TO DREDGE OR NOT TO DREDGE

Data-driven feature engineering of side channels

GEOIT1501: Synthesis Project

2025-11-12

Michel Beeren
Geomatics
TU Delft
5422833

Luc Jonker
Geomatics
TU Delft
4836111

Yair Roorda
Geomatics
TU Delft
5467543

Vincent Vanderheeren
Geomatics
TU Delft
5376203

TU Delft Van Oord Marine ingenuity

To Dredge or not To Dredge

Data-driven feature engineering of side channels

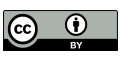
Michel Beeren
Geomatics
TU Delft
5422833

Luc Jonker
Geomatics
TU Delft
4836111

Yair Roorda
Geomatics
TU Delft
5467543

Vincent Vanderheeren
Geomatics
TU Delft
5376203

Supervisors: Edward Verbree (TU Delft), Martijn Meijers (TU Delft), Pam Sterkman (Van Oord), Irene Pleizier (Van Oord)
Date: 2025-11-12
Faculty: Faculty of Architecture and the Built Environment, Delft



The cover page of the report showcases an exaggerated visualisation of a point cloud representing the bottom of a side channel

Abstract

To help prevent flooding of rivers and cities, Dutch maritime contractor Van Oord regularly dredged 52 side channels as part of the Dutch Department of Waterways and Public Works' (Rijkswaterstaat) "Room for Rivers" strategy. Side channels make rivers more resilient to flooding by providing increased flow capacity, buffer space, and a secondary path downstream for water. Van Oord wishes to know how they can better leverage their growing historical data collection to enable predictive maintenance of side channels in the form of dredging. Instead of developing a complex hydrological model, which would require deep knowledge of river morphology. We, as Geomatics students, extracted insights directly from the available geospatial data. For our 10-week MSc Geomatics Synthesis Project, our main research question is as follows: ***How can the features of a side channel be identified and extracted to enable predictive maintenance?***

In order to answer this question for our client Van Oord, we performed a literature review and interviewed domain experts to identify relevant characteristics of side channels. Then, we explored the available geo-spatial data to determine which characteristics can be modeled as features, before processing the data in an FME pipeline to calculate these feature values in an automated, extendible, and understandable way. These features were then stored in a geo-spatial database. Reading from this database, we created a prototype machine learning model that takes the features as input. The model enables analysis of the side channels to derive insights into the sedimentation of side channels, reaching 84% accuracy within a 5cm error for the Bakenhof channel.

The result is a robust FME-based data processing pipeline, a geo-spatial database with 19 unique features for 26 suitable side channels, and a prototype neural network showing significant predictive ability. The product enables the client to better estimate side channel behavior, enabling informed predictive maintenance, as well as allowing the client to better decide moments when expensive channel measurements can be skipped.

Acknowledgments

We would like to sincerely thank the team at Van Oord for their generous support throughout this project. We are especially grateful to them for allowing us to work with their data and their consistent willingness to assist us whenever needed.

In particular, we thank Pam Sterkman for acting as a sponsor for this project, providing guidance, organizing trips, and connecting us with many experts; Irene Pleizier for her role in providing this project and representing the interests of the GIS Team; Jan Span for making the project possible for us to contribute towards; Marten Sein for sharing data and guiding us in interpreting it; Etienne de Jong for his help with GIS-related questions; Ton van de Sande for his advice on side channel–related topics; Ondrej Urban for his insights into predictive modeling of morphological behavior; and Laetitia de Boer for her assistance with watershed surveys.

We are grateful to our supervisors, Edward Verbree and Martijn Meijers, for their valuable feedback, suggestions, contacts, and literature that greatly supported our work.

We would also like to give our thanks to fellow students Phuong Anh Ho, Lars van Blokland, Alexandre Bry, Daan Schlosser, and Mingjie Teo for peer reviewing our draft report.

Finally, we thank the external experts who kindly shared their knowledge and experience: Denise Thus, for providing us with valuable information on morphological behavior in side channels; Pepijn van Denderen, for his insights into our project design and feature selection; and Hans van der Kwast, for his guidance on spatial dynamic modeling and the use of QGIS for hydrological applications.

The students declare that they have not used generative AI tools in the preparation of this assignment. Specifically, AI tools *were* used for proof reading and suggesting improvements to the language choice in the report, as well as for some Python code assistance, but *not* for generating research data or drawing conclusions. All intellectual and creative work, including data analysis and interpretation, is original and conducted by the students without AI assistance.

Contents

List of Figures

List of Tables

1 Introduction

Global climate change is a driving force behind many of the challenges we face both today and in the future. One such challenge is increasingly extreme weather (Ripple et al., 2025). Such weather causes peak rainfall to increase and, at the same time, periods of drought to intensify. For many coastal and water-rich parts of the planet, this will result in an increased risk of flooding (Mahato et al., 2022).

