. sediments	10
	Slate w. knollekalk	2

The table above shows the normalization process of all the classes within each criterion. The criteria "Land use and land cover", "Soil type" and "Rock type" all contain nominal values, where the classes are distinguished by a naming system rather than a numeric value, while "slope", "distance to streams", "TWI" and "GW-table" all have ordinal values, which are given a numeric value where most of them have given unit.

The slope layer originally had stretched values which ranged from 0 to 15 degrees. As mentioned from subchapter 2.6.1, slope usually are normalized in a manner that classes containing flatter slopes are given lower values in the normalization process. What is considered "lower" values is quite relative. (Ghorbanzadeh et al., 2018) for instances divided the slope classes evenly where a new class are created for every fifth degree , i.e (0-5, 5-10, 10-15, etc.). Similarly was done when defining the slope classes, creating a new class for every third degree. The normalization process of the slope criterion is chosen with respect to runoff water's capability to infiltrate to the subsurface. Since flatter terrain tend is usually considered easier for water to infiltrate according to Yeh et al. (2016), the lowest slope class were given the highest value, while the value for the next classes decreases the higher slope values are within a class.

The choice of classes within the "distance to streams"-criterion follow a similar pattern. Studies from Pourghasemi and Saravi (2019) have shown that there are to a larger extent some correlation between distance to river and subsidence events where areas closer to river streams. This also related to how proximity to rivers could the groundwater level in the area and eventually also how the risk of erosion could affect the subsidence rate (Hakim et al., 2020b). It was thus decided to normalize the river class where the first 110 meters closest to river streams were given the highest values while the area furthest away were given the lowest value during the normalization process.

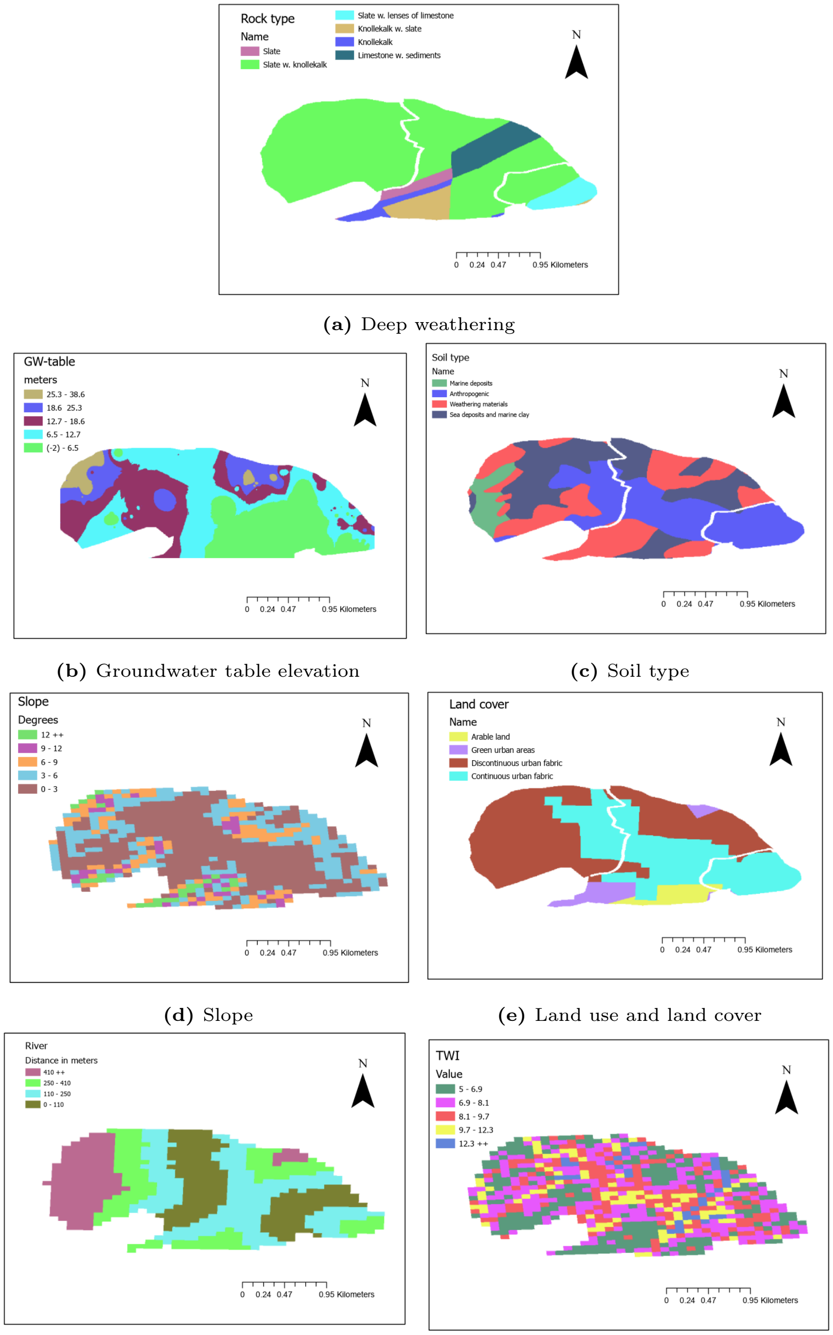
The land cover layer was one of the layers with nominal values. It is somewhat difficult to determine which type of land cover that could potentially contribute most to the subsidence process as each type of land cover do to some experience land cover due to groundwater depletion. One could argue that it depends on the water content of each land cover type. However, urbanization appears to be the main factor of groundwater pumping, and studies has shown that urbanization was the clearest indication that groundwater was consumed. Additionally, the municipality has allegedly planned to build new housing areas and a new metro station within the study area (*Områderegulering for Skøyen*, 2021), which indicates that the urbanization process is still active in the area, and thus giving these type of land cover the highest value. Vegetation and arable land has both the potential cause significant depletion of groundwater as well. The "arable land" class was however given the lowest class since it is unclear whether or not the area is fully cultivated or if the area is irrigated. The water class were assigned with "NODATA". This is because those classes are irrelevant for measuring susceptibility ground subsidence. When a class is assigned as "NODATA", the reclassification process in ArcGIS Pro will exclude those classes and the pixels belonging to these classes will no longer be counted in for the weighted overlay analysis.

The soil type layer originally contained 11 soil types, but was reduced to 4 classes. This is because these soil types are the only classes that is confined within the study area. The process of assigning the values are given accordingly to grain size, water content and permeability. Thus, the lower the porosity of soil type, the lower is the class value. On a side note, the class "anthropogenic" is given a value in the middle of the normalization scale, since anthropogenic materials could have a varying rate of porosity and permeability.

As mentioned from chapter 2.6.6 TWI is a topographic variable that expresses flow accumulation in soil due to slope and upstream catchment areas. The higher the value, the larger the flow accumulation, and the normalization is therefore arranged in a way where higher values the lowest value would be in the lower scale. TWI is a unitless scale from 0 and upwards. Since the lowest TWI-value in the area are 5, the class value begins from there and new classes are formed for every increase of 1.9. This is because the highest possible value is in a decimal value. Something similar was also used by Oh et al. (2019) where TWI was considered as one of the more important conditioning factors for land subsidence.

Changing the altitude of the groundwater table is a crucial conditioning factor for land subsidence since excessive pumping of groundwater could weaken the soil. It could be discussed if the groundwater level would be representative enough as a criterion as other studies have used it as a conditioning factor instead of (Rezaei et al., 2020a). However, the shape and height of a water table is influenced by the surface that lies above it. It curves up under hills and drops under valleys. The groundwater found below the water table comes from precipitation that has seeped through the surface. In the normalization process of GW-table, the lower water table altitudes , given in meters, were ranked the highest, and the next classes would decrease in value. This is because smaller differences between altitude and water table could potentially increase the risk of groundwater drawdown within an area compared to an area where the water table lies very low Ghorbanzadeh et al. (2018). The elevation would change over time due to variations of groundwater levels as a result of groundwater extractions (Chen, Wang, Hsu, Yu and Kuo, 2010).

As for the rock type layer, the normalization process were based upon on the strength of the rock materials and the risk of weathering. For instances, limestone with sediments were given the lowest value since dissolution of carbonate rocks are a quite common cause of sudden collapse. This is because rocks containing calcite is very susceptible to dissolution by groundwater during the process of chemical weathering. The same applies for "knollekalk", which contains lenses of limestone but contain slate as well. Slate were given the highest value since it can be considered a very durable stone and usually resistant to weathering (Wichert, 2020).



(a) Deep weathering

(b) Groundwater table elevation

(c) Soil type

(d) Slope

(e) Land use and land cover

(f) Distance to rivers

(g) Topographic wetness index

Figure 15: The seven criteria prior to reclassification

3.4 Analytical hierarchy process

The analytical hierarchy process (AHP) is a combination where the analyst (i.e the author) uses expert knowledge to assign weights to a series of parameter maps. It is a method based on the principles of decomposition, comparative judgement, and synthesis of priorities Malczewski and Rinner (2015a). When decomposing, the decision problem were divided into a hierarchy that captures the elements of the problem. The principle of comparative judgement on the other hand requires an assessment of pairwise comparisons of the elements with a given level of the hierarchical structure Malczewski (1999b). The synthesis principle takes each of the derived ratio-scale priorities in the various levels of priorities for the elements of the lowest level of the hierarchy. In other words, the AHP involves developing the hierarchy, assigning weights of importance to each element/criterion of the hierarchical structure using pairwise comparison, and finally constructing an overall priority rating.

The ranking process of each criterion tend to variate and the expert knowledge of the in this project is to some extent limited. Some of the evaluation were based upon similar studies such as (Karlsson et al., 2017). It is also worth to mention that all seven criteria could in theory also apply for other natural hazards as well such as flooding or landslides. The normalization process has thus played an important role upon how the criteria were ranked with each other. Land cover and slope were for instances the first and second most important criteria respectively since the normalization were more specifically tailored towards risk of land subsidence. For instances, lower slope classes were given higher normalization values, while the opposite is often more common when evaluating landslides (Skilodimou et al., 2019). This also explains why criteria such as TWI and distance to streams were not prioritized higher in the pairwise comparison matrix. Despite these criteria have been used as conditioning factor for land subsidence in (Arabameri et al., 2020), they are also applied more often in general studies of natural hazards such as in (Karlsson, 2016). Altitude of groundwater were on the other hand ranked lower despite being related to land subsidence. This is because the risk of water drawdown on itself could be affected by other criteria as well such as land use and rock type. It is also among the conditioning factors that are not as much prioritized from other studies (Rezaei et al., 2020b).

The pairwise comparison matrices in the analysis involved comparing all possible pairs of criteria to one another in order to determine which of the criteria is of higher priority. This creates a 7x7-matrix containing the land subsidence criteria.

The 7 criteria were then ranked accordingly to an underlying scale that was proposed by Saaty and Vargas (2001) for pairwise comparisons in AHP, with values from 1/9 to 9:

Table 3: Criteria ranking for AHP

Importance level	Scalar value	reciprocal scalar value
Equally important	1	1
Equally important to slightly more important	2	1/2
Slightly more important	3	1/3
Slightly important to much more important	4	1/4
Much more important	5	1/5
Much more important to very much more important	6	1/6
Very much important	7	1/7
Very much more important to extremely important	8	1/8
Extremely important	9	1/9

The 7x7-matrix in this project were stored in an Excel-file in order to calculate the weights, and contained the following values:

Table 4: The pairwise comparison matrix

	A	B	C	D	E	F	G
Rock type (A)	1	3	1/3	7	1/2	2	3
Altitude of groundwater table (B)	1/3	1	1/4	2	1/3	1/5	4
Land use and land cover (C)	3	4	1	6	2	3	7
Distance to streams (D)	1/7	1/2	1/6	1	1/5	1/3	1/3
Slope in degrees (E)	2	3	1/2	5	1	3	7
Soil type (F)	1/2	5	1/3	3	1/3	1	1
Topographic wetness index (G)	1/3	1/4	1/7	3	1/7	1	1

Fraction values indicates that the criteria is n-times less important than the other criterion it compares to.

After setting up the pairwise comparison matrix, the weights are estimated by averaging over normalized columns. This involves normalizing the entries in the comparison matrix C:

$$C^*_{kp} = \frac{C_{kp}}{\sum_{k=1}^{n} C_{kp}} \quad \forall \quad k = 1, 2, \ldots, n \tag{23}$$

The weigthts are then given by:

$$w_k = \frac{\sum_{p=1}^{n} C^*_{kp}}{n} \quad \forall \quad k = 1, 2, \ldots, n \tag{24}$$

The sum of all the weights are strictly required to be equal to 1.0.

The AHP-process is intended to be as transitive as possible. For any given criteria such as A, B, C, a consistent set of pairwise comparison would need that if land use and land cover (C) is three times more important as the rock type, i.e $3C > A$, and rock type is twice as important as altitude of groundwater table (B), $2A > B$, then the land use and land cover layer must be six time as important as the altitude of the groundwater table. However, considering that inconsistent judgement is inevitable when assigning importance of each criterion (Saaty, 1980), it was decided to implement an algorithm for measuring inconsistency during the decision making progress in for the pairwise comparisons. The inconsistency was determined according to the observation $\lambda_{max} > n$ for, positive reciprocal matrices and $\lambda_{max} = n$ if C is a consistent matrix. The consistency ratio is defined as:

$$CR = \frac{\lambda_{max} - n}{RI(n-1)} \tag{25}$$

where RI is the random index, which is the consistency index of a randomly generated pairwise comparison matrix dependent on the number of criteria used and λ_{max} is the principal eigenvector of the matrix. The consistency ratio must be at least less than 0.1, indicating that there is a reasonable level of consistency in the pairwise comparisons, whilst a consistency ratio exceeding 0.1 implies that it is necessary to reconsider and revise the primary values in the pairwise comparisons. The pairwise comparison matrix from table 4 has consistency ratio of 0.0664 which is within the recommended value. The algorithm used for calculating the consistency ratio can be found at the very end of appendix A.

From the AHP-process, the criteria was assigned the following weights from most to least important:

1. Land use and land cover: 0.335858
2. Slope: 0.243126
3. Rock type: 0.165127
4. Soil type: 0.104140
5. Altitude of groundwater table: 0.068511
6. Topographic Wetness Index: 0.050288
7. Distance to river streams: 0.032950

3.5 Weighted overlay

After rating the classes in each criterion and comparing the relative importance of the criteria to each other through AHP, the overall score of land subsidence susceptibility (LSSI) was estimated by the means of the weighted linear combination method according to the following equation:

$$LSSI = \sum_{i=1}^{n} R_i w_i \tag{26}$$

where LSSI correspond to the land subsidence susceptibility index, n is the number of criteria, R_i represents criterion layer containing the ratings of criterion number i and w_i is the weights for each criteria number i, which was generated from the AHP. ¨

The results of the susceptibility map are originally given in a continuous unitless scale from 0 to 10, where 0 labels areas least susceptible to ground subsidence, while 10 shows the most susceptible areas to subsidence. A score of 5 will thus be interpreted as neither low risk or high risk. Those results were later reclassified into four different classes with an interval based on natural breaks prior to investigating the the Skøyen-area. In brief, natural breaks classification are based on natural groupings inherent in the data. The class breaks groups similar values together and maximizes the differences between classes. The reclassified maps will be given as default values further in the analysis. The unitless scale are reclassified as following:

Table 5: Class division in MCDA

Class	Description	Value range
1	Low	0 - 3.8
2	Medium-low	3.8 - 5.6
3	medium-high	5.6 - 7.4
4	High	7.4 - 10.0

The class column represent the numerical value of each class in the reclassification. The description column shows the labels these classes are assigned during the analysis, and the value range will show how the original values are distributed into the classes. During the sensitivity analysis the results will be interpreted by the descriptions "Low", "Medium-low", "Medium-high", and "High" unless told otherwise. For comparing the results to MCDA with machine learning method, the first to classes will be merged into one class, and labeled as low-risk zone.

3.6 The data set

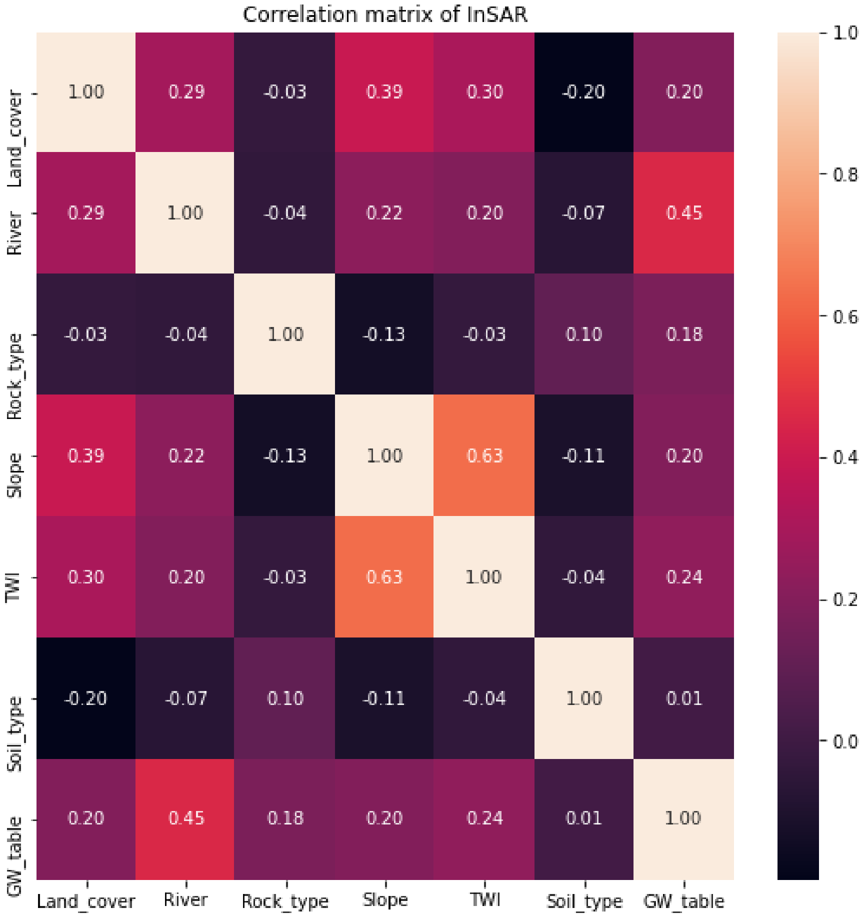


Figure 16: Correlation matrix between the criteria within the study area. The values inside each cell represent a score where a value close to ±1.0 indicates a strong correlation. The criteria are mostly not correlated to each other

The table above shows the correlation matrix of the criteria. It is observed that each individual criterion are relatively little correlated to the other criteria as the score remains mostly nearby 0. This can in particular be observed on the River-layer where almost all of its score-values lie within ±0.1. Its highest correlation is with the land cover layer with 0.45. The correlation is significant compared to the other criteria, but still quite low considering that the groundwater contributes to streams in most physiographic and climatic settings to a certain degree. The correlation between the slope and TWI-layer is the highest within the matrix with a value of 0.63. This is to be expected as the creation of the TWI-layer is dependent on the slope-layer. This might also explain the TWI-layers slight correlation with the river-layer.

On the other hand, the land cover layers is fairly correlated between most criteria with the exception of the rock type layer. Interestingly enough, its highest correlation is with the slope-layer (0.39). The river and TWI-layer are slightly correlated with scores of 0.29 and 0.3 respectively. There seems to be a negative correlation with the soil type. This might be explained that most of the surface area in the land cover layer is heavily urbanized while the soil in the study area does not exclusively include anthropogenic material. This might also be the same case for the rock type layer.