To continue to protect its citizens, the Dutch Department of Waterways and Public Works (RWS) has developed the "Space for Rivers" (Ruimte voor de rivieren) strategy. The strategy involves the creation and maintenance of side channels as one of the ways to protect the lowlands from fluvial flooding (Ministerie van Infrastructuur en Waterstaat, 2025). Within the context of the strategy, over twenty man-made side channels have been constructed since 1996 (van Denderen et al., 2019).

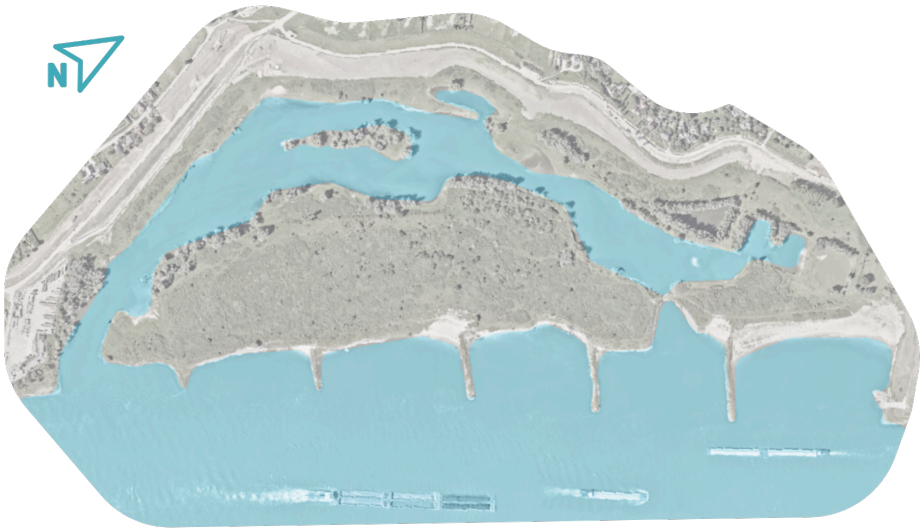


Figure 1. Example of a typical side channel, located near Passewaaij in the Netherlands (own figure)

Figure 1 shows how these side channels branch off from main channels, like the Lek and Waal rivers. They increase the flow capacity of these rivers during high water by providing a secondary path downstream. They also provide a buffer space for the river to overflow and store water during extreme weather. These properties improve the river's ability to both transport and store water, decreasing the likelihood and severity of fluvial (river) flooding.

Besides playing an important role in the flood safety of the Netherlands, side channels also fulfill an important ecological function (Simons et al., 2001). The unique conditions they provide create diverse and rare habitats for many species of animals and plants (Vendrig, 2001).

However, over time, natural processes slowly change the morphology of the side channel. If these processes run unchecked, the side channel can stop flowing as designed. This can reduce the intended benefits of flood prevention and impact the flora and fauna in and around the channel.

The high-level processes that impact the morphology are sedimentation and erosion. Sedimentation (deposition of sediment flowing in from the main channel) happens more frequently in a side channel than in the river itself due to the comparatively low flow rate inside the side channels (Hai et al., 2017; van Denderen et al., 2019). This causes certain areas to silt up and hinder the flow. Erosion occurs primarily on banks and can be a result of waves generated by passing ships (Meijer & Winden, 2020).

To manage these processes in side channels, dredging operations are periodically performed to ensure that the channels can continue to fulfill their intended purpose. These dredging operations are costly and time-consuming projects that are performed by specialized companies that have access to the required staff and equipment.

Van Oord is one such company, responsible for the maintenance of over 50 of these side channels along the Dutch section of the Rhine delta as part of the WOCU Rijntakken project. For monitoring the channels, they regularly map the riverbed using a multibeam echo sounder (MBES) (Figure 2) mounted to a boat or unmanned surface vehicle (USV) (Figure 3).

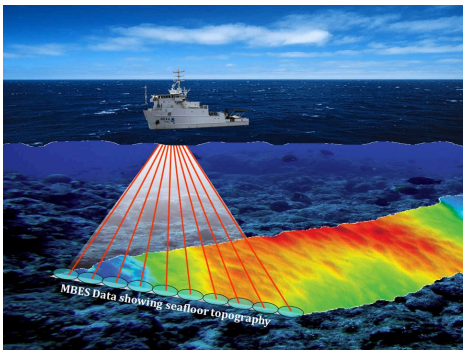


Figure 2. MBES measurement (NOAA Photo Library, 2016)

Figure 3. Example USV used by Van Oord, via www.vanoord.com[1]

One motivation for collecting these measurements is the possibility of combining them with other available data and using this information to enable analysis of side channel sedimentation over time. This could be done in the form of a predictive model. Such a model would be a valuable tool for data-driven analysis of the channels, which could help inform the development of a more efficient predictive maintenance workflow. This would lower operation costs as well as assist in a more simplified and accurate decision-making process (Zonta et al., 2020).

Traditionally, to predict how hydrological phenomena such as sedimentation develop, one might task a hydrological engineer with in-depth knowledge of the mathematical formulae that model these processes to create or use a water simulation tool in the hopes of predicting how the side channel may behave over time. These simulations can be quite insightful, but often make assumptions for the purpose of keeping the computation feasible that can lead to them degrading in accuracy.

In this study, we tackle the problem the other way around. That is to say, we work from *actual* measurements taken from river side channels, and use the data itself to extract patterns and behaviors to perform prediction. This enables accurate prediction without needing in-depth knowledge of the actual physical processes that are taking place. In order to enable this strategy, we first must choose how we encode the side channel characteristics in the form of features.

Thus, the goal of this project is to use the available data to design and evaluate a set of *features* – as defined in Section 2.6 – that describe side channels in a way machine learning algorithms can interpret. These features should be designed to encode as many important aspects of side channel sedimentation as possible. Once a meaningful set of features has been established, they form the basis for any number of analytical or predictive modeling approaches. The outputs of which can give us insight into which features are most valuable and which are potentially still missing. Based on the encoded features, we develop a working prototype of such a predictive model that can be iterated upon by the client.

This report is made up of eight chapters. Chapter two outlines various pieces of useful theory and context to assist in understanding both the problem and our solution. Chapter three outlines the problem in more depth, describing the concrete research question and sub-questions. Chapter four goes into detail about the methodology of the research. Chapter five covers the various datasets used during the project. Chapter six presents our results. Chapter seven contains a discussion of the results of the research, as well as future work and limitations. Finally, chapter eight presents the conclusions drawn from the project.

[1] https://www.vanoord.com/nl/updates/van-oord-schaft-tweede-onbemande-surveyschip-aan/

2 Theory and Context

This chapter describes relevant theory that will assist in understanding the context of the research, as well as the developed solution. For a glossary of relevant theoretical terms and abbreviations, see Appendix A.

2.1 Client

Van Oord is a Dutch marine contractor with a global presence, specializing in complex maritime projects. The company's core activities include dredging and marine construction, land reclamation, coastal defense, and the development of ports and waterways.

With a history stretching over 155 years, Van Oord is a privately owned, family-run business known for its large and specialized fleet of vessels and its focus on innovative and sustainable solutions for a variety of marine and infrastructure challenges around the world.

2.2 Study Area

Van Oord has been selected as a partner by RWS to work on the 'Waardengedreven OnderhoudsContract Uiterwaarden' (WOCU) Rijntakken project. This project group focuses on maintaining water safety, quality, and nature management in various areas along the Wal, Lek, and Nederrijn rivers. The project is managed and funded by RWS (Team Rijntakken, 2023) in cooperation with partners like Van Oord. The internal working group within Van Oord for the WOCU project is the project provider. Within the WOCU project area are a number of river side channels (Figure 4), which are the subject of this research.

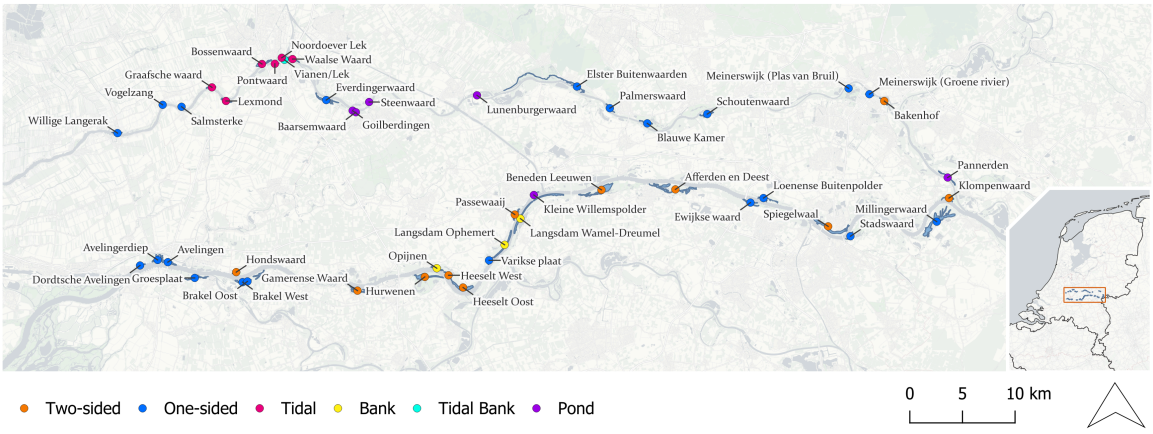


Figure 4. Side channels managed by Van Oord (own figure)

2.3 Side Channel Dredging

Figure 5. Dredging techniques, via paansvanoord.com[2]

Figure 5 shows a vessel used by companies like Van Oord to perform dredging. Broadly speaking, a specialized vessel is brought out to the site, which then proceeds to remove sediment and debris from the river bed using large suction pipes, cranes, etc. This sediment is then moved elsewhere to be stored/disposed of.