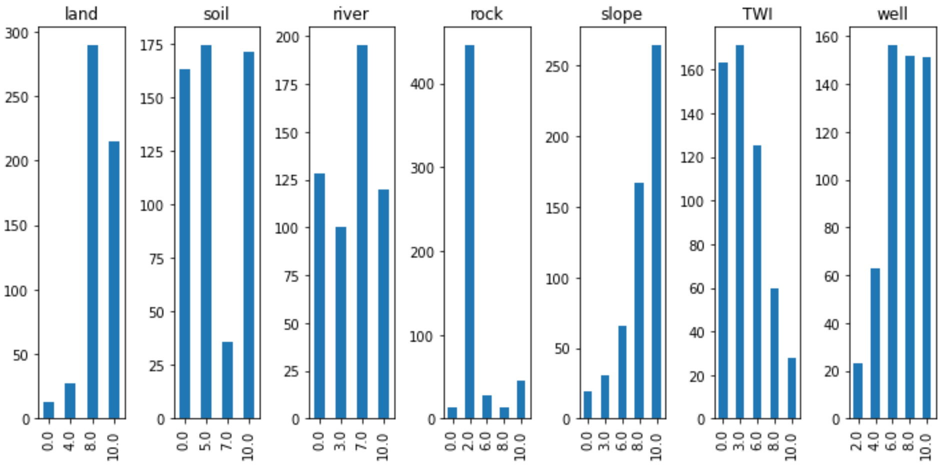


Figure 17: The distribution of the feature data set. The density of each criterion are measured and presented in a histogram.

The histograms above shows how the values from the different criteria are distributed. The x-axis shows the labeled classes for each criterion during the normalization process on chapter 3.3 on table 2.

The histogram of well, i.e GW-table elevation, in the study area are given given in m.a.sl. Based upon the histogram, most of the elevations of the water tables are on the three highest classes evenly distributed among them. Most of the area do have a water table elevation below 18.6 m.a.s.l on average.

The histograms of the slope shows that the vast majority of the terrain is is flat since barely any pixels in the analysis belong to a class lower than 4, indicating that only a few areas have a slope steeper than 6 degrees. The TWI-layer follows a similar pattern, but most of the values are classified in the lower classes instead. Barely any area of the map exceeds a wetness index above 9.7. The correlation between TWI and slope are as expected since TWI are derived from the slope-layer. The histogram for the river layer show that most of the study area is between 110 and 250 meters away from a major river stream, but there are also a fair number of pixels that are more than 250 meters away.

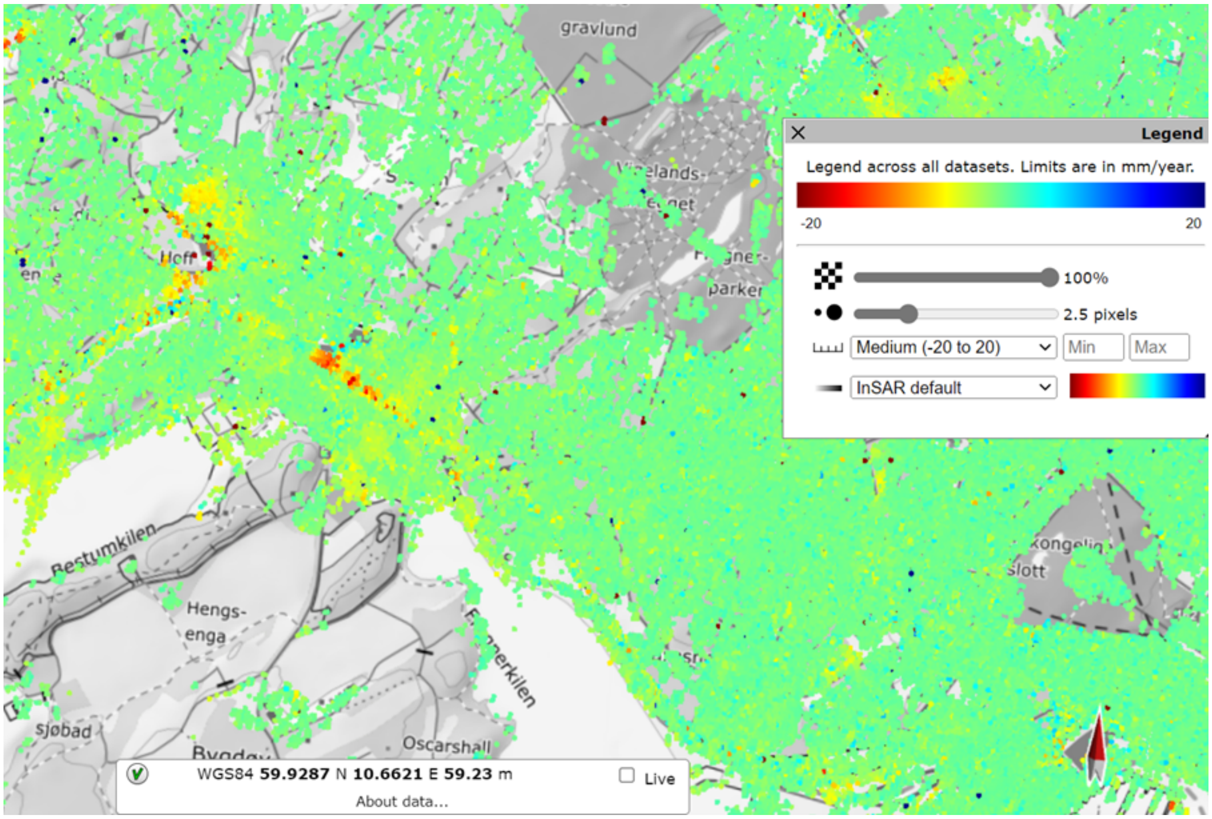
The soil, land use and rock all had their classes changed during the normalization process. For instances, the "soil"-histogram originally had four nominal classes that were converted into numerical values. Those were: Anthropogenic materials (5), marine deposits (0), sea deposits and marine clay (10), weathering materials (7). Considering the nature of the study area, it was an unexpected outcome that the pixels distribution is quite even between anthropogenic materials, sea deposits and marine clay, and weathering materials. Most of the anthropogenic layer are to be found on the core of the city area, but also near the docks while the type of soil with the lowest class rank are only to be found on the westernmost side of the study area, which mostly consist of green park vegetation. The pixel distribution of the rock type is very sparse. It appears that most of the area are covered in slate with "knollekalk", class 2. The only other significant rock to be found on the area is the limestone on the eastern part of the area. In addition, the land cover layer is quite homogeneous with two classes dominating, namely discontinuous urban fabric (8) and continuous urban fabric (10). Arable land (0) are only found on the southernmost tip of the study area.

3.7 Sensitivity analysis and comparisons

As the multicriteria analysis grows more complex in nature, enhancing GIS-based MCDA with sensitivity analysis procedures is crucial to understand the model behavior and its limitation. The sensitivity analysis explores the dependency of model outputs of the weights of the input parameters, identifying criteria that are especially prone to weight changes and to show impacts of changing weights to the model outcomes in a spatial dimensions (Chen, Yu and Khan, 2010). A common approach in the sensitivity analysis is to change the input factors one at the time, the OAT-method, to see what effect this produces on the output. This was applied recursively by removing one criterion at a time to evaluate changes in the pairwise comparison matrix during the AHP-process and to observe any improvements consistency ratio. By changing one factor at a time, all other factors can be fixed to some extent, and the comparability of the results increases. It also worth to mention that the OAT-method is convenient to implement due to its simplicity and not considered computationally expensive. Performing the sensitivity analysis serves the specific aspects of interests: (1) Quantify changes in weights once after removal of a criterion; (2) visualizing spatial changes; and (3) check for improvements of the consistency ratio. Regression analysis were also implemented between the values obtained from the original results vs. the new values gathered from the sensitivity analysis.

3.8 InSAR reference map

Information about ground subsidence activity is limited for the study area, and there are per this date no national database containing information about previous subsidence events. Processed InSAR-images were therefore used to attempt validating the ground subsidence in the area. The InSAR data set was gathered from the Norwegian Geological Survey (NGU). It consists of point data with different color codes to indicate ground movement from the Earth's surface to the satellite. The InSAR dataset originates from the two satellites, Sentinel-1A and Sentinel-1B, which is a part of the EU program for Earth observation. Those satellites move in polar orbit around the Earth and are labeled as ascending and descending depending whether the satellite orbit moves towards north or towards south. The radar aims right in the direction of the satellite orbit, where the azimuth angle are different for ascending and descending data.



Legend

Legend across all datasets. Limits are in mm/year.

-20 20

100%

2.5 pixels

Medium (-20 to 20) Min Max

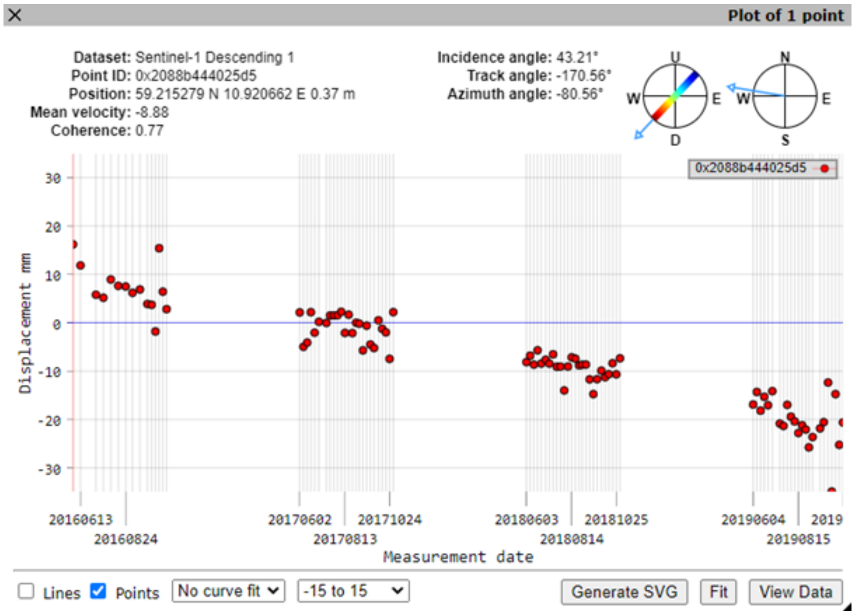
InSAR default

WGS84 **59.9287** N **10.6621** E **59.23** m □ Live

About data...

Figure 18: An example of InSAR-map from NGU. The units of the mean velocity of ground movement are described in mm/year. A positive value represents uplift while negative values indicates subsidence. Source: insar.ngu.no

Information about each point was gathered by simply clicking on a desired spot. Clicking will also reveal more detailed information about the point as illustrated on the figure below. Time series for multiple points can be gathered by drawing a polygon of the desired map. Each point contains the mean velocity, incident angle, track angle and azimuth angle. Approximately 80 000 InSAR-points were gathered from InSAR Norway, but only 554 InSAR points within the study area was taken in the analysis as the areas where those points are located, also contains data from the 7 criteria. Information of all InSAR-point within the study area has been gathered by drawing a polygon on the website insar.ngu.no stored as a csv-file.



Plot of 1 point

Dataset: Sentinel-1 Descending 1
Point ID: 0x2088b444025d5
Position: 59.215279 N 10.920662 E 0.37 m
Mean velocity: -8.88
Coherence: 0.77

Incidence angle: 43.21°
Track angle: -170.56°
Azimuth angle: -80.56°

☐ Lines ☑ Points
No curve fit
-15 to 15

Figure 19: Information of a random single point in the study area. The point data indicates that the area is gradually descending.

The data from InSAR Norway are still under development, and the current version used for the analysis still needs improvements. For instances, noises and outliers might occur occasionally, giving some points wrong values. Some outliers might however be accurate, and the outliers must be interpreted with precautions. Seasonal variation in wetlands might also impose anomalies due to variation of water content. Since the InSAR data mostly measures in the period from June to October, such variation can cause issues with the processing of data. This can be shown either by areas with large pointwise variations, i.e remarkable variations in mean velocity, or areas with continuous positive values.

3.9 InSAR-data as target value

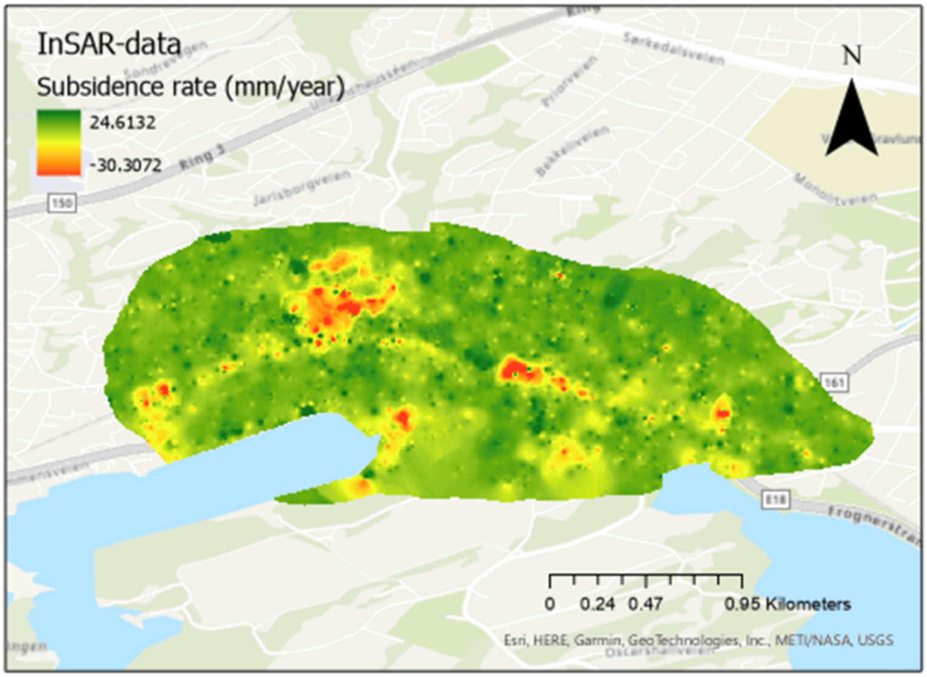

InSAR-data
Subsidence rate (mm/year)
24.6132
-30.3072
N
0 0.24 0.47 0.95 Kilometers
Esri, HERE, Garmin, GeoTechnologies, Inc., METI/NASA, USGS

Figure 20: The interpolated land subsidence map based on InSAR-data from NGU. IDW was used as interpolation method with 9374 InSAR data point values in total within the study area.

The figure above represents the initial target value in the machine learning analysis prior to classification. The units were given as mm/year, and the scales are based on the maximum and minimum pixel values detected in the study area. The extrema values are interpreted as outliers since they only consist of one pixel value each. Most of the observed values from InSAR lie between -6.3 to 2.3 mm/year as shown from the figure below. Moreover, the mean value of the study area was considered to have a subsidence rate of -0.8 mm/year indicating that the area are slightly subsiding on average.

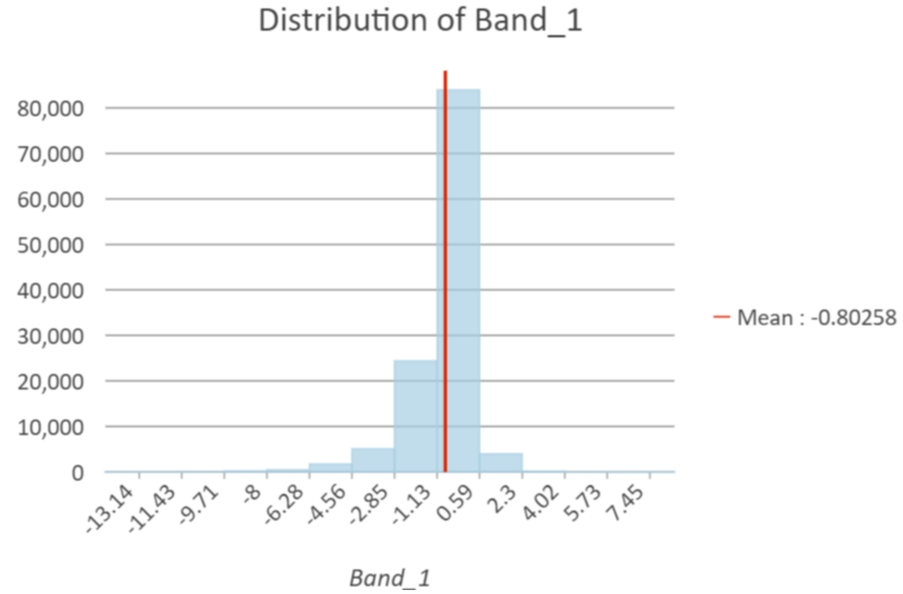


Figure 21: The interpolated land subsidence map based on InSAR-data from NGU. IDW was used as interpolation method with over 80 000 InSAR data point values in total within the study area.

The creation of land subsidence map was originally attempted to be solved as a regression problem. The results from generating subsidence map on a regression problem did however yield negative results as the R2-score remained very low. Thus, the initial subsidence map was reclassified accordingly to the method sections. The class values and colors defined in the initial target value will be used for all maps generated from neural networks and XGBoost.

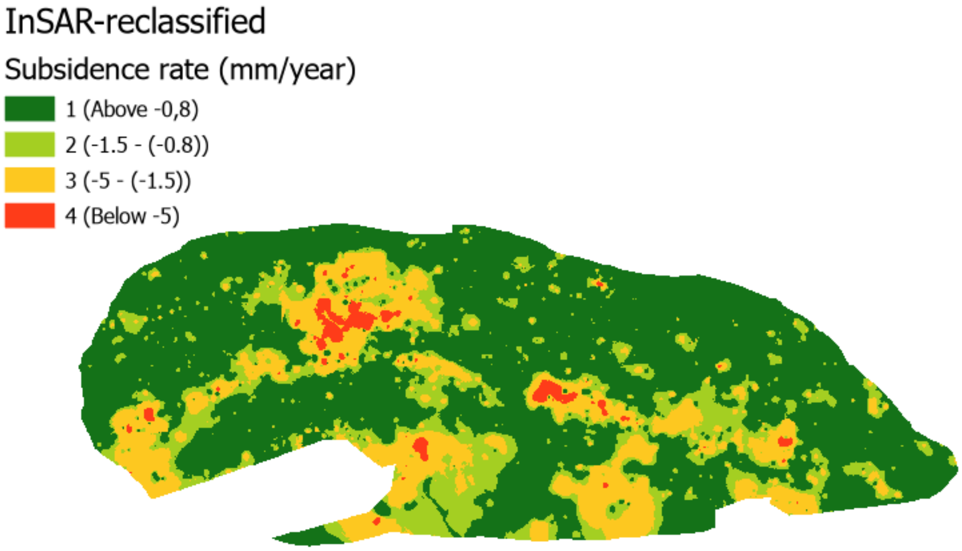


Figure 22: The reclassified InSAR-map. The values are distributed into four classes based on the degree of subsidence. Water was not clipped in the InSAR-layer, but all point data that would be transferred for processing in Python will be based upon the raster layer from the MCDA-analysis, which did erase water.

The reclassification are divided into four classes. The red zones shows areas that are subsiding more than 5 mm/year. Subsidence in this case was defined as a negative value. Class value 2 in yellow shows all values between -5 and -1.5 mm/year. The light green class is for all areas with subsidence rate between -1.5 and -0.8 mm/year, while the last class in dark green shows all subsidence rate above -0.8 mm/year including areas experiencing an uplift. The patterns are somewhat similar to the subsidence map from MCDA with subsidence commonly occurring nearby bodies of urbanized settlements. There however much more low-risk areas, considering that over half of the pixel values are categorized in class 1, but at the same time, the InSAR-layer has as mentioned earlier a finer resolution compared to MCDA. The southern tip on the study area did however contain more pixels within class 3, and the easternmost part of the area appears to be a lot "safer" compared to the results in the MCDA.

Table 6: Pixel distribution of the target value InSAR-data

Risk class	**No. of pixels**	**Percentage of total area (%)**
1 (Above -0.8 mm/year)	76872	63.9
2 (-1.5 - (-0.8))	20602	17.1
3 (-5 - (-1.5))	20817	17.3
4 (Below -5 mm/year)	1943	1.7

The resolution from the maps made by machine learning techniques are much finer in comparison to the MCDA-maps. The subsidence map contains 19 506 pixels in total. In the target value as shown above, almost half (47 %) of the area are considered to not being vulnerable to subsidence while just as many are classified to be in the middle (45 %), leaving around 8 % of the area to be considered as strongly subsiding.

3.10 Code implementation of Neural networks

3.10.1 Data preparation

As mentioned, 554 data points where picked from the study area based on where InSAR-data is available. Values from the 7 criteria in its reclassified form are extracted into each data point containing InSAR-data. The points from the processed InSAR-image will be used as a target value when performing the analysis in the neural network.

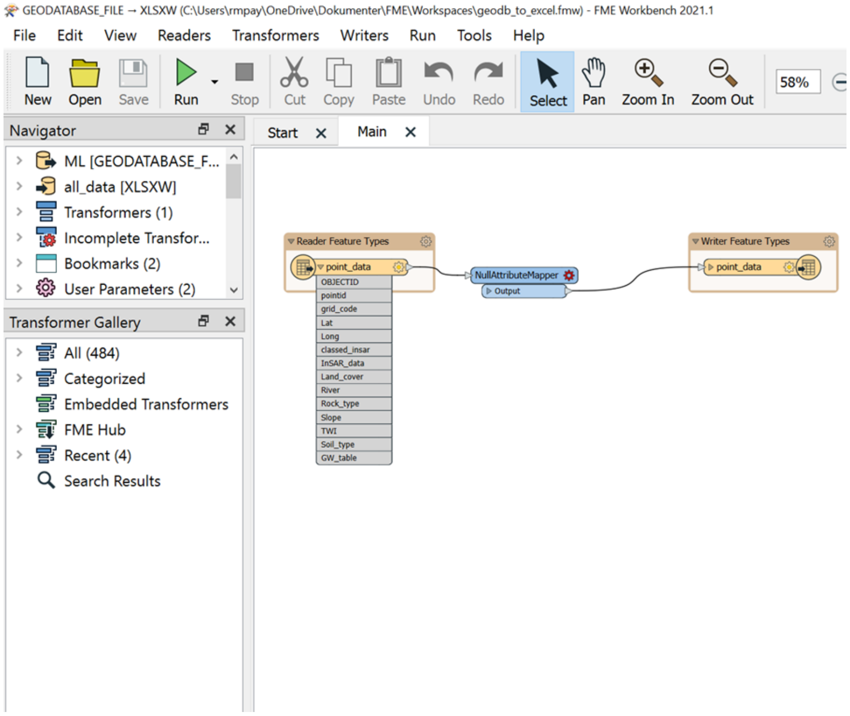


Figure 23: Converting the file geodatabase into csv-format in FME. The file geodatabase has in total 11 extracted attributes. These includes the 7 criteria, the latitude and longitude, and InSAR subsidence rate. A transformer in FME "Tester" was used to remove any empty values in the data set before conversion. Latitude and longitude was only converted so that the predicted data can be transferred back to ArcGIS, but was otherwise not involved to train the data.

The point data set are originally stored as a file geodatabase (.gdb). The data set must be converted into a csv-format in order to perform the neural network analysis and XGBoost in Python. Feature Manipulation Engine (FME) was utilized to convert the files, which is a platform that streamlines the translation of spatial data between geometric and digital formats. This can be observed from figure 23. Three transformers were used in FME to manipulate the data. Some attributes that are irrelevant to the analysis such as track angle, burst and amplitude distortion of the InSAR dataset were removed. Furthermore, a transformer called "tester" was used use to raise a condition that none of the attributes should contain a null-value. Every point instance with null-values in one of the eight attributes will be removed. In the final transformer, two new attributes will be created, namely "Binary" and "Multiclass" which classifies the values of InSAR mean velocity. This is in order to perform a prediction from the neural network. In "Binary", there are only two possible values: 1, which represents all negative mean velocities, and 0 that represents all positive values. Although very rare, if the mean velocity is exactly 0 mm/year, the point will be classified into 0 along with the positive values since it does not represent subsidence. The "Multiclass"-attribute on the other hand contains 4 classes. Unlike "Binary", the "Multiclass"-attribute also consider the degree of subsidence/uplift, and has the following class distribution:

Table 7: Class division in Multiclass attibute. In some cases, the classes 1, 2, 3 and 4 might also be refered to as "Low", "Medium-low", "Medium-high" and "High" respectively just like the MCDA.

Class	Subsidence (mm/year)
1	$-0.8 \geq$
2	(-1.5) - (-0.8)
3	(-5) - (-1.5)
4	$-5 \leq$

The latitude and longitude of each data point containing data of the 7 criteria and InSAR were added as field values before converting the file geodatabase into a csv-file. This is to ensure that it is possible to convert the file back into geospatial data when the predicted values are generated from neural network and added to the csv-file.

3.10.2 Training the neural network

The code produced in the neural network is built upon defining the structure of the neural network. It is already defined that there are 7 criteria taken into consideration to predict land subsidence, and thus 7 nodes will be needed to cover these features. It is also known that 4 output nodes are necessary since the target value are divided into 4 classes. When it comes to the of hidden layers, a reasonable number of nodes should be somewhere in-between the number of input layers and the number of output layers. In the Python code, the structure of the hidden layers of the neural network will be defined in a list such as $hidden\_neuron\_list = [5, 5, 5]$ or $hidden\_neuron\_list = [2]$. The first case gives 3 hidden layers with 5 neurons each while the second example defines one hidden layer with two hidden neurons. It is also possible to define a flexible number of neurons for each hidden layer.

The aim of the neural networks is to use it for classification problems. The entire structure is defined as an object-oriented function. The general neural network architectures contains the feed forward and back propagation algorithms training the network, and initializing the weights and biases. The neural network structure for classification can be found in appendix C. Additionally, the cost functions and activation functions used for the problem are stored in the class function appendix B. When initializing the classification problem through the class function, the type of activation functions used in hidden layers and the output layers are fixed. This also applies to the cost function as well. The feedforward and backpropagation algorithms are expressed in a matrix format which represents the formula from equation (5) and (13) in chapter 2.6.1, giving:

$$z^l = (w^l)^T h^{l-1} + b^l \tag{27}$$

and

$$\delta^l = \delta^{l+1} (w^{l+1})^T \sigma'(z^l) \tag{28}$$

To predict land subsidence with artificial neural networks the seven criteria was set up as a design matrix X with the multiclass pixel values from the InSAR-image used as a target value y. The data set was split into training and test data which consist of 30 % of the data set. To avoid outliers the design matrix was scaled before training the data in the neural network.

The number of neurons depends on the input data set, while the output layer on the other hand depends on the type of problem. In a binary case, there will be two output neurons and seven inputs. In a multiclass case, it is necessary for the output layer to classify multiple samples. A sensible neural network architecture would thus have an output layer of 4 nodes, where each of these nodes represents a class from 0 to 3. To make the neural network architecture as flexible as possible, a function in the Python-code for neural networks was created to convert a single vector that lines up with the n-node output layers. In this data set, if a point was predicted to class 2, the number will be converted as a vector $[0, 0, 1, 0]$.

Softmax was used as the activation function with cost function being defined by the cross-entropy function that uses the output error:

$$\delta^L = h^l - y \tag{29}$$

When training the neural network, the entire data set are scaled to help with the convergence of the neural network, which in particular important when combining different data types. The scaling was applied using a standard scaler from the Python module Scikit.

After defining and implementing different activation functions, the weights were randomly initialized for each $W^{(l)}$-layer. This was performed by using a loop over the number of iterations/epochs where the weight-matrix ΔW and its respective bias Δb were initialized to zero.

A feed forward propagation was performed through all the n number of layers where the activation function output $h^{(l)}$ is stored. Then, the $\delta^{(n_1)}$ value for the output layer is calculated while the backpropagation algorithm are calculated for the layers. Finally, the stochastic gradient descent step is implemented in the code. This is an iterative algorithm that starts from a random point on a function and travels down its slope stepwise until it hits the global minimum of that function. It would make the algorithm much faster since there is very little data to manipulate for every iteration, and due to its randomness, it can also escape local minima in the function (Géron, 2017). Mini-batches were also added before setting the data into training. They refer to equally sized subsets of the dataset over which the gradient is calculated and weights updated. The chosen cost function is averaged over a small number of samples. In this particular case, 20 mini-batches, bs, were created in default. The cost function will then have the structure of:

$$J(W, b, x^{(z:z+bs)}, y^{(z:z+bs)}) = \frac{1}{bs} \sum_{z=0}^{bs} J(W, b, x^{(z)}, y^{(z)}) \tag{30}$$

The stochastic gradient descent routine is repeated until the average of the chosen cost function has reached a minimum. At that stage, the network is trained and should ideally be ready for use. The cost function is important because it determines how well a machine learning model performs for a given set of data. As explained earlier, the cost function calculates the difference between anticipated and expected value.

3.11 Testing hyperparameters

The grid search algorithm was performed on both ML-methods. It is the process that searches exhaustively through the specified subset choosen from neural networks and XGBoost and it is a manually specified subset of the hyperparameter space of the targeted algorithm. It also evaluated the cost function of the respective algorithm based upon the generated hyperparameter set.

In most statistical models, it is preferable to tune different hyperparameters to obtain optimal values of the error metrics. This involves tuning every parameter and testing out different values for each of them such as the learning rate, number of epochs, number of hidden layers etc.

It is ideal to test as many hyperparameters as possible since they are often strongly related to each other. However, the neural network algorithm often has a large number of hyperparameters. In its simplest form, the neural network can have up to 5-10 different hyperparameters. This could for instance be the type of activation function used, the cost function and the number of neurons in a hidden layer. Another recurring issue with classifying the data is the runtime for optimizing every parameter. Visualizing what combinations of hyperparameters yields the best results also proves to be quite challenging. If 10 hyperparameters were tested for three instances each, then it would be approximately 59 000 combinations to test.

It was therefore decided to limit the testing to approximately 4 hyperparameters: number of epochs, learning rate, regularization value, and the number of neurons per layer. The type of activation function will be occasionally tested as well. A full list of values to test can be seen below. Every combination was run 15 with a randomly selected test data.

regularization value :	0.0, 0.1, 0.01, 0.001
Neurons per layer:	2, 5, 6
Epochs:	100, 500, 1000
Learning rate:	0.001 0.01, 0.1

Table 8: Table showing set of values tested for hyperparameter optimalization in neural networks

3.12 Predicting data with XGBoost

Just like in neural networks, the data set in ArcGIS were exported to Python for further processing. Each pixel value in the study area were defined as point data containing extracted values from the seven criteria, the coordinates, and the generated values from InSAR-data. The latitude and longitude were first added in the field by using the geometry calculator. The point data were converted from a File Geodatabase to an Excel-file through the conversion tool FME, where null-values were removed, the latitude and longitude were given proper UTM-values, and the InSAR-values were reclassified. Then the machine learning algorithms were performed in Python using the generated Excel-file. 70% of data set were used for training data while the rest were used for test data. Only the prediction from test data were converted back to ArcGIS, containing only the predicted values along with its latitude and longitude. FME were also used on this step, but no other modification were used at this point other than transforming the excel-file to an ESRI shapefile. Finally, IDW as an interpolation technique were used to predict the rest of the study area. The predicted values should then have a value of either 0,1,2 or 3 just like the InSAR-data. The grid search algorithm on XGBoost can be found on appendix D.

Similarly to neural networks, training in XGBoost is also executed by passing pairs of training and testing data. Key parameters in XGBoost includes the maximum depth, describing the depth of the decision tree; subsample, which is equivalent to percentage of data evaluated; and objective, which specifies the classification algorithm. Other relevant parameters also includes the number of estimators, learning rate, and parameters that could prevent overfitting of the data set. The hyperparameters mus be determined with parameter optimization. Each parameter to be estimated is represented by a list of values, and each combination is hen tested by the model whose metrics are compared to deduce the best combination. The search for parameters are then guided with metrics using cross-validation. Both grid search and randomized search will be utilized to find the best combination of parameters, which contains:

Number of estimators:	50, 100, 250, 500
Learning rate:	0.001, 0.1, 1.0
Maximum depth:	5, 10, 25, 50

Table 9: Table showing set of values tested for hyperparameter optimalization in XGBoost

The definition of learning rate in XGBoost remains the same as for neural networks. The number of estimators represents the base learners, and the maximum depth controls the depth of the tree that will be created. It can be described as the length of the longest path from the tree root to a leaf.

The analysis is also performed using the matplotlib library where the training results are plotted for each run in each XGBoost output. This is verified in order to understand if the iteration chosen to build the model was best one possible.

4 Results

4.1 Maps from MCDA

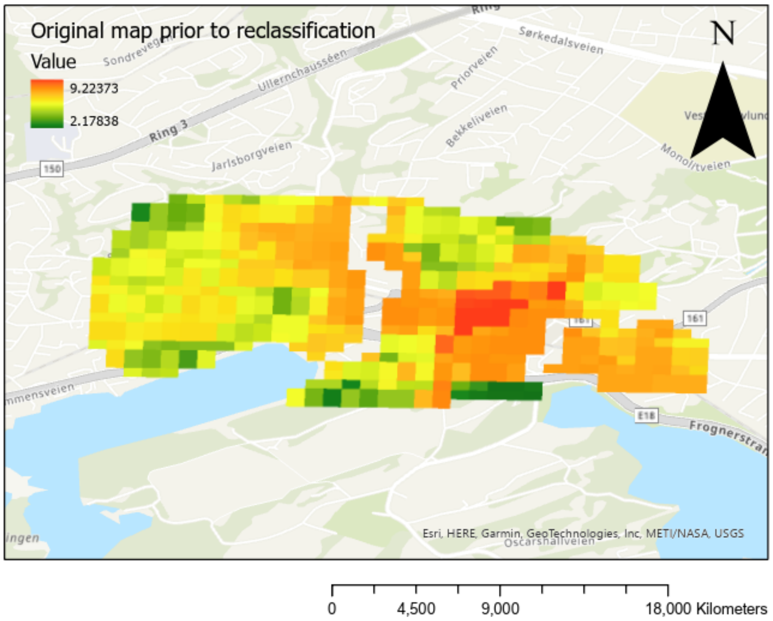


Figure 24: The land susceptibility map created from MCDA over the region. The value is a continuous unitless scale. Lower values indicates larger risk for subsidence while higher values indicates high risk of subsidence.

The figure above shows the final results of the MCDA using the 7 criteria and weighting each criterion by the standards of AHP. It is based on a unitless scale from 0 to 10 where the most ideal places are located in areas with lower values. In this particular area however, none of the pixel values managed to achieve a lower value than 2.1 or any higher value than 9.2. A visual interpretation of the map indicates that the most susceptible areas appears to be in places with many infrastructures, while the "safer" places are located in the western parts which also have a lot of vegetation. The majority of the pixel values seems to be closer to the maximum value than to the minimum value, which in turn might indicate that the entire study area is in general quite susceptible to land subsidence. This is mostly due to the prioritization of land use and land cover criterion which is dominated by urban fabric as observed from figure 15.

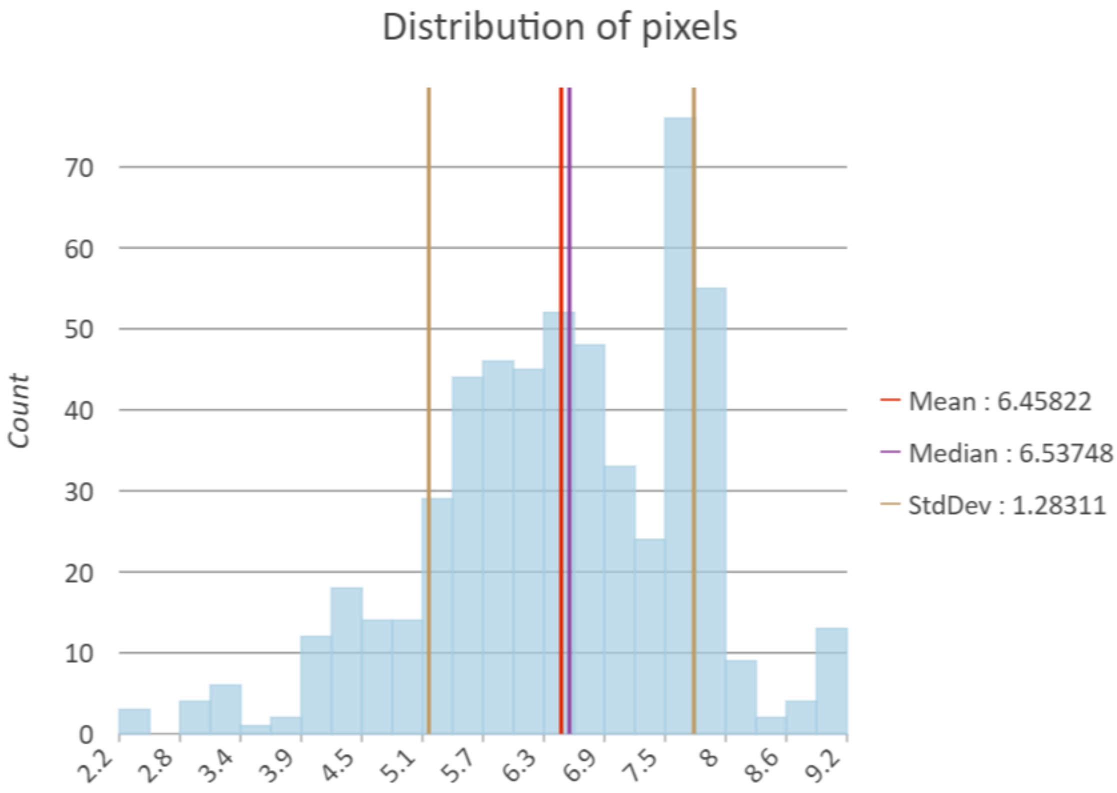


Figure 25: The distribution of pixel values from the MCDA in the entire area.

Looking at the distribution of pixel values in the first MCDA-map, the minimum value achieved from the weighted overlay analysis is approximately 2.2 while the maximum pixel in the area gained a score of 9.2. The occurrences of such minimum values are however insignificant since there are less than 10 pixels in total that are classified within the five lowest bars in the histogram. The mean pixel value lays around 6.45 which is quite high. The median value is a bit larger than the mean value with 6.5 as the most common pixel. Mos pixel values are within the range between 5.1 and 8.0 which signalize that the majority of the area are prone to land subsidence according to the 7 criteria used in MCDA. This might be an indication that anthropogenic activities, infrastructural projects could have an effect as there multiple ongoing site projects planned (*Områderegulering for Skøyen*, 2021).

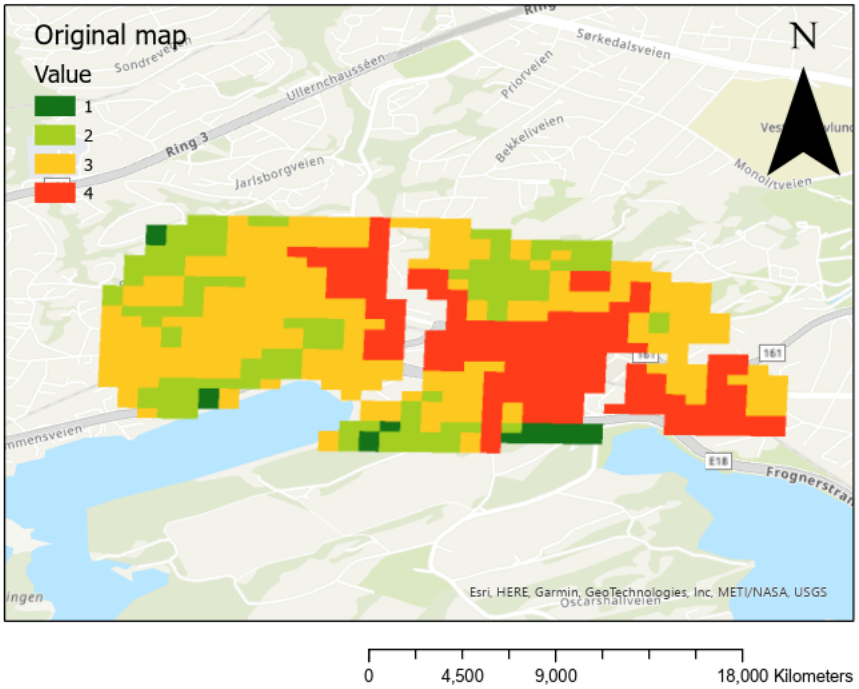


Figure 26: A simplified version of the susceptibility map in the study area. The values from the MCDA are reclassified into four different classes: low, medium-low, medium-high, and high risk. The intervals are decided upon natural breaks.