In the Netherlands, dredging is required on a regular basis to maintain rivers and coastlines. Like rivers, side channels also periodically need to be dredged to ensure they continue to operate as intended.

While it heavily depends on the project, in general, dredging is a costly operation. Especially for side channels that may be difficult to reach from the main river, in which case the dredging vessel may need to be brought in using a crane. Operators, coordinators, and various other specialists are required to ensure the successful execution of a dredging project.

In the context of the WOCU Rijntakken project, Rijkswaterstaat sets guidelines for the side channels that companies like our client, Van Oord, have to adhere to by performing the dredging operations when they become necessary. At the moment, deciding when they are necessary is a game of analyzing the current measurements carefully, discussing with experts, and attempting to determine if channels are at risk of breaching the guidelines set by RWS. A data-driven predictive analysis tool like the one being developed with this project can assist these experts in more efficiently determining when a channel might be at risk.

2.4 Types of Side Channels

Side channels are a feature of rivers in which a smaller portion of the water can diverge from the river's main channel. Side channels can either occur naturally via hydrological phenomena or be man-made through processes like dredging and digging. Side channels can be broadly categorized as one-sided connected, two-sided connected, or ponds, as shown in Figure 6. Additionally, a channel can be a bank channel or a tidal channel. Each of these types has unique behavior and characteristics that set them apart from other side channels. That being said, it is important to emphasize that side channels are each individually unique and complex hydrological features, thus some may be a part of more than one type (e.g. a bank channel can also be tidal).

[2] https://www.paansvanoord.com/activiteiten

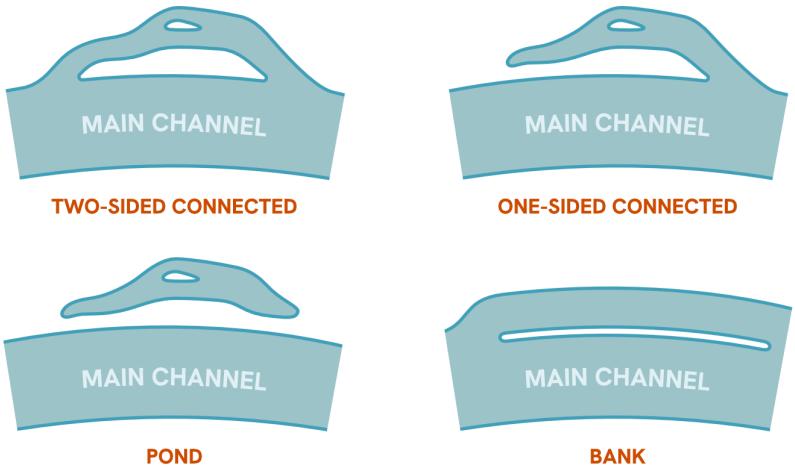


Figure 6. Different types of side channel present in the dataset (own figure)

1) Two-sided connected: Has both an inlet and an outlet, which allows for some amount of co-flow (water flowing from the river, through the channel, and returning back to the river) throughout the year. The amount of water flowing consistently is usually quite low (1-3%). However, at higher discharge levels, the threshold height can be reached, allowing for more co-flow (Meijer & Winden, 2020).
2) One-sided connected: Has only an inlet *or* outlet (usually just an outlet), and thus does not co-flow unless the water threshold height is reached.
3) Pond: This type of side channel is, at least for part of the year, not connected to the main channel at all. It can even fall dry (Meijer & Winden, 2020). Similar to one-sided connected channels, at higher discharge levels, a channel of this type can co-flow with the main channel.
4) Bank: A bank is a unique form of side channel that is situated along the main channel, separated from it by a man-made longitudinal dam, called a 'langsdam' in Dutch.
5) Tidal: A tidal channel is affected by tidal forces. That is to say that the water height may increase or decrease depending on the tides, and the direction of water flow can also change accordingly. They do not have a distinct shape and could look like any of the types showcased in Figure 6.

2.5 Characteristics

For this report, a side channel characteristic is any identified quality of the side channel that may impact how the channel develops. For example, this could be the length of the channel, the depth at a given point, the inlet structure, soil types, vegetation in and around the channel, velocity of the flow, etc.

These are physical features in the real world, and may interact with each other in deeply complicated ways. For example, Shield's formula describes the stability