The classes in the reclassification follow the values from table 5 in chapter 3.5 and are based upon the distribution of the original map from MCDA where the classes are divided into 4 equal value ranges from 2.1 and 9.2. Class number 1 is extended from 0 to 3.8 to include pixel values lower than 2.1 for the maps created in the sensitivity analysis. Likewise, class 4 is also extended to 1 value 10 to include values over 9.2 in the sensitivity analysis.

It is evident that the classes follow a certain pattern to some extent. The center of the area, where most buildings are set up tend to be in the same areas classified as high risk. Despite proximity to river was among the criteria ranked the least important, every single pixel value adjacent to major river streams were either classified as medium-high or high risk. There are few outliers that have occurred in the process. This could for instances be the low risk pixels in the southern part of the map, surrounded by mostly high-risk pixel values. On the other hand, the distribution of the pixel values seems to be mostly consistent with high-risk pixels are far more prevalent in the central-eastern part of the area. It could also be observed that the majority of areas further away from the city center are classified as low or medium-low risk. The table below shows how the pixels in the susceptibility map are distributed.

Table 10: Pixel distribution From first MCDA results

Risk class	**No. of pixels**	**Percentage of total area (%)**
Low	16	2.29
Medium-low	116	20.9
Medium-high	260	46.9
High	162	29.91

There were 554 classified pixels in the study area in total. The majority of the pixels in the study area were classified within the medium-level risk of the category with 116 and 260 pixels classified as "Medium-low" and "Medium-High" respectively. The category "Low" and "High" cover 32.2 % of the area, which is not as much compared to the medium-level pixel which takes more than half of the map area in total with 67.8 % with barely any low-risk pixels at all.

4.2 Sensitivity analysis

To check the robustness of the results, the MCDA-method was redone by using the OAT-method as mentioned in section 3.9. One criterion was removed to observe how the assigned weights change and to see if there are any major changes from the original susceptibility map generated from the MCDA. Another method is to slightly adjust the weighted values from section 3.4.

In the first sensitivity analysis, the original weighted values are rounded into the following weights:

1. Land use and land cover: 0.30
2. Slope: 0.20
3. Rock: 0.20
4. Soil: 0.10
5. Altitude of groundwater table: 0.10
6. Topographic Wetness Index: 0.05
7. Distance to rivers: 0.05

During this weighting, slope and rock are equally weighted with each other. This also applies between soil and GW-table, and for TWI and river distance respectively. Land use and land cover are however still the criterion with the most weighting.

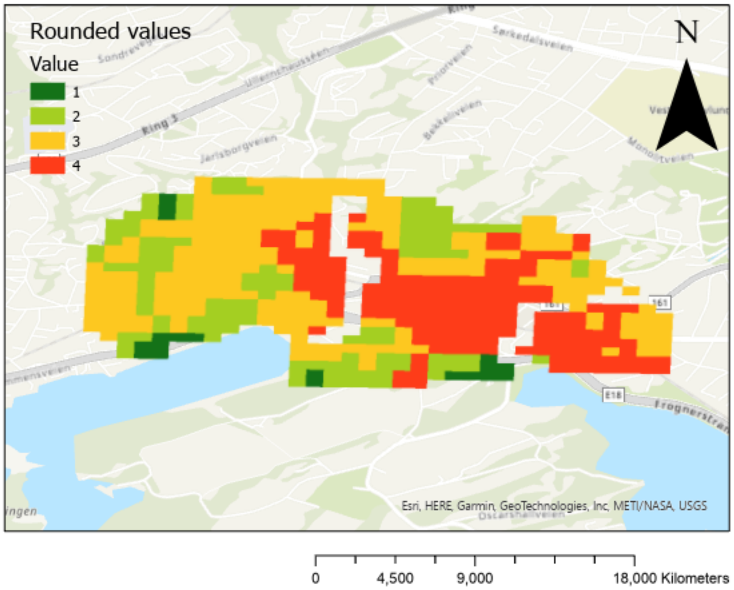
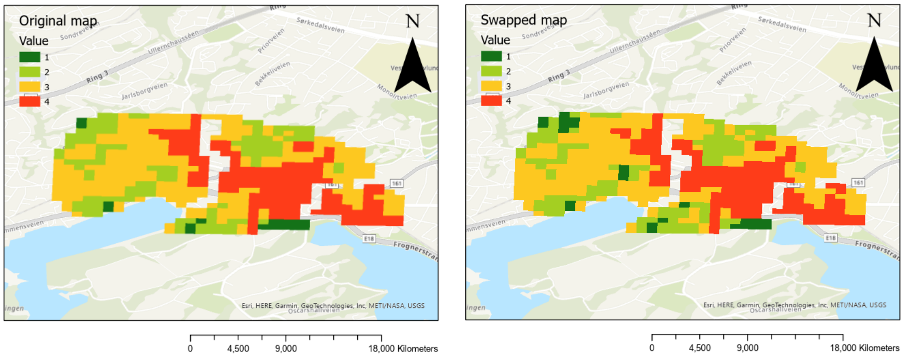


Figure 27: Rounded values to nearest fifth

As seen from the pattern of figure 27, the distribution of classes are still mostly the same. The medium-high classes are still more prevalent in the area. However, some high-risk pixels along the river canal are now ranked as medium-high in the northernmost part of the river. There slight changes in the low class pixels, but are mostly on the same place. Due to the rounded weighting produced mostly the same results as the original, it was decided not to include it anymore further when comparing results.

On the second sensitivity analysis, the weights of the most important criterion and the second most important criterion are swapped, giving slope a weight of 0.34 and Land use and land cover are assigned with a weight of 0.24. The distribution of the pixel values have then slightly changed compared to the original MCDA map.



(a) Original MCDA

(b) MCDA with swapped values

Figure 28: The original MCDA and swapped MCDA side by side.

Surprisingly enough, there are still minimal changes when swapping the weights between the land use and land cover criterion with the soil layer as seen from figure 28b. The shape of the high risk class remained the same and there are only minor changes, particularly in the northwestern tip of the map, which gained a few extra low-class pixels compared to the original map.

Most of the "shape" in the southern part of the map layer remained the same during the swap of weights. The city center and the most urbanized areas are still labeled as mostly high risk areas. The easternmost part are still considered as the safest, but the medium-high class has slightly shrunk eastwards. Other parts of the area do however still remain the same, but there are a bit more occurrences of outliers where low class pixels is being surrounded by medium-high pixels. It is also worth to mention that the pixels closest to water bodies mostly did not experience any change, with most of them still being classified as medium-high, but there are some pixels that transferred from high to medium-high.

On the third part of the sensitivity analysis, a single criterion was removed to check its impact on the result as a whole. It was decided to create a susceptibility map without the soil type layer, one without the land cover layer, and another one without the river distance layer in order to perform sensitivity analysis based on removing criteria with varying importance. This has naturally changed the distribution of weights and will affect the consistency ratio as well.

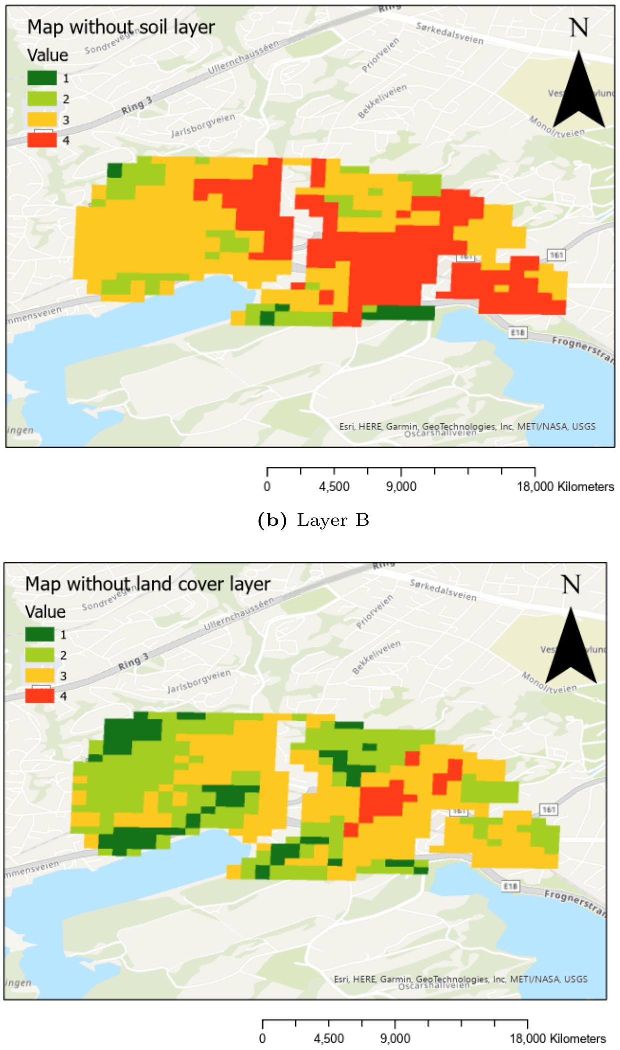
The susceptibility maps generated from MCDA without soil type and without land cover as a criterion will be referred to as layer A and layer B respectively unless told otherwise. It appears that all the criteria in layer B would gain a notable change in weight value due to LULC being originally considered as the most important criteria. The weights assigned became less consistent as

Table 11: Columns showing the changed weights when removing a specific criterion. Layer A represents an overlay analysis without the soil type layer, and layer 2 represents an overlay analysis without land use and land cover. The last row shows the consistency ratio for each of the resulting map layer.

Criterion	Layer A (% of weight)	Layer B (% of weight)
Rock type	0.1778	0.2573
GW-table altitude	0.0936	0.0967
Soil type	–	0.1503
LULC	0.3808	–
River distance	0.0366	0.044
Slope	0.2612	0.3777
TWI	0.0499	0.074
Consistency ratio	0.04933	0.08964

the consistency ratio increased up to 0.08961 compared to the original result (0.0664). It is however less than 0.1, which still made the weights consistent to an acceptable level. Removing the soil type layer did on the other hand slightly improved the consistency ratio with 0.04933.

Figure 29: Generated MCDA land subsidence map with one criterion removed. The distribution of pixel values between the 4 classes are significantly more different than the original subsidence map.



(a) Layer A

(b) Layer B

Map without soil layer
Value
1
2
3
4
N
Esri, HERE, Garmin, GeoTechnologies, Inc, METI/NASA, USGS
0
4,500
9,000
18,000 Kilometers
Map without land cover layer
Value
1
2
3
4
N
Esri, HERE, Garmin, GeoTechnologies, Inc, METI/NASA, USGS
0
4,500
9,000
18,000 Kilometers

The maps above illustrates land subsidence maps when removing one criterion in the MCDA. Figure 29a shows the subsidence susceptibility map when the soil criterion was removed while figure 29b shows the susceptibility map when the land use and land cover criterion was removed. It appears that the two highest classes has expanded slightly northwards when removing the soil criterion with a more defined high risk area on the eastern part adjacent to the river. Otherwise, the area are quite similar, but the low and medium-low class are starting to diminish.

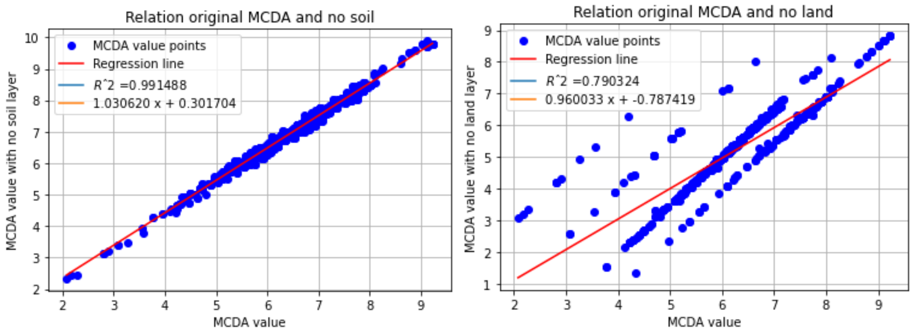
Removing the land use and land cover criteria on the other hand made the low risk classes expand. The medium-high classes has been reduced, and the high-risk class are now more concentrated in a particular area. Contrarily to the map without the soil type and the original MCDA risk map, there has been a drastic increase in the number of low and medium-low pixels. This is to expected as removing the criterion with most weights also had a lot of urban fabric. When removing the land use and land cover criterion, it is basically evaluating the study area without taking urbanization into consideration.

Layer	**Risk class**	**No. pixels**	**(%) of total area**
A	Low	14	2.5
	Medium-low	65	11.7
	Medium-high	269	48.5
	High	206	40.1
B	Low	69	12.4
	Medium-low	211	38.0
	Medium-high	249	44.9
	High	25	4.7

Table 12: Distribution of pixel of the two susceptibility maps.

The removal of one criteria has drastically change the pixel distribution of the risk maps and the impact is slightly different for each map. The table shows the exact number of pixels for each class in these two maps. The susceptibility map of layer A had a similar pattern as the original map. There are major differences with the medium-low and high classes. Layer A has 9.2 percentage points less in the medium-low category while it gained 10.19 percentage point in the high risk class. It has a more defined high-risk class area on the eastern part as well.

The susceptibility map without considering land cover (layer B) had a much more different distribution compared to the original. While there are minimal changes in the medium-high risk class, the low risk and medium low risk class has gained 10.11 and 17.1 percentage point respectively. The increase came at the expense of the high-risk class which now only cover 4.7% of the study area compared to 29.91 % in the original.Overall, layer B had by far the most areas labeled into the two lowest risk classes. Excluding the soil layer in the weighting process has lead to the majority of the eastern part of the study area being classified as medium-high risk. The areas near the borders of the study did on the other hand remain unaffected during the sensitivity analysis.



(a) Regression analysis of layer A

(b) Regression analysis of layer B

Figure 30: A simple regression analysis showing the relation between original MCDA and the map layers excluding soil layer or land cover respectively.

Additionally, a regression analysis between the map layers with removed criteria and the original MCDA was compared to each other based on their original MCDA-value scale prior to reclassification. This is to ensure if there are significant or major changes when excluding important criteria that were ranked highly during the decision making process and the source code are available on appendix E. The R^2-score for both maps still were relatively high with approximately 0.99 and 0.79 for the maps without soil or land cover respectively. It turns out that removing the soil layer did not have as much impact and were originally almost a perfect fit with the original MCDA. The slight changes when removing the soil layer would thus be more likely related to the class boundaries during the reclassification process. Removing the land cover layer on the other hand did show that the MCDA values are more shifted. Some pixels had even two or more units in difference between the original MCDA and the map without land use and land cover.

Table 13: Number of changed pixels compared to original MCDA map.

Layer	No. of pixels changed	Total change from original (%)
A	99	17.9
B	354	63.9

The table above shows how many pixels has changed when creating alternate maps from the sensitivity analysis and comparing it to the original MCDA-map. Out of 554 pixels from the original MCDA-map, about 99 pixels has changed class value when removing the soil layer. This is about 17.9 % of the total area. Furthermore, it appears that all pixel value that changed, moved up 1 class value, indicating that the risk of subsidence overall increased when removing the soil criterion. Otherwise, there are no major changes removing the criterion.

On the other hand, removing the land use and land cover criterion changed the pixel distribution significantly where 354 out of 554 changed value. This makes about 63.9 % of the total study area. Most of the pixels has either moved 1 class up or 1 class down compared to the original MCDA-value. The majority of the pixels (340) moved one class down, which consist of 61.3 % of all pixels. Removing the land use and land cover has thus an opposite effect on the susceptibility map, making the study area appear much less prone to land subsidence. This is also the only case in the sensitivity analysis to significantly change the susceptibility map of the study area. This is also the only case where the easternmost part of the study area mostly consist of low or medium-low class pixels. The results from layer should on the other hand be interpreted with caution as it appears to be most inconsistent on assigning the weights to each criterion based on the calculated consistency ratio. As mentioned earlier, it did have an approximate consistency ratio of 0.09, barely passing the requirement of having a value under 0.1.

4.3 Machine learning

4.3.1 XGBoost and tuning the algorithms

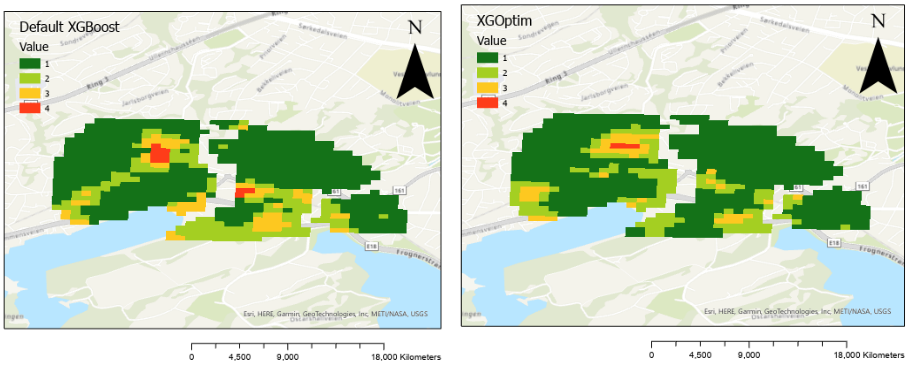
XGBoost was tuned based on testing a combination of hyperparameters as illustrated on table 14 below. Then, the combination that gave the highest accuracy score were considered to be the optimized version of the susceptibility map.

Table 14: The error metrics of XGBoost based on number of estimators, learning rate, and depth

XGBoost				
Estimator	**learning rate**	**Depth**	**Acc**	**AUC**
50	0.001	5	0.6596	0.6679
50	0.1	50	0.6596	0.7563
50	1	25	0.6330	0.7440
100	0.001	50	0.5957	0.7079
100	0.1	50	0.6143	0.8163
100	0.1	10	0.7128	0.8200
250	0.001	25	0.6383	0.7783
250	0.1	10	0.6649	0.7692
250	1	50	0.6755	0.7712

The accuracy score in XGBoost were within the range between 0.6 and 0.67, while the highest reached AUC-score mostly had values between 0.7 and 0.8, except for the set [estimator = 100, learning rate = 0.1, Depth = 10], which reached an AUC-score below 0.7. If the land subsidence map were created solely depending on accuracy score, the optimal set would be a tie between the sets [estimator = 250, learning rate = 1, Depth = 50] and [estimator = 100, learning rate = 1, Depth = 10] . This would however be the set with not the most optimal AUC-score. If the value with highest AUC-score were taken into consideration then the optimal set of parameters would be [estimator = 100, learning rate = 0.1, Depth = 50]. This set had however generated the lowest accuracy score. Furthermore, the learning rate tend to influence the AUC-score where a learning rate of 0.1 often gives good results regardless of number of estimator or depth. The most optimal values also appeared to favor sets with large depths, but the number of estimators did not matter much as any values of the estimator had the potential to generate decent AUC-scores as long as the other parameters are tuned well.

To summarize, the classifiers for XGBoost reached an optimized value where the AUC-score is substantially larger compared to their respective accuracy scores. The XGBoost appeared to yield far higher average AUC-scores when running the code 5 times. However, in some cases the AUC-score might even surpass 0.8.


Default XGBoost
Value
1
2
3
4
N
0 4,500 9,000 18,000 Kilometers
XGOptim
Value
1
2
3
4
N
0 4,500 9,000 18,000 Kilometers
Esri, HERE, Garmin, GeoTechnologies, Inc, METI/NASA, USGS

(a) Original XGBoost **(b)** Optimized XGBoost

Figure 31: The resulting land subsidence susceptibility maps generated from XGBoost by default (left) and the optimized result after tuning (right)

The same color scale for the initial InSAR-map were used for defining the results from machine learning techniques and figure 31 a and b shows the predictions from XGBoost before and after optimization. XGBoost appear to follow a similar pattern to the InSAR map in figure 22 even before tuning and optimizing the hyperparameters. It follows a pattern where there are a particular spot in the eastern part of the study area classified as high risk. Most values rame the same after optimization, but it appears that there are no longer a strip of medium-low class pixels that follows the red spot in the optimized XGBoost in figure 31b compared to 31a. The westernmost part mostly remains the same and it also applies for the pixels along water bodies.

Method	**Risk class**	**No. pixels**	**(%) of total area**
XGBoost	1	440	67.9
	2	137	21.1
	3	61	9.4
	4	10	1.6
XGBoost w. optimization	1	476	73.4
	2	127	19.7
	3	42	6.5
	4	3	0.4

Table 15: Distribution of pixels in the susceptibility maps from XGBoost. The classes follow the same subsidence rate ranges as in table 6 from chapter 3.6

The distribution of classes prior to and after optimization of XGBoost tend to not variate much. Both maps had the "Low"-class value as the most common pixel value (67.9 % and 73.4 % respectively). This also applies for the initial

InSAR-map. Both InSAR and XGBoost also did agree with the highest class being the least prevalent in the area, which is mostly concentrated on a spot in the eastern part of the area. XGBoost has managed to detect risk areas within the same area to some extent, but not on the same level as the target value map in figure 22. The target value map has detected far more medium-high pixels scattered on the western part as well and there are in fact more high risk areas defined in the western parts as well.

After tuning the parameters for XGBoost, the occurrences of the two highest class values has been reduced. There are otherwise no significant major changes as the lowest class value still dominate in the map, and the spot of high-risk pixels are still present on the same area of the map.

4.3.2 Hyperparameter tuning for neural networks

Tuning and tweaking the hyperparameters had the purpose of checking how certain parameters would affect the accuracy and the auc-score of the predictions. This is to see if the current land subsidence maps could be improvised and resemble a similar pattern as the target value. This were also the case for neural networks.

For the neural networks, the hyperparameters were tested manually using the combinations mentioned in section 3.8. The best performing networks are given in the table below, where the maximum accuracy score reaches 0.4938 when using sigmoid. There are however no clear tendencies of which hyperparameter combination of hyperparameters performed the best, since every possible combination at least appeared once in the list. Most of the accuracy scores and AUC-scores also remained stable with accuracy scores being around 0.5, and AUC-scores with approximately 0.62. This most likely indicates that the neural network algorithm do not perform as well as expected compared to XGBoost. In some cases, it appears that neural networks make random guesses when classifying pixel values. Moreover, the AUC-scores are mostly the same regardless of values except when the learning rate reaches 0.1 or higher.

Table 16: List of best performing accuracy and AUC-score by manual tuning. The algorithm was run 5 times and the table illustrates the average accuracy and AUC-score based on all the 5 runs.

Activation function	regularization	nr of neurons	Epochs	learning rate	AUC-score	Accuracy score
ReLU	0	5	500	0.001	0.4938	0.6468
Sigmoid	0.01	6	500	0.01	0.4900	0.6159
Leaky ReLU	0.001	2	1000	0.001	0.6285	0.6108
ReLU	0.001	6	1000	0.1	0.51	0.5
Sigmoid	0.001	5	500	0.001	0.65	0.6238

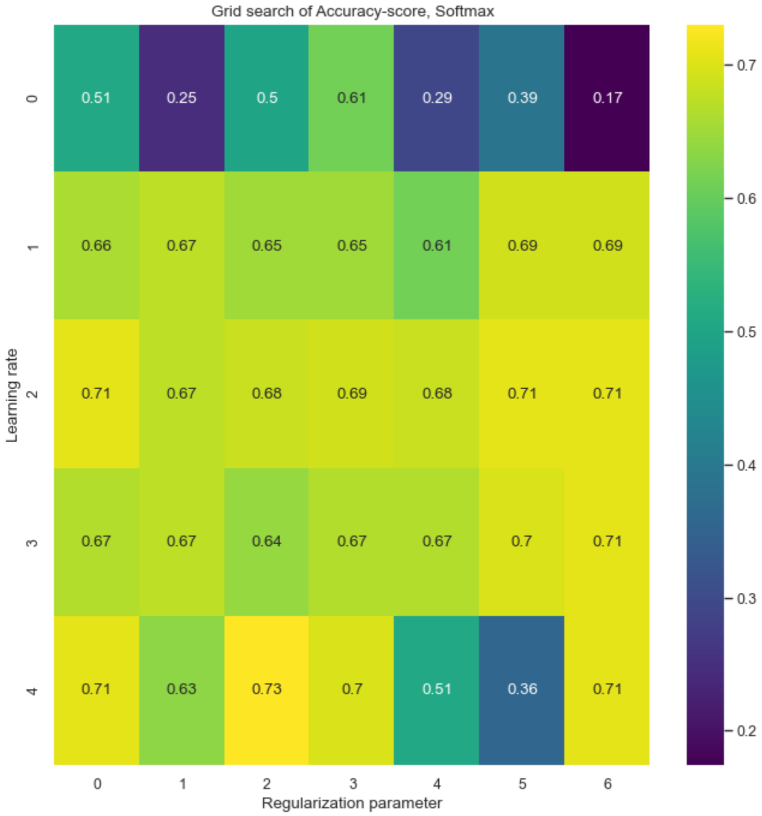


Figure 32: Grid search between learning rate and regularization parameter. Since the R2-score appears to flatten for larger epochs, the number of epochs is fixed at a value of 500.

In spite of having no clear solution of which hyperparameter combination yields optimal values, there were some slight indications of correlations between the AUC-score and the combination of learning rate and regularization parameter. A grid search matrix containing the AUC-score of each combination set of learning rate and regularization parameter were able to prove this. The rest of the parameters were given fixed values while different combinations of learning rates and regularization parameters were tested. The learning rate was checked for the values 0.00001, 0.0001, 0.001 , 0.01, and 0.1. The heat map in figure 32 have assigned the lowest value as 0 and the largest value as 4. The regularization parameter on the other hand was tested for 7 values, being: 0.00001, 0.0001, 0.001 , 0.01, 0.1, 1.0 and 10. Again, lowest value gets labeled as 0 and highest as 6. The optimal results were obtained in a learning rate between 0.001 and 0.01. Any smaller or larger learning rates drastically worsens the results. Within this range of learning rate, higher AUC-scores can be obtained when $\lambda \in [10^{-3}, 10^{-1}]$ where $\lambda = 10^{-2}$ contained the optimal value of approximately 0.6. It should be mentioned that the AUC-score based on the random search table generated slightly higher AUC-scores, which shows that the other hyperparameters have smaller influence when optimizing the values.

The AUC-values from the neural networks are still quite low in comparison to XGBoost, and an attempt to simplify the problem by only operating with binary classes has been tried. In this attempt, every subsidence value above -1.5 mm/year were assigned as risk class 0, and everything below are classified in risk class 1. This gave slightly higher average accuracy scores being about 0.8 after 30 runs. However, the AUC-scores remained mostly at the same value as for the original multiclass classification. It has therefore been decided not convert these classification back to ArcGIS Pro.

At its best, the neural network algorithm generated an AUC-score of 0.65 and in some cases, the AUC-score remained at 0.5, which means that the algorithm are guessing randomly.

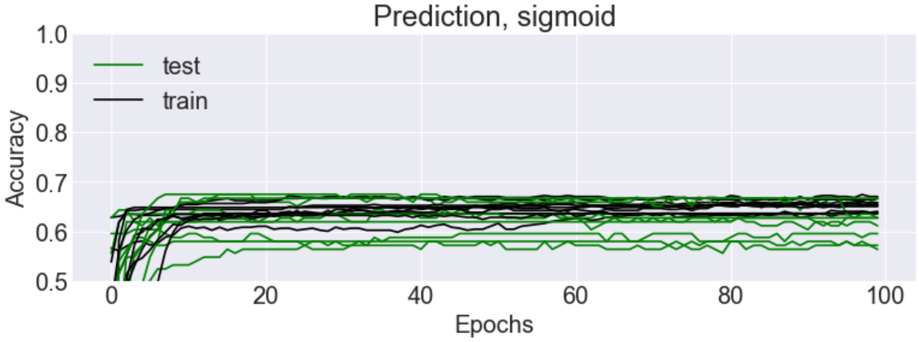
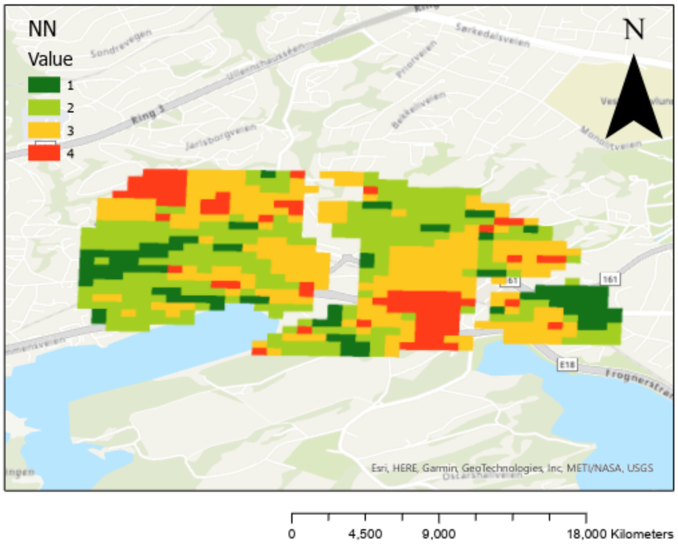


Figure 33: Plot showing the accuracy score with epochs. All five runs are showcased for both the training data and test data.

While studying the correlation between the accuracy score and the number of epochs in the five runs of the neural network algorithm, the accuracy score reaches its peak at approximately 10 epochs shown in figure 33. Some runs might even reach its peak earlier. The accuracy score flattens for all five cases the higher number of epochs used.

The map below shows land subsidence susceptibility with supposedly optimized parameters based upon the observed grid search map. The parameters on the last row on table 17 were used in the neural networks algorithm.



Esri, HERE, Garmin, GeoTechnologies, Inc, METI/NASA, USGS

Figure 34: Land subsidence susceptibility maps with tuning showing XGBoost (left) and neural network (right) respectively.

The resulting subsidence map after tuning a set of hyperparameters for the neural network algorithm managed to create more complex patterns with some similarities to the map from InSAR-data. It does rather have an inconsistent pattern compared to the XGBoost maps on figure 31 a and b. Most of the westernmost and easternmost part are however still classified as low-risk zones, and an overwhelming majority of the southwestern tip is classified as high-risk which deviate from the other maps such as XGBoost, InSAR and even MCDA. It also have far more outliers of "high"-risk pixels scattered around the map and the pattern seems to be randomly distributed.

Method	Risk class	No. pixels	(%) of total area
Neural network	1	87	13.4
	2	248	38.3
	3	235	36.2
	4	78	12.1

Table 17: Distribution of pixels after tuning.

The distribution of pixel values for the neural networks was also more evenly distributed among four classes according to table 17. The distribution of pixels happens to differ the most from the original InSAR-map. The intermediate-classes (class 2 and class 3) were the most common class value prior to tuning covering about half of the map layer, which is not the case for XGBoost.

4.4 Comparison between MCDA and machine learning

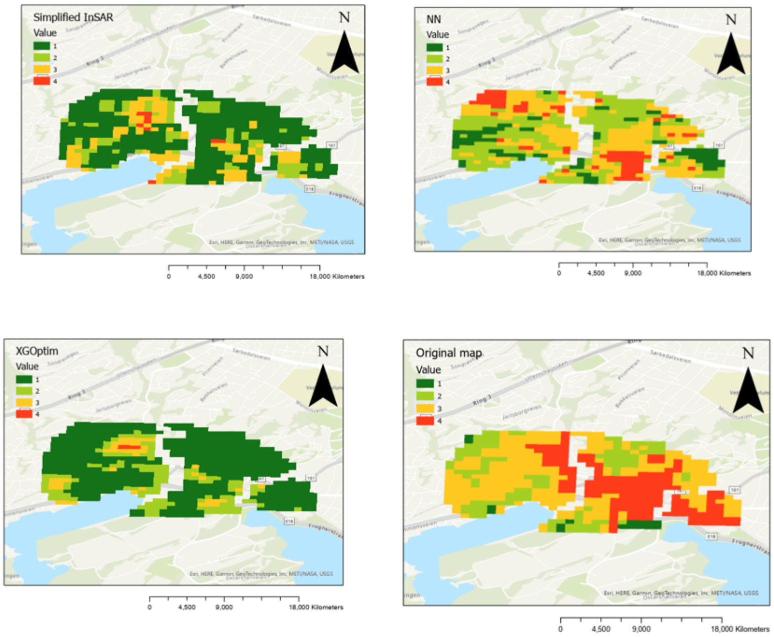


Figure 35: A recap of all the final four maps in the project. InSAR (top left), NN (top right), XGBoost (bottom left), and MCDA (bottom right). The map from MCDA were reclassified into three classes as mentioned in section 3.5

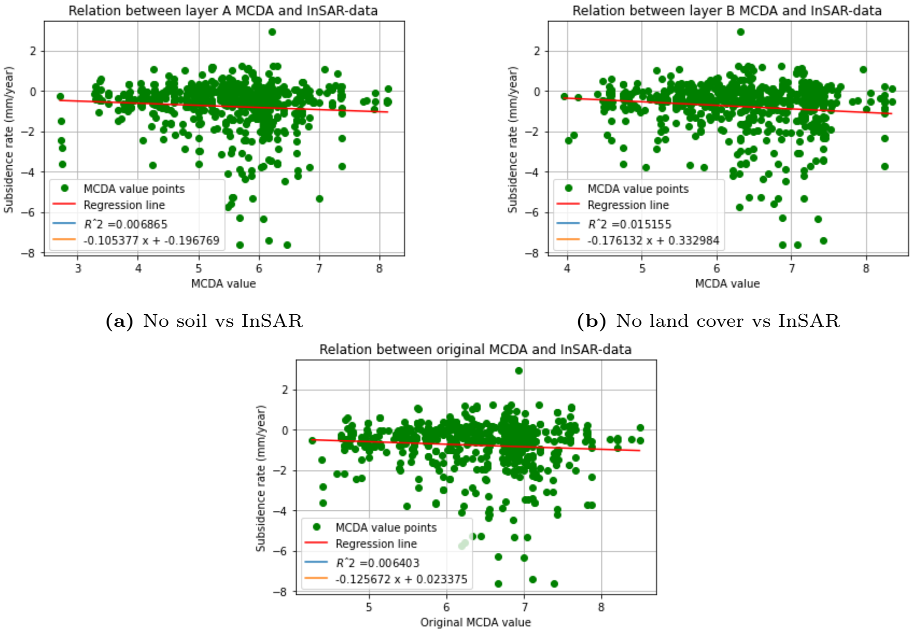
In short, all land subsidence susceptibility maps created in the project has shown certain patterns, but was not able to replicate the exact extent as the the InSAR-reference map as shown on figure 22. The results between the InSAR-generated map, MCDA and the machine learning algorithms tend to show quite different results, with the XGBoost algorithm being more similar to InSAR, and neural network being a combination between InSAR and MCDA since it has marked high risk spots on different sides of the river canal in the study area. The MCDA on the other hand had more defined shapes for each class with very few outliers. The medium-high and high-risk classes which are more frequent in MCDA and neural networks. It is difficult to observe similarities of all the four maps. While XGBoost and the InSAR-map point out the northwestern part of the area as the most susceptible to land subsidence, the neural network and MCDA has several high-risk zones across the study area. Not only did these maps also point out a part of the northwestern area as either medium-high or high risk, but also larger areas east from the major river. The maps from neural networks was however much more noisy compared to the maps generated from the other two. The rather large difference between MCDA and InSAR was expected since it were based more upon decision-making processes and expert judgement. Unlike neural networks and XGBoost, the MCDA-map did not use the original InSAR-map as reference point. The results of MCDA would thus be different as shown from the sensitivity analysis in section 4.3 since it heavily relied upon the decision-makers' preferences. The neural network map on the other hand did perform quite badly despite using the same inputs as XGBoost.

Table 18: Difference between MCDA and InSAR. Given the pixel class values for both map layers

Difference	**No. of pixels**
-3	87
-2	193
-1	143
0	69
1	20
2	5

When subtracting the map layer of InSAR with the MCDA subsidence map, only 69 pixels had the same value for both MCDA and the InSAR-reference map, which means that only 13.8 % of MCDA and InSARs map matched. The table shows that MCDA has a tendency overestimate the risk of subsidence compared to InSAR. Approximately 84% are classified 1, 2 or even 3 classes higher in MCDA than the InSAR map. The fact that 87 pixels are classified three classes higher means that MCDA had classified certain areas as high-risk classes despite the InSAR-map had labeled those in the lowest risk class.

This shows that there are significant gaps between those two map layers since the majority of the area in MCDA had different values than InSAR. The biggest difference is the easternmost part of the study area, where the MCDA-map almost classified the portion of the area as medium-high or high risk. Out of all maps, the original MCDA was the only map that classified most of the pixels adjacent to major rivers as either medium-high or high risk. This would also be the case for the other MCDA-maps generated from the sensitivity analysis.



(a) No soil vs InSAR

(b) No land cover vs InSAR

(c) Original MCDA vs InSAR

Figure 36: Regression analysis between InSAR and MCDA

Regression analysis was once more performed to see the fit between original InSAR-values and MCDA. In MCDA, the produced maps excluding soil or land cover data were also included since they had the most impact on the change of class distribution. Nevertheless, none of the maps fitted InSAR well enough with very many outliers. The original MCDA and the layer excluding soil data had R^2-values of 0.006 while the layer not including land cover did perform slightly better with 0.015. Based upon the regression analysis and earlier results, the MCDA map differs a lot from the initial InSAR-data that was used as output for the machine learning algorithms. This could explain why the maps generated different areas as high risk zones.

Table 19: Optimal accuracy of all three methods compared to InSAR

Accuracy score	
Neural network	**XGBoost**
0.6468	0.7128

5 Discussions

5.1 Distribution of high risk areas

The subsidence susceptibility maps from the results of MCDA had the tendency to create different patterns when the sensitivity analysis was performed. This lead to various uncertainties of the susceptibility assessment. Unlike machine learning methods, the assessment of MCDA was more or less based upon expert judgement and subjectivity to some extent. In addition, many assumptions had to made, such as ranking of criteria classes, dividing each criterion into sub-classes, as well as the break values between low susceptibility and high susceptibility. All these assumptions will affect the overall results of the land subsidence susceptibility analysis. This was also proven to be the case for other studies of susceptibility assessments for natural hazards. For instances, MCDA-GIS used by Rezaei et al. (2020a) to map land subsidence susceptibility in Neyshabur Plain, Iran gained less prediction accuracy than the other methods of Certain Factor, which relied less on decision making. Similarly, in Karlsson et al. (2017), the susceptibility assessment of natural hazards were partly inconsistent due to the experts participating in the decision making process had different opinions of how each criterion should be ranked. The inconsistencies did however partly disappear when applying AHP, which made the results perform equally well as weighting the criteria equally. All susceptibility maps created from MCDA fulfilled the requirements having a consistency ratio less than 0.1. The result with the highest consistency ratio value (i.e the map layer without land cover criterion in figure 29b) did however perform differently compared to the other susceptibility maps with slightly mpre of the area being classified in the two lowest categories. It was also the map with the highest number of pixels within the low category, deviating from the original MCDA-map by 10.11 % percentage points. When measuring the credibility of the weighting process in MCDA, it is important to keep the consistency ratio as low as possible. The closer the consistency ratio is to 0.1, the more likely it is for the judgement to be untrustworthy (Saaty, 1980). This is because the judgments made are too close for comfort randomness and it is then advisable for the ranking process to be repeated.

On the other hand, the results of MCDA also managed to illustrate how much a single criterion could impact the outcome of the susceptibility assessment. The land use and land cover (LULC) layer was considered the most important criterion for almost all the maps from MCDA. With the exception some slight changes in the high category, the pattern of the susceptibility map remained similar to the original even after decreasing the weights of LULC by 10 % shown from figure 28b. On the other hand, it was also observed that the MCDA-map with consistency ratio closest to 0.1 were in fact the map without taking LULC into consideration (fig. 29b), which had the most change during the sensitivity analysis. Land use and land cover often play a significant role for assessment of natural hazards (Meneses et al., 2019). LULC is considered as a dynamic variable that is driven by climate change and direct anthropogenic impact. Thus, land use and land cover are usually taken into consideration for susceptibility assessment of natural hazards since the type of land cover also could determine other factors such as slope stability or depletion of groundwater (Paola et al., 2013).

5.2 The machine learning algorithms

Neural network-based hyperparameter tuning was performed in order to identify the parameters with the greatest impact on the assessment of land subsidence. Even after tuning 4-5 hyperparameters, the performances of neural networks were still lower than expected since the accuracy often lies around 0.5 while the highest AUC-score barely surpassed 0.64. It is difficult to find an exact explanation for the results of the neural networks algorithm. One reason could be that there is simply not enough data in the area to ensure the algorithm could perform optimally. In fact, there are only 648 data points taken into consideration, and much more data point might be necessary to improve the AUC-score. Another reason could be structure of the algorithm itself. The algorithm in this project did not emphasize too much on the number of layers when tuning the parameters. Most of the runs did only contain a single layer. Although feedforward network with a single layer is sufficient enough to represent any function, the layer may be infeasibly large and may fail to learn and generalize correctly (Goodfellow et al., 2016). On the other hand, the neural networks algorithm still had higher accuracy than MCDA with AHP. The lower accuracy of MCDA with AHP compared to other studies is also notable in other previous studies of land subsidence assessments. For instances, in the study from the Neyshabur Plain, Iran, Rezaei et al. (2020b) concluded that the MCDA with AHP had slightly less prediction accuracy than the use of Certainty Factor with accuracies of 85 % and 90.2 % respectively. Subsequent R-index methods did also confirm that the Certainty Factor method had a better match with reality compared to AHP. It is however also important to note that the results from the MCDA-analysis in this project had much worse fit with InSAR as shown in figure 36 compared to neural networks which at least had an accuracy score of 63 %. The accuracy score of neural networks in this project is also considered to be quite low. One could argue that some of the criteria used in this project need to use more map layers and data related to certain conditions that causes land subsidence. Rezaei et al. (2020b) did to a higher extent use a more detailed and specified criteria towards hydrogeological conditions causing land subsidence. The criteria used for land subsidence predictions in Iran used for instances, alluvium thickness, saturation thickness and the thickness of compressive clay layers. Another study conducted in the Rafsanjan plain also included groundwater level decline as criteria, which was determined to be the parameter with the greatest impact on the assessment of land subsidence using artificial neural networks (Bagheri et al., 2019b). The correlation matrix from figure 16 confirms the criteria used in the project does not follow a certain pattern, which makes it more difficult to train the machine learning algorithm.

The performance of XGBoost did exceptionally well when calculating the AUC-score. Compared to neural networks and other traditional machine learning methods, it had faster calculation speed and stronger generalization. Similar studies of the Beijing plain were conducted by Shi et al. (2020) using linear regression, Random Forest, and XGBoost as a regression problem. Out of the three algorithms, XGBoost had the highest R^2-score with 0.9431 after optimization compared to 0.4234 using linear regression. The prediction effects of XGBoost were good and reasonable and had very few deviations. This were also the case for the study in Skøyen-area were the sample results from figure 24 mostly reflected the original map from the InSAR-layer. The use of InSAR as output data could also be debated upon since data could be exposed to atmospheric effects or vegetation as discussed in previously in the background chapter, but the study area are mostly urbanized and would thus have limited influence on the raw data from NGU.

Although various supervised machine learning methods have been used for mapping natural hazards before, the use of XGBoost algorithm is fairly new. Can et al. (2021) did use similar criteria as this study such as TWI, soil type and land cover for mapping landslide susceptibility in the upper basin of Ataturk dam, Turkey. The overall accuracy was 90.18 % with an AUC-score of 0.96. The input features from Can et al. (2021) are similar to this project by selecting effective factors that are highly dependent on geoenvironmental settings. Using multiple criteria related to topographic derivatives such as slope and TWI may have multicollinearity, which in turn could have affected the prediction. This should however not be the case for decision tree algorithms such as XGBoost as they usually are immune to these problems (Piramuthu, 2008). Multicollinearity can reduce the accuracy when making predictions and it should therefore be better to use independent variables that are not correlated or repetitive when building models, and could to some extent explain the performance of XGBoost. On the other hand, there are studies that had gained a more accurate model using neural networks when comparing XGBoost. Pradhan and Kim (2020) studied and mapped landslides in two catchments in South Korea and gained 76.73 % and 83.71 % accuracy from XGBoost and neural networks respectively. An explanation of this could be the use of different criteria or conditioning factors, but also due to the different qualities of DEM's used or the methods of hyperparameter optimization itself. The study area of the mentioned studies using XGBoost is also significantly larger than the studies itself. The results from these studies are therefore more representative for risk assessment in a regional scale. Another improvement could also be to first simplify the problem by only dividing the machine learning algorithm to only include two classes similarly to (Ayalew and Yamagishi, 2005). By only creating class 0 (No subsidence) and class 1 (Subsidence), it would be easier to know the threshold and assess the accuracy of the functions for both neural networks and XGBoost.

5.3 The study area and use of MCDA

The local area itself should also be taken into consideration when interpreting the resulting maps. The Skøyen-area, as discussed in the background-section, is heavily urbanized with few exceptions. In the maps from MCDA and neural networks, all of the high-risk areas were placed upon areas with tall buildings and dense infrastructure. Oslo in general has been rising from the sea since the last ice age due to isostatic uplift. Natural compaction with thick sets of marine and fluvial sediments might have an impact on subsidence to some extent. However, main settling can also occur within anthropogenic fill material to expand and areas. Such occurrences of subsidence had in fact affected infrastructure and buildings in central parts of Oslo (Eriksson et al., 2021). It has also been claimed in the report from Eriksson et al. (2021) that the development in central Oslo might have accelerated subsidence in the area by altering the groundwater table reducing the pore pressure in sediments. This might explain the pattern of the MCDA-map in this project as anthropogenic materials have been emphasized during the ranking process. Groundwater has also been included in the study. As mentioned earlier, the study area is also a connection hub for public transportation and there has been ongoing building projects currently (*Områderegulering for Skøyen*, 2021). This requires lot of deconstruction in the local area, and the results from MCDA with many high risk areas could be reasonable with respect to groundwater depletion. Furthermore, studies from Hakim et al. (2020a) using machine learning had shown a strong correlation between land subsidence and urban development, which was most likely related to excessive groundwater extraction within the urban areas.

On the other hand, it is still quite unclear why the map from MCDA differs from XGBoost despite using the exact same input criteria. The differences could be explained with MCDA being more based upon more on subjective decision-making. AHP was implemented in the analysis to make sure all decisions and rankings were more consistent. One could also argue that creating land subsidence maps through MCDA is more challenging due to its slow processes, similarities to other natural hazards or the fact that they tend to variate year by year depending groundwater depeletion relating to infrastructural projects Cui (2018). There is also a limited number of studies creating land subsidence through MCDA. Most of the previous MCDA-studies are conducted through multi-hazards analysis such as Karlsson (2016) or Skilodimou et al. (2019). In Skilodimou et al. (2019), the multihazard-analysis emphasized more on finding suitable places for urban development with respect to previous events of natural hazards in the area. The resulting MCDA-map of the Skøyen-area might have the potential showing "safe" areas for site planning, but would most likely not predict the true risk of land subsidence compared to maps from InSAR and machine learning itself.

Similarly from (Karlsson et al., 2017), the susceptibility analysis for natural hazards assessment in Swedish roads faced some uncertainties due to the nature of AHP modeling. The assessment was more or less subjective, and most of the experts' judgement were inconsistent from the beginning. Higher uncertainties also lies upon the break values used for dividing different classes of susceptibilities since the results of MCDA is a unitless value between 0 to 10. More studies need to be done before proper guidelines are developed to decide what should be considered low, medium-low, medium-high or high susceptibility. This is because any slight change in the way break values are selected for dividing the results would affect the percentage of pixels belonging to certain classes. MCDA is also highly dependent of defining main objectives, the region of interest and choosing criteria. Although small changes during the weighting process in the sensitivity analysis barely affect the overall ranking of each criterion, some areas would completely change in susceptibility by simply removing or swapping the rank of a criterion. More successful MCDAs focused to narrow down their respective study areas such as Gigović et al. (2019) and Rikalovic et al. (2014). The purpose of narrowing down the study area was to make sure that the choice of sites meet the basic criteria for their respective problems. This was to some extent also done in this project. The choice of study area was based on the subsidence activities from InSAR Norway and technical reports proving incidences of subsidence. However, this has also lead to some issues gathering enough data for the study area, which had some limitations regarding the choice of criteria.

5.4 Comments about choice of criteria

Land use and land cover were considered the most important criteria during MCDA. This criteria has commonly been used before as an important factor for predicting other types of natural hazards such as landslides (Meneses et al., 2019). The use of this criteria from earlier studies was argued due to the properties of data used produce land cover was closely integrated to landslide susceptibility models. The connection between land subsidence and and land cover is uncertain. Considering the study area mostly consist of urban fabric, there might also be some bias ranking the classes in the land cover map.

Bias could also have occurred in other criteria as well. For instances, the distribution of rock type is rather homogeneous mostly consisting of slate. The distribution were more diverse in other studies such as in Gigović et al. (2019), Rikalovic et al. (2014). Interestingly, the study of earthquake-induced natural hazards in China from Li et al. (2012) found that there are no significant connection between soil type and natural hazards occurring in the area. The purpose of including soil type in the analysis is because certain rock types such as sand-stone or phyllite are more likely to trigger subsidence, but there are however no such occurrences in the study area, and the structural characteristics could play a larger role affecting land subsidence. It could therefore be a discussion to include a map layer containing distance to faults as a criteria instead, but this was not included in the analysis due to lack of information and data access. The calculation of TWI and its spacial distribution of its values appear to be reasonable when creating the layer. It was based on a general method that would be valid for spacial distribution of soil, pH-value groundwater and soil moisture (Sørensen et al., 2006). This is however adapted towards Fennoscandian forests, and the author of this TWI-method could not identify a single best method that could suit all topographic conditions.

The altitude of groundwater table could also use some improvements regarding choice of interpolation method. The criteria layer of groundwater level were created by interpolating point data within the study area using. This was not too problematic as there were over 100 wells which are fairly distributed in the study area. Kriging as a geostatistical interpolation method was used by Modoni et al. (2013) when interpolating 200 water heads land in Bologna, Italy. This procedure was chosen to combine different information in the same geographical points, regardless of the positions they have been determined. This is however not much the case for Skøyen as a study area where none of the point data are located within the same positions.

From the sensitivity analysis, it could be discussed more whether or not the land use and land cover were weighted too much in the MCDA. When land use and land cover were removed as a criterion in figure 29b, a large portion of the westermost part of study area moved towards lower risk classes, while the red spot from InSAR in figure 22 were represented as medium-high risk. This is something expected as the majority of the land use and land cover layer consist of urban fabric, which had the highest values during the normalization process. The reasoning behind the weighting of land use and land cover is because intense urbanization has been documented to be among the land cover types that has caused land subsidence (Orhan, 2021). However, it could be argued that the analytical hierarchy process did have some form of bias towards land use and land cover, as it was evaluated to be much more important than all other criteria. AHP has a subjective nature, which means that methodology cannot guarantee the decisions are definitely true, and is more based upon probability and possibility measures (*WHY FUZZY ANALYTIC HIERARCHY PROCESS APPROACH FOR TRANSPORT PROBLEMS?*, 2011). This was evident from Karlsson et al. (2017), where more than thirty experts had different opinions which criterion are considered important when evaluating natural hazards. Some of the criterion could however be evaluated differently during the AHP. Orhan (2021) did also emphasize that groundwater usage is considered to be the most important parameters caused by anthropogenic conditions. This could mean that the GW-table criterion could in theory be ranked more important than land use and land cover during the ranking process or at least more equally important as land use and land cover since groundwater exploitation and building infrastructure often go hand in urbanization projects such as in building new railroads (Braathen, 2018) as explained in chapter 2.2. However, (Zeitoun and Wakshal, 2013) also mentioned that the materials on the subsurface often have much more impact the contribution of subsidence rather than the level of groundwater level alone, which is why the GW-table criterion was not ranked any higher during the analytical hierarchy process.

6 Conclusion

In conclusion, the main purpose of this study was to create land subsidence susceptibility maps using the methods of MCDA and the machine learning algorithms of neural networks and XGBoost within a heavily urbanized area in Skøyen to see how they compare to each other. Analytical hierarchy process was added in the MCDA process to increase the consistency in the decision making process. Sensitivity analysis were also applied to check the robustness of the MCDA-model and observe how the susceptibility map would change when slightly tweaking the weights of the criteria or removing one criterion altogether. Likewise, for the machine learning methods, hyperparameters were tuned in order to optimize the accuracy AUC-score of the performances. Seven criteria were used as input, namely: Land use and land cover, topographic wetness index, soil type, rock type, distance to river streams, slope and altitude to groundwater level. InSAR-data from NGU was used as output data for neural networks and XGBoost.

The results for MCDA and machine learning methods turn out to be substantially different from each other. While MCDA mostly had medium to high subsidence susceptibility scattered across the area, the initial InSAR-data and XGBoost mostly had low or medium-low pixels with the exception of the northwestern part of the area. Surprisingly enough, neural networks appear to generate a pattern in between MCDA and InSAR with a great mix of pixels from all classes with more outliers. The use of neural network is however the machine learning that generated the lowest accuracy score and AUC-score with optimized value of 0.6468 and 0.65 respectively after several runs. XGBoost performed much better with an optimized accuracy score of 0.7128 and an AUC-score of 0.82.

It is difficult to find an exact explanation why neural networks performed less than expected. One could argue that there simply are not enough data used in the study or there is a need to tune more hypermaterers in order to gain better results. The only pattern observed from neural networks is the fact that larger learning rates perform slightly better regardless of the other hyperparameters. There is a lot of potential to explore a larger set of hyperparameters to optimize the results, not only for neural networks, but also for XGBoost. It might also be beneficial to experiment on other activation functions or to use other type of gradient descent optimization algorithms such as AdaGrad or RMSProp. The AUC-scores remained mostly unchanged regardless of the set of hyperparameters used neural networks, but the accuracy score tend to vary. This showed how important it is to choose an error metric accordingly to the data set. Only small parts of the initial InSAR-data belong to the highest subsidence class, which in turn gives a large number of low- and medium-low class values. The issue of such class imbalance is that it could have resulted in a bias towards the majority class in XGBoost, reducing the classification performance. Thus, it was perhaps taken for granted that an unbalanced data set would not cause too much of an issue. A common remedy to handle imbalanced data set could for example be to use undersampling techniques where the distances between all instances of the majority class and minority class are calculated.

Considering the results of MCDA, it can be concluded that it requires more expert judgement in order to compete with machine learning methods to predict land subsidence. The regression analysis performed against initial InSAR-data showed very low R^2-values which indicates the results from MCDA do not coincide with InSAR-data. The sensitivity analysis had on the other hand shown consistency throughout the decision making process, there are some reasonable arguments why incidences of medium-high and high pixels are dominating from the MCDA-map since highly urbanized areas are often more vulnerable to land subsidence due to extraction of groundwater. This is however not verified in the area, and there are uncertainties of MCDA as a model itself. MCDA are more prone to subjective opinions and former studies such as Karlsson et al. (2017) has shown that even experts can disagree on how to rank chosen criteria properly. The choice of criteria and the ranking of those criteria were based solely upon the author of this project. An improvement could be consulting multiple experts and rank the respective criteria based on their expertise and calculate the average weights from them. Alternatively, there is a need for criteria more specified towards hydrogeological conditions such as rainfall, specific yield and aquifer type. The criteria chosen can easily be linked towards other kinds of natural hazards as studies of landslide uses similar criteria. In its current form, it might also be beneficial for the project to conduct studies in a more regional scale rather than a local scale or eventually perform MCDA in other urbanized areas where InSAR has shown to be strongly subsiding.

References

Andersen, R., Skepp, O., Sævold, H., Johansson, E., Lindsten, K., Berg, S. and Velstad, V. (2019). Regional kvikkleirekartlegging - Risiko for kvikkleireskred i Moss, Rygge og Råde kommuner, *Technical Report 1665114-1*, NVE, Oslo, Norway.

Arabameri, A., Saha, S., Roy, J., Tiefenbacher, J., Cerdà, A., Biggs, T., Pradhan, B., Thao, N. and Collins, A. (2020). A novel ensemble computational intelligence approach for the spatial prediction of land subsidence susceptibility, *Science of The Total Environment* **726**.

Ayalew, L. and Yamagishi, H. (2005). The application of GIS-based logistic regression for landslide susceptibility mapping in the Kakuda-Yahiko Mountains, Central Japan, *Geomorphology* **65**(1-2): 15–31.

Bagheri, M., Dehghani, M., Esmaeily, A. and Akbari, V. (2019a). Assessment of land subsidence using interferometric synthetic aperture radar time series analysis and artificial neural network in a geospatial information system: case study of rafsanjan plain, *Journal of Applied Remote Sensing* **13**: 1.

Bagheri, M., Dehghani, M., Esmaeily, A. and Akbari, V. (2019b). Assessment of land subsidence using interferometric synthetic aperture radar time series analysis and artificial neural network in a geospatial information system: case study of rafsanjan plain, *Journal of Applied Remote Sensing* **13**: 1.

Beiler, M. and Treat, C. (2014). Integrating gis and ahp to prioritize transportation infrastructure using sustainability metrics, *Journal of Infrastructure Systems* **21**: 04014053.

Binti Saiful Bahari, N. I., Ahmad, A. and Burhanuddin, M. (2014). Application of support vector machine for classification of multispectral data, *IOP Conference Series: Earth and Environmental Science* **20**: 012038.

Braathen, R. (2018). InterCity-prosjektet, Østfoldbanen, Fredrikstad - Sarpsborg Ingeniørgeologisk Rapport for Strekningen Rolvsøy - Klavestad, *Technical Report ICP-16-A-25510*, Bane Nor, Rolvsøy-Klavestad.

Buckley, S., Rosen, P., Hensley, S. and Tapley, B. (2003). Land subsidence in houston, texas, measured by radar interferometry and constrained by extensometers, *Journal of Geophysical Research* **108**.

Can, R., Kocaman, S. and Gokceoglu, C. (2021). A comprehensive assessment of xgboost algorithm for landslide susceptibility mapping in the upper basin of ataturk dam, turkey, *Applied Sciences* **11**.

Caruso, C. and Quarta, F. (1998). Interpolation methods comparison, *Computers and Mathematics with Applications* **35**: 109–126.

Chen, C.-H., Wang, C.-H., Hsu, Y.-J., Yu, S.-B. and Kuo, L.-C. (2010). Correlation between groundwater level and altitude variations in land subsidence area of the choshuichi alluvial fan, taiwan, *Engineering Geology - ENG GEOL* **115**: 122–131.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system, pp. 785–794.

Chen, W., Kornejady, A., Xie, X. and Pourghasemi, H. R. (2019). GIS-Based Landslide Susceptibility Evaluation Using Certainty Factor and Index of Entropy Ensembled with Alternating Decision Tree Models, *Natural Hazards GIS-based Spatial Modeling Using Data Mining Techniques*, number 48 in *Advances in Natural and Technological Hazards Research*, Springer Nature Switzerland AG, Cham, p. 311.
URL: *doi.org/10.1007/978-3-319-73383-8_10*

Chen, Y., Yu, J. and Khan, S. (2010). Spatial sensitivity analysis of multicriteria weights in gis-based land suitability evaluation, *Environmental Modelling & Software* **25**: 1582–1591.

Cui, Z.-D. (2018). *Land Subsidence Induced by the Engineering-Environmental Effect.*

Dai, F., Lee, C. and Zhang, X. (2001). GIS-based geo-environmental evaluation for urban land-use planning: a case study, *Engineering Geology* **61**(4): 357–271.

De Luca, M., Dell'Acqua, G. and Lamberti, R. (2012). High-speed rail track design using gis and multi-criteria analysis, *Procedia - Social and Behavioral Sciences* **54**: 608–617.

Dehghani, M., Rastegarfar, M., Ashrafi, R., Ghazipour, N. and Khorramrooz, R. (2014). Interferometric sar and geospatial techniques used for subsidence study in the rafsanjan plain, *American Journal of Environmental Engineering* **2014**: 32–40.

Domenico, P. A. and Mifflin, M. D. (1965). Water from low-permeability sediments and land subsidence, *Water Resources Research* **1**(4): 563–576.
URL: *https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/WR001i004p00563*

Eriksson, I., Borchgrevink, J., Sæther, M., Daviknes, H., Adamou, S. and Andresen, L. (2021). Oslo cost sub-urban wg1 report, *Technical Report TU1206-WG1-012*, European Cooperation in Science and Technology.

ESRI (n.d.). How flow accumulation works. File: `flowacc.png`.
URL: *https://pro.arcgis.com/en/pro-app/2.8/tool-reference/spatial-analyst/how-flow-accumulation-works.html*

Feizizadeh, B., Jankowski, P. and Blaschke, T. (2013). A gis based spatially-explicit sensitivity and uncertainty analysis approach for multi-criteria decision analysis, *Computers & Geosciences* **64**.

Ferretti, A., Prati, C. and Rocca, F. (2001). Permanent scatterers in sar interferometry., *Geoscience and Remote Sensing, IEEE Transactions on Geoscience and Remote Sensing* **39**: 8 - 20.

Galloway, D., Jones, D. and Ingebritsen, S. (1999). Land subsidence in the united states, *U.S. Geological Survey Circular* **USGS Circ. 1182**.

Gariano, S. L. and Guzzetti, F. (2016). Landslides in a changing climate, *Earth-Science Reviews* **162**.

Gaspar, J., Goulart, C., Queiroz, G., Silveira, D. and Gomes, A. (2004). Dynamic structure and data sets of a gis database for geological riskanalysis in the azores volcanic islands, *Natural Hazards and Earth System Science* **4**.

Ghorbanzadeh, O., Blaschke, T., Aryal, J. and Gholamnia, K. (2018). A new gis-based technique using an adaptive neuro-fuzzy inference system for land subsidence susceptibility mapping, *Journal of Spatial Science* .

Gigović, L., Drobnjak, S. and Pamučar, D. (2019). The Application of the Hybrid GIS Spatial multi-Criteria Decision Analysis Best-Worst methodology for Landslide Susceptibility Mapping, *ISPRS International Journal of Geo-Information* **8**(2): 79.

Glopper, R. D. and Ritzema, H. (2006). *Land subsidence*, ILRI, Wageningen, The Netherlands.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*, MIT Press. http://www.deeplearningbook.org.

GoogleDev (2020). Machine learning crash course - classification: Roc curve and auc. Accessed = 2021-09-30.
URL: *https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc*

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st edn, O'Reilly Media, Inc.

Hakim, W., Rizqiyanto Achmad, A. and Lee, C.-W. (2020a). Land subsidence susceptibility mapping in jakarta using functional and meta-ensemble machine learning algorithm based on time-series insar data, *Remote Sensing* **12**: 25.

Hakim, W., Rizqiyanto Achmad, A. and Lee, C.-W. (2020b). Land subsidence susceptibility mapping in jakarta using functional and meta-ensemble machine learning algorithm based on time-series insar data, *Remote Sensing* **12**: 25.

Huang, B., Shu, L. and Yang, Y. (2012). Groundwater Overexploitation Causing Land Subsidence: Hazard Risk Assessment Using Field Observation and Spatial Modelling, *Water Resour Manager* **26**: 4225-4239.

Kanungo, D., Arora, M., Sarkar, S. and Gupta, R. (2006). A comparative study of conventional, ann black box, fuzzy and combined neural and fuzzy weighting procedures for landslide susceptibility zonation in darjeeling himalayas, *Engineering Geology* **85**: 347-366.

Karlsson, C. S. (2016). *Geo-Environmental Considerations in Transport Infrastructure Planning*, PhD thesis, Department of Sustainable Development, Environmental Science and Engineering School of Architecture and the Built Environment Royal Institute of Technology (KTH), Stockholm, Sweden.
URL: *http://kth.diva-portal.org/smash/get/diva2:973817/FULLTEXT01.pdf*

Karlsson, C. S. J., Kalantari, Z., Mörtberg, U., Olofsson, B. and Lyon, S. W. (2017). Natural Hazard Susceptibility Assessment for Road Planning Using Spatial Multi-Criteria Analysis, *Environmental Management* **60**(5): 823-851.
URL: *http://link.springer.com/10.1007/s00267-017-0912-6*

Klemsdal, T. (2002). Landformer i Østfold, *Natur i Østfold* **21**(1/2): 7-31.

Lauknes, T. R., Dehls, J., Larsen, Y., Høgda, K. and Weydahl, D. (2006). A comparison of sbas and ps ers insar for subsidence monitoring in oslo, norway, *P.O. Box* **25**.

Lee, S., Park, I. and Choi, J.-K. (2012). Spatial Prediction of Ground Subsidence Susceptibility Using an Artificial Neural Network, *Environmental Management* **49**: 347-358.

Lee, S., Ryu, J.-H. and Park, H.-J. (2004). Determination and application of the weights for landslide susceptibility mapping using an artificial neural network, *Engineering Geology* **71**(3): 289 - 302.
URL: *http://www.sciencedirect.com/science/article/pii/S001379520300142X*

Li, Y., Chen, G., Tang, C., Zhou, G. and Zheng, L. (2012). Rainfall and earthquake-induced landslide susceptibility assessment using GIS and Artificial Neural Network, *Natural Hazards and Earth System Sciences* **12**(8): 2719-2729.

Malczewski, J. (1999a). Criterion Weighting, *GIS and multicriteria decision analysis*, John Wiley & Sons, Toronto, pp. 177-192.

Malczewski, J. (1999b). *GIS and multicriteria decision analysis*, John Wiley & Sons, Inc, New York.

Malczewski, J. and Rinner, C. (2015a). *Multiattribute Decision Analysis Methods*, pp. 81-121.

Malczewski, J. and Rinner, C. (2015b). *Multicriteria decision analysis in geographic information science*, Advances in geographic information science, Springer, New York. OCLC: 910069911.

McDonnell, R., Lloyd, C. and Burrough, P. (2015). *Principles of Geographical Information Systems*.

Meneses, B. M., Pereira, S. and Reis, E. (2019). Effects of different land use and land cover data on the landslide susceptibility zonation of road networks, *Natural Hazards and Earth System Sciences* **19**(3): 471-482.

Modoni, G., Darini, G., Spacagna, R. L., Saroli, M., Russo, G. and Croce, P. (2013). Spatial analysis of land subsidence induced by groundwater withdrawal, *Engineering Geology* **167**: 59-71.

NGU (n.d.). Reg. File: skoy$_a$*rea.png.* URL:*https://geo.ngu.no/kart/berggrunn_mobil/?lang=engg*

Nielsen, M. (2015). *Neural Networks and Deep Learning*, Determination Press.

Nordgulen, O., Lutro, O., Solli, A., Roberts, D. and Braathen, A. (1998). Geologisk og strukturgeologisk kartlegging for jernbaneverket: Utbygging i asker og bærum, *Technical Report 2755*, Norsk Geologisk Undersøkelse.

Oh, H.-J., Syifa, M., Lee, C.-W. and Lee, S. (2019). Land subsidence susceptibility mapping using bayesian, functional, and meta-ensemble machine learning models, *Applied Sciences* **9**: 1248.

Områderegulering for Skøyen (2021). *Technical report*, Plan- og bygningsetaten.

Orhan, O. (2021). Monitoring of land subsidence due to excessive groundwater extraction using small baseline subset technique in konya, turkey, *Environmental Monitoring and Assessment* **193**.

Paola, R., Busca, C., Mondini, A. and Rossi, M. (2013). The influence of land use change on landslide susceptibility zonation: The briga catchment test site (messina, italy), *Environmental management* **54**: 10573-.

Petersen-Perlman, J., Veilleux, J. and Wolf, A. (2017). International water conflict and cooperation: challenges and opportunities, *Water International* **42**: 1-16.

Piramuthu, S. (2008). Input data for decision trees, *Expert Systems with Applications* **34**: 1220-1226.

Pourghasemi, H. R. and Saravi, M. (2019). *Land-Subsidence Spatial Modeling Using the Random Forest Data-Mining Technique*, pp. 147-159.

Pradhan, A. and Kim, Y.-T. (2020). Rainfall-induced shallow landslide susceptibility mapping at two adjacent catchments using advanced machine learning algorithms, *International Journal of Geo-Information* **9**: 569.

Promper, C., Puissant, A., Malet, J. P. and Glade, T. (2014). Analysis of land cover changes in the past and the future as contribution to landslide risk scenarios, *Applied Geography* **53**: 11-19.

Rafiei Sardooi, E., Pourghasemi, H., Azareh, A., Sardoo, F. and Clague, J. (2021). Comparison of statistical and machine learning approaches in land subsidence modelling, *Geocarto International* pp. 1-41.

Rahmati, O., Golkarian, A., Biggs, T., Mohammadi, F., Keesstra, S. and Daliakopoulos, I. (2019). Land subsidence hazard modeling: Machine learning to identify predictors and the role of human activities, *Journal of Environmental Management* **236**: 466-480.

Rezaei, M., Noori, Z. and Dashti Barmaki, M. (2020a). Land subsidence susceptibility mapping using analytical hierarchy process (ahp) and certain factor (cf) models at neyshabur plain, iran, *Geocarto International* pp. 1-20.

Rezaei, M., Noori, Z. and Dashti Barmaki, M. (2020b). Land subsidence susceptibility mapping using analytical hierarchy process (ahp) and certain factor (cf) models at neyshabur plain, iran, *Geocarto International* pp. 1-20.

Rikalovic, A., Cosic, I. and Lazarevic, D. (2014). GIS Based Multi-Criteria Analysis for Industrial Site Selection, **69**: 1054-1063.

Romsenter, N. (n.d.). Skøyen innsynkning. File: Skoeyen-innsynkning.png.
URL: *https://www.romsenter.no/no/Media/Images/Jordobservasjon-Satellitter/Nedsynking/Skoeyen-innsynking*

Rød, J. K. (2015). *GIS: Verktøy for å forstå verden*, Fagbokforlaget, Trondheim.

Saaty, T. L. (1980). *The Analytical Hierarchy Process*, McGraw-Hill, New York.

Saaty, T. and Ozdemir, M. (2003). Why the magic number seven plus or minus two, *Mathematical and Computer Modelling* **38**(3): 233 - 244.
URL: *http://www.sciencedirect.com/science/article/pii/S0895717703900835*

Saaty, T. and Vargas, L. (2001). *Models, Methods, Concepts Applications of the Analytic Hierarchy Process*.

Scotto di Santolo, A., Forte, G., Falco, M. and Santo, A. (2016). Sinkhole risk assessment in the metropolitan area of napoli, italy, *Procedia Engineering* **158**: 458-463.

Sevil, J., Zarroca, M., Desir, G., Carbonel, D., Guerrero, J., Linares, R., Roqué, C. and Fabregat, I. (2017). Sinkhole investigation in an urban area by trenching in combination with gpr, ert and high-precision leveling. mantled evaporite karst of zaragoza city, ne spain, *Engineering Geology* **231**.

Shahabi, H., Jarihani, B. A., Tavakkoli Piralilou, S., Chittleborough, D., Avand, M. and Ghorbanzadeh, O. (2019). A semi-automated object-based gully networks detection using different machine learning models: A case study of bowen catchment, queensland, australia, *Sensors* **19**: 4893.

Shen, S.-L., Zhu, H. and Zhang, X.-L. (2004). Land subsidence due to groundwater drawdown in shanghai, *Geotechnique* **54**: 143-147.

Shi, L., Gong, H., Chen, B. and Zhou, C. (2020). Land subsidence prediction induced by multiple factors using machine learning method, *Remote Sensing* **12**: 4044.

Skilodimou, H., Bathrellos, G., Chousianitis, K., Youssef, A. and Pradhan, B. (2019). Multi-hazard assessment modeling via multi-criteria analysis and GIS: a case study, *Environmental Earth Sciences* **78**.

Sowter, A., Che Amat, A., Cigna, F., Marsh, S., Athab, A. and Alshammari, L. (2016). Mexico city land subsidence in 2014-2015 with sentinel-1 iw tops: results using the intermittent sbas (isbas) technique, *International Journal of Applied Earth Observation and Geoinformation* **52**.

Sui, D. (1994). Integrating neural networks with GIS for spatial decision making, **11**(2): 13-19.

Sørensen, R., Zinko, U. and Seibert, J. (2006). On the calculation of the topographic wetness index: Evaluation of different methods based on field observations, *Hydrology and Earth System Sciences* **10**: 101-112.

Tateosian, L. (2015). *Python For ArcGIS*, Springer International Publishing AG.

USGS (n.d.a). Landsat bands. File: screensar.png.
URL: *https://www.usgs.gov/programs/VHP/insar-satellite-based-technique-captures-overall-deformation-picture*

USGS (n.d.b). Landsat bands. File: landsat$_8$*.png.*
URL:*https://www.usgs.gov/media/images/landsat-8-band-designations*

Vegvesen, S. (2016). Erfaringsrapport bjørvika - bygging av gateanlegg i oslo bysentrum, *Technical Report 525*, Statens Vegvesen, region øst.

Wang, I.-T. (2019). Safety Assessment of Tunnel Portals for Site Selection based on Spatial Information Geoprocessing, *Infrastructures* **4**(4): 70 p.

Wang, Y., Fang, Z. and Hong, H. (2019). Comparison of convolutional neural networks for landslide susceptibility mapping in yanshan county, china, *Science of The Total Environment* **666**: 975-993.

WHY FUZZY ANALYTIC HIERARCHY PROCESS APPROACH FOR TRANSPORT PROBLEMS? (2011). ERSA.

Wichert, J. (2020). *Weathering of Slate*, Springer International Publishing, Cham, pp. 135-158.
URL: *https://doi.org/10.1007/978-3-030-35667-5*$_5$

XGBoostDev (2020). Introduction to boosted trees.
URL: *https://xgboost.readthedocs.io/en/stable/tutorials/model.html*

Yang, T.-L. and Gong, S.-L. (2010). Microscopic analysis of the engineering geological behavior of soft clay in shanghai, china, *Bulletin of Engineering Geology and the Environment* **69**: 607-615.

Yang, Y. (2015). Hotel location evaluation: A combination of machine learning tools and web gis, *International Journal of Hospitality Management* **47**.

Yeh, H.-F., Cheng, Y.-S., Lin, H.-I. and Lee, C.-H. (2016). Mapping groundwater recharge potential zone using a gis approach in hualian river, taiwan, *Sustainable Environment Research* **26**.

Yilmaz, I. and Ercanoglu, M. (2019). Landslide Inventory, Sampling and Effect of Sampling Strategies on Landslide Susceptibility/Hazard Modelling at a Glance, *Natural Hazards GIS-Based Spatial Modeling Using Data Mining Techniques*, Springer International Publishing, Cham, pp. 205-224. URL: *https://doi.org/10.1007/978-3-319-73383-8_9*

Yuan, L., Cui, Z.-D., Yang, J.-Q. and Jia, Y. (2020). Land subsidence induced by the engineering-environmental effect in shanghai, china, *Arabian Journal of Geosciences* **13**.

Zeitoun, D. G. and Wakshal, E. (2013). *The Subsidence Phenomenon Throughout the World*, pp. 9-23.

Zhou, J. and Civco, D. (1996). Using Genetic Learning Neural Networks for spatial Decision Making in GIS, *Photogrammetric Engineering and Remote Sensing* **62**.

Appendix A Processing raw data of MCDA, including formula for AHP

```
#!/usr/bin/env python
# coding: utf-8

# In[2]:


"""This is the entire script for the ordinary MCDA-analysis.
Some preprocessing of the criteria are not
included due to it being a pre-planning stage that is independent from MCDA.
All seven criteria exist in the work environment raw_data.gdb."""


import arcpy
from arcpy import env
from arcpy.sa import *

# Set up work-environment
arcpy.env.workspace = 'D:/Master_skøyen/skript/raw_data.gdb/'
mydir = arcpy.env.workspace
arcpy.env.overwriteOutput = True


# First we need to remove water from our dataset
# river is considered raster!!!
area = "skoyen2"
river = "river"
material = "Loose_material"
corine = "corine_land"
rock = "rock"
Bronn = "Bronn"

arcpy.Buffer_analysis("river", "D:/Master_skøyen/skript/raw_data.gdb/buffer",
                      "10 Meters")
buffer = "buffer"
arcpy.analysis.Erase(area, buffer, "clipped_area")
new_area = "clipped_area"

# Clip analysis of current raw data

Indata = [[new_area,material],[new_area,corine],[new_area,rock],
          [new_area,Bronn]]
```

```
Outdata = ["material_map","corine_map","rock_map","Bronn_map"]

for i,j in zip(Indata,Outdata):
    arcpy.analysis.Intersect(i,j)





# In[3]:


"""
Now, we create the water table altitude map
"""

dtm = "D:/Master_skøyen/Criteria/eksport_537364_20211226/dtm50/data/dtm50_6602_50m_33.dem"
outDEM = ExtractByMask(dtm,area)
inDEM = 'Altitude'
outDEM.save(inDEM)

ExtractValuesToPoints('Bronn_map', 'Altitude',
                      'alt_and_wells')



"""Must do polygontoraster before recassify"""
in_features = 'alt_and_wells'
field_name = 'water_altitude'

arcpy.AddField_management(in_features, "Raster",
                          field_type = 'DOUBLE')
arcpy.CalculateField_management(in_features, "Raster",
                                "!RASTERVALU!")

expression = '!Raster! - !vannstandb!'


arcpy.AddField_management(in_features, field_name,
                          field_type = 'DOUBLE')

arcpy.CalculateField_management(in_features, field_name,
                                expression)


input_points = 'alt_and_wells'
```

```
zField = 'water_altitude'
outlay = 'IDW_wells'
outRast = 'IDW_RastWells'
cellSize = 50
power = 2


# Set up variables for search neighborhood
majSemiaxis = 300000
minSemiaxis = 300000
angle = 0
maxNeighbors = 15
minNeighbors = 10
sectorType = 'ONE_SECTOR'

searchNeigborhood = arcpy.SearchNeighborhoodStandard(majSemiaxis,
                                                     minSemiaxis,angle,
                                                    maxNeighbors,minNeighbors,
                                                    sectorType)
# Execute IDW
arcpy.IDW_ga(input_points,zField,outlay,outRast)


# In[4]:



# Converting polygons to raster
arcpy.conversion.PolygonToRaster("material_map", "jorda_navn",
                                 "raster_material")
arcpy.conversion.PolygonToRaster("rock_map", "FID_rock", "raster_rock")
arcpy.conversion.PolygonToRaster("corine_map", "FID_corine_land",
                                 "raster_corine")


# In[6]:


"""
Now for reclassification
"""

inWell = "well_map"
inSlope = "slope_map"
inLand = "raster_corine"
inRock = "raster_rock"
```

```
inMaterial = "raster_material"
inRiver = "river_map"
inTWI = "twi_map"


# Remapping values
myRemapRiver = RemapRange([[0,110,10],[110,250,7],
                           [250,410,3],[410,6000,0]])

myRemapRock = RemapValue([[25,0],[190,2],[195,6],[206,6],[224,8],
                          [239,8],[372,10],[797,2]])

myRemapSlope = RemapRange([[0,3,10],[3,6,8],[6,9,6],
                           [9,12,3],[12,20,0]])

myRemapLand = RemapValue([[2,"NODATA"],[61,0],[120,8],
                          [36,4],[66,4],[74,10],[65,4],[62,10]])

myRemapTWI = RemapRange([[5,6.9,0],[6.9,8.1,3],
                         [8.1,9.7,6],[9.7,12.3,8],[12.3,30,10]])

myRemapMaterial = RemapValue([[41,10],[120,5],[70,7],[42,0]])

myRemapWell = RemapRange([[0,6.5,10],[6.5,12.7,8],
                          [12.7,18.6,6],[18.6,25.3,4],
                       [25.3,38.6,2],[38.6,40,0]])

# Reclassification
outReclassRiver = Reclassify(inRiver,"VALUE",myRemapRiver)
outReclassRock = Reclassify(inRock,"VALUE",myRemapRock)
outReclassSlope = Reclassify(inSlope,"VALUE",myRemapSlope)
outReclassLand = Reclassify(inLand,"VALUE",myRemapLand)
outReclassTWI = Reclassify(inTWI,"VALUE",myRemapTWI)
outReclassMaterial = Reclassify(inMaterial,"VALUE",myRemapMaterial)
outReclassWell = Reclassify(inWell,"VALUE",myRemapWell)

# Save reclass-files to geodatabase containing
outReclassRiver.save('D:/Master_skøyen/skript/vectorize.gdb/river')
outReclassRock.save('D:/Master_skøyen/skript/vectorize.gdb/rock')
outReclassSlope.save('D:/Master_skøyen/skript/vectorize.gdb/slope')
outReclassLand.save('D:/Master_skøyen/skript/vectorize.gdb/land')
outReclassTWI.save('D:/Master_skøyen/skript/vectorize.gdb/TWI')
outReclassMaterial.save('D:/Master_skøyen/skript/vectorize.gdb/material')
outReclassWell.save('D:/Master_skøyen/skript/vectorize.gdb/Well')
```

```
# In[ ]:


"""
For the next step, we need to apply Analytical Hierarchy process.
Now, we have to measure the relative importance of the criteria to each other.
Use the Python-module Pandas to read the Excel-file containing the
pairwise comparison matrix to calculate the results.
"""

def AHP(n, sheet_no):

    # n: No of criteria
    # sheet_no: Sheet name in Excel
    # RI: Random Index
    # CI: Consistency Index
    # CR: Consistency Ratio


    # Reading file location of the 7x7 pairwise comparison matrix
    file_loc = "C:/Users/rmbp/GIS-project/Excel-files/AHP.xlsx"
    df = pd.read_excel(file_loc,sheet_name=sheet_no)
    df1 = df.drop(["Letter"],axis=1)

    # Normalizing the nth root of products to get appropriate weights
    root = df1.product(axis = 1)**(1/n)
    sum_val = sum(root)

    # The final priority vectors
    weights = root/sum_val

    # Calculating the consistency ratio
    if n == 7:
        col_list = ["A","B","C","D","E","F","G"]
        RI = 1.32
    elif n == 6:
        col_list = ["A","B","C","D","E","F"]
        RI = 1.24

    col_sum = []
    for i in col_list:
        col_sum.append(df[i].sum())

    lam_max = np.sum(col_sum*weights)
    CI = (lam_max - n)/(n - 1)
    CR = CI/RI
    return weights, CR

# Original weights
W1, C1 = AHP(7,"Sheet2")

# Removal of DW
W2, C2 = AHP(6,"Sheet3")

# Removal of Lithology
W3, C3 = AHP(6,"Sheet4")

# Removal of land cover
W4, C4 = AHP(6,"Sheet5")


# Example printing results
print(W1)
print(C1,C2,C3,C4)
```

Appendix B Core algorithm of neural network classification

```
# -*- coding: utf-8 -*-
"""
Created on Mon Nov  2 12:35:56 2020
@author: rmbp
"""

import numpy as np


class Classify():

    def __init__(self,
                 hidden_activation="ReLU",
                 output_activation="softmax",
                 cost_func="cross_entropy"):

        self.h_a = hidden_activation
        self.o_a = output_activation
        self.cost = cost_func


    def hidden_activation(self,x,deriv=False):

        if self.h_a == 'ReLU':
            if deriv:
                return self._ReLU_deriv(x)
            else:
                return self._ReLU(x)
        elif self.h_a == 'sigmoid':
            if deriv:
                return self._sigmoid_deriv(x)
            else:
                return self._sigmoid(x)
        elif self.h_a == "leaky_ReLU":
            if deriv:
                return self._leaky_ReLU_deriv(x)
            else:
                return self._leaky_ReLU(x)

    def output_activation(self,x,deriv=False):
```

```
    if self.o_a == 'sigmoid':
        if deriv:
            return self._sigmoid_deriv(x)
        else:
            return self._sigmoid(x)

    if self.o_a == 'softmax':
        return self._softmax(x)

def output_error(self,a,t,x=None):
    if self.cost == 'cross_entropy':
        return (a-t)

def cost_function(self,a,t):
    if self.cost == 'cross_entropy':
        return self._cross_entropy_cost(a,t)

def _cross_entropy_cost(self,a,t):
    return -np.sum(np.nan_to_num(t*np.log(a)-(1-t)*np.log(1-a)))



# Again the activation function, now including softmax


_softmax = lambda self, x: np.exp(x)/np.sum(np.exp(x),
                                            axis=1, keepdims=True)
_sigmoid = lambda self, x: 1/(1+np.exp(-x))
_sigmoid_deriv = lambda self, x: self._sigmoid(x)*(1 - self._sigmoid(x))

_leaky_ReLU = lambda self, x: np.where(x > 0, x, x * 0.01)
_leaky_ReLU_deriv = lambda self, x: np.where(x > 0, 1, 0.01)

_ReLU = lambda self,x: np.where(x<0,0,x)
_ReLU_deriv = lambda self, x: np.where(x<0,0,1)
```

Appendix C Running neural network

```python
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 13 12:17:35 2022

@author: rmpay
"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns


from tqdm import tqdm

from sklearn.model_selection import cross_val_score, KFold, train_test_split
from sklearn.metrics import mean_squared_error as MSE, roc_auc_score as AUC_score
from sklearn import preprocessing

#from classification_problem import Classification
from Classify import Classify

mpl.rcdefaults()
plt.style.use('seaborn-darkgrid')
mpl.rcParams['figure.figsize'] = [10.0, 4.0]
mpl.rcParams['figure.dpi'] = 80
mpl.rcParams['savefig.dpi'] = 100
mpl.rcParams['font.size'] = 18

"""
The hidden layer and output activation functions
are previously defined in Classify.py
"""

class NeuralNetwork:

    def __init__(
            self,
            X_data,
            Y_data,
            problem,
            n_hidden_neurons_list =[2],
            n_output_neurons=2,
            epochs=10,
            batch_size=100,
            lr_rate=0.1,
            lmbd=0.0):

        self.X_data_full = X_data
        self.Y_data_full = Y_data

        self.n_inputs = X_data.shape[0]
        self.n_features = X_data.shape[1]
        self.n_layers = len(n_hidden_neurons_list)
        self.n_hidden_neurons_list = n_hidden_neurons_list
        self.n_output_neurons = n_output_neurons

        self.Problem = problem
        self.epochs = epochs
        self.batch_size = batch_size
        self.iterations = self.n_inputs // self.batch_size
        self.lr_rate = lr_rate
        self.lmbd = lmbd

        self.accuracy_train = np.zeros(epochs)
        self.accuracy_test = np.zeros(epochs)
        self.auc_train = np.zeros(epochs)
        self.auc_test = np.zeros(epochs)

        self.initialize_layers()

    def initialize_layers(self):
        n_hidden = self.n_hidden_neurons_list

        self.bias_list = [np.zeros(n)+0.01 for n in n_hidden]
        self.bias_list.append(np.zeros(self.n_output_neurons)+0.01)

        self.weights_list = [np.random.randn(self.n_features,n_hidden[0])]
        for i in range(1,self.n_layers):
            self.weights_list.append(np.random.randn(n_hidden[i-1],
                                                     n_hidden[i]))
        self.weights_list.append(np.random.randn(n_hidden[-1],
                                                 self.n_output_neurons))


    def FeedForward(self):

        problem = self.Problem
        self.a_list = [self.X_data]
        self.z_list = []

        for w,b in zip(self.weights_list,self.bias_list):

            self.z_list.append(np.matmul(self.a_list[-1],w)+b)
            self.a_list.append(problem.hidden_activation(self.z_list[-1]))

        self.a_list[-1] = problem.output_activation(self.z_list[-1])

    def FeedForward_out(self, X):
        problem = self.Problem
        a_list = [X]
        z_list = []

        for w,b in zip(self.weights_list,self.bias_list):
            z_list.append(np.matmul(a_list[-1],w)+b)
            a_list.append(problem.hidden_activation(z_list[-1]))

        a_list[-1] = problem.output_activation(z_list[-1])
        return a_list[-1]

    def Backpropagation(self):

        problem = self.Problem

        error_list = []; grad_w_list = []; grad_b_list = []

        output_error = problem.output_error(self.a_list[-1],self.Y_data)
        error_list.append(output_error)

        L = self.n_layers

        for l in range(2,L+2):
            prev_error = error_list[-1]
            prev_w = self.weights_list[-l+1]
            current_z = self.z_list[-l]
            error_hidden = np.matmul(
                prev_error,prev_w.T)*problem.hidden_activation(
                current_z,deriv=True)
            error_list.append(error_hidden)
        error_list.reverse()

        for l in range(L+1):
            grad_b_list.append(np.sum(error_list[l],axis=0))
            grad_w_list.append(np.matmul(self.a_list[l].T,error_list[l]))
            if self.lmbd > 0.0:
                grad_w_list[l] += self.lmbd * self.weights_list[l]

            self.weights_list[l] -= self.lr_rate*grad_w_list[l]
            self.bias_list[l] -= self.lr_rate*grad_b_list[l]

    def predict(self, X):
        probabilities = self.FeedForward_out(X)
        return np.argmax(probabilities, axis=1)

    def predict_proba(self, X):
        probabilities = self.FeedForward_out(X)
        return probabilities

    def SGD(self,auc=True):
        data_idx = np.arange(self.n_inputs)

        for i in range(self.epochs):
            for j in range(self.iterations):
                chosen_datapoints = np.random.choice(
                    data_idx, size=self.batch_size, replace=False
                )

                self.X_data = self.X_data_full[chosen_datapoints]
                self.Y_data = self.Y_data_full[chosen_datapoints]

                self.FeedForward()
                self.Backpropagation()

            pred_test = self.predict_proba(X_test_scaled)
            pred_train = self.predict_proba(X_train_scaled)
            self.accuracy_test[i] = accuracy_score(from_one_hot(Y_test),
                                                   np.argmax(pred_test,axis=1))
            self.accuracy_train[i] = accuracy_score(
                from_one_hot(self.Y_data_full),np.argmax(pred_train,axis=1))

            if auc==True:
                self.auc_test[i] = AUC_score(Y_test,
                                             pred_test,multi_class='ovr')
                self.auc_train[i] = AUC_score(Y_train,pred_train,
                                              multi_class='ovr')

def accuracy_score(Y_test, Y_pred):
    return np.sum(Y_test == Y_pred) / len(Y_test)
```

```
"""
Defining one-hot encoder
"""
def to_one_hot(category_array):
    ca = category_array # 1D array with values of the categories
    nr_categories = np.max(ca)+1
    nr_points = len(ca)
    one_hot = np.zeros((nr_points,nr_categories),dtype=int)
    one_hot[range(nr_points),ca] = 1
    return one_hot

def from_one_hot(one_hot_array):
    category_arr = np.nonzero(one_hot_array)[1]
    return category_arr


# Running the data

# Defining the data sets
df = pd.read_excel("D:/Master_skøyen/INSAR/only_two.xlsx")
df = df.dropna()


# Input and output data
input_data = df[["Lat","Long","Land_cover","River","Rock_type",
                 "Slope","TWI","Soil_type","GW_table"]].to_numpy()
output = df["classed_insar"].to_numpy(int)


output_one_hot = to_one_hot(output)



"""
Hyperparameters for tuning, runs decide how many times
the algorithm should perform
"""
hidden_neuron_list = [3,3,3]
epochs = 100
runs = 30
lr_rate = 0.001
lmbd = 0


# Storing values
```

```
AUC = []
accuracy = []

# Defining the parameters run for grid search
acc_test = np.zeros((runs,epochs))
acc_train = np.zeros((runs,epochs))
clf = Classify(hidden_activation="sigmoid",output_activation="softmax")

for i in tqdm(range(runs)):
    X_train, X_test, Y_train, Y_test = train_test_split(input_data,
                                                        output_one_hot,
                                                        test_size=0.2)
    Scaler = preprocessing.StandardScaler()
    X_train_scaled = Scaler.fit_transform(X_train)
    X_test_scaled = Scaler.transform(X_test)


    #Storing values for latitude and longitude
    X_train_feats = X_train_scaled.copy()
    X_test_feats = X_test_scaled.copy()
    latlong = X_train_feats[:,[0,1]]


    nn = NeuralNetwork( X_train_scaled,
                        Y_train,
                        problem = clf,
                        n_hidden_neurons_list=hidden_neuron_list,
                        n_output_neurons=2,
                        epochs=epochs,
                        batch_size=100,
                        lr_rate=lr_rate,
                        lmbd=lmbd)
    nn.SGD(auc=True)
    AUC.append(nn.auc_test[-1])
    accuracy.append(nn.accuracy_test[-1])
    acc_test[i,:] = nn.accuracy_test
    acc_train[i,:] = nn.accuracy_train


    hotter = from_one_hot(nn.predict_proba(X_test_scaled))

    #Saving predicted alues
    #np.savetxt("D:/Master_skøyen/skript/testnn.csv", hotter)
    #np.savetxt("D:/Master_skøyen/skript/latlong.csv",latlong, delimiter=',')
```

```
# Printing mean accuracy after n runs
AUC_mean = np.mean(AUC)
accuracy_mean = np.mean(accuracy)
print('AUC mean = ',AUC_mean, ' accuracy mean = ',accuracy_mean)

fig,ax = plt.subplots()
for i in range(len(acc_test)):
    ax.plot(acc_test[i],color='green',label='test')
    ax.plot(acc_train[i],color='black',label='train')
    if i == 0:
        ax.legend(loc=2)
        ax.set_ylim(0.5,1)
        plt.title("Prediction, sigmoid")
        ax.set_ylabel('Accuracy')
        ax.set_xlabel('Epochs')
        plt.tight_layout()



#Grid search
eta_vals = np.logspace(-5, -1, 5)
lmbd_vals = np.logspace(-5, 1, 7)
# store the models for later use
DNN_numpy = np.zeros((len(eta_vals), len(lmbd_vals)), dtype=object)


for i, lr_rate in enumerate(eta_vals):
    for j, lmbd in enumerate(lmbd_vals):
        nn = NeuralNetwork( X_train_scaled,
                        Y_train,
                        problem=clf,
                        n_hidden_neurons_list=hidden_neuron_list,
                        n_output_neurons=2,
                        epochs=epochs,
                        batch_size=100,
                        lr_rate=lr_rate,
                        lmbd=lmbd)
        nn.SGD(auc=False)

        DNN_numpy[i][j] = nn

        test_predict = nn.predict_proba(X_test_scaled)
        print("Learning rate  = ", lr_rate)
        print("Lambda = ", lmbd)
        print("Accuracy score on test set: ",
              accuracy_score(from_one_hot(Y_test),
                             np.argmax(test_predict,axis=1)))
        print()


# Creating a grid search matrix
sns.set()
test_accuracy = np.zeros((len(eta_vals), len(lmbd_vals)))
for i in range(len(eta_vals)):
    for j in range(len(lmbd_vals)):
        nn = DNN_numpy[i][j]

        test_predict = nn.predict_proba(X_test_scaled)
        test_accuracy[i][j] = accuracy_score(from_one_hot(Y_test),
                                             np.argmax(test_predict,axis=1))

fig, ax = plt.subplots(figsize = (10, 10))
sns.heatmap(test_accuracy, annot=True, ax=ax, cmap="viridis")
ax.set_title("Grid search of Accuracy-score, Softmax")
ax.set_ylabel("Learning rate")
ax.set_xlabel("Regularization parameter")
plt.show()
```

Appendix D XGBoost Grid search

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jan 11 12:15:12 2021

@author: rmbp
"""


from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, roc_auc_score as AUC_score
from sklearn import preprocessing

import numpy as np
import pandas as pd



# Opening the data set and dividing them to train and test set
df = pd.read_excel("D:/Master_skøyen/INSAR/all_data_content.xlsx")
df = df.dropna()
X = df[["Lat","Long","Land_cover","River","Rock_type","Slope",
        "TWI","Soil_type","GW_table"]].to_numpy()
y = df["classed_insar"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)

# Storing latitude and altitude for later, excluding them from analysis
latlong = X_test[:,[0,1]]

#Scaling the model
Scaler = preprocessing.StandardScaler()
X_train = Scaler.fit_transform(X_train)
X_test = Scaler.transform(X_test)


#Fitting the model
model = XGBClassifier(objective='multi:softprob',n_estimators=100,
                      learning_rate=0.1,max_depth=10)
model.fit(X_train,y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test,y_pred)
```

```python
#Making the prediction
y_pred_prob = model.predict_proba(X_test)
pred = y_pred[:,np.newaxis]
together = np.hstack((latlong,pred))


# Save the results for converting back to ArcGIS
np.savetxt("D:/Master_skøyen/INSAR/xgoptim.csv", together, delimiter=",")
AUC = AUC_score(y_test,y_pred_prob, multi_class="ovr")


# Printing the AUC and accuracy score
print("Accuracy: %.2f%%" % (accuracy * 100.0))
print("AUC: %.2f%%" % (AUC* 100.0))
```

Appendix E Regression analysis

```python
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 19 10:34:06 2022

@author: rmpay
"""

from scipy import stats
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

def rsquared(x,y):
    slope, intercept, r_value, p_value, std_err = stats.linregress(x,y)
    return r_value**2



df = pd.read_excel("D:/Master_skøyen/INSAR/Regression.xlsx","to_regression")
df = df.dropna(how='any',axis=0)


x = df["sense_nosoil"].values
y = df["InSARdata_Resample"].values

gradient, intercept, r_value, p_value, std_err = stats.linregress(x,y)

mn = np.min(x)
mx = np.max(x)
x1 = np.linspace(mn,mx,100)
y1 = gradient*x1+intercept


plt.plot(x,y,"og",label="MCDA value points",markersize=6)
plt.plot(x1,y1,"r-",label="Regression line" )
plt.xlabel("MCDA value")
plt.ylabel("Subsidence rate (mm/year)")
plt.grid("on")
plt.title("Relation between layer B MCDA and InSAR-data")
plt.plot([],[],"" ,label= r'$R^2$ =%f' %rsquared(x,y))
plt.plot([],[],"",label= '%f x + %f' %(gradient,intercept))
plt.legend(loc=0,prop={'size': 10})
```

Appendix F Code for showing distribution of data

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime
import time


import matplotlib as mpl
import seaborn as sns
from sklearn.model_selection import cross_val_score, KFold, train_test_split
from sklearn.metrics import mean_squared_error as MSE
from sklearn import preprocessing
from sklearn.linear_model import Lasso, LinearRegression, Ridge
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import r2_score
from tqdm import trange


# Showing distribution of data to the depression data set
df = pd.read_excel("D:/Master_skøyen/INSAR/all_data_content.xlsx")

X = df[["Land_cover","River","Rock_type","Slope","TWI",
        "Soil_type","GW_table","classed_insar"]]
y = df["classed_insar"]



# Creating heat map
plt.figure(figsize=(9, 9))
correlation = X.corr()
heatmap = sns.heatmap(correlation, annot=True,fmt='.2f')
plt.title("Correlation matrix of InSAR")
plt.show()

#Plotting density of each criterion
for i in X:
```

```
plt.title(i)
sns.set_style('darkgrid')
sns.distplot(df[i])
plt.figure()
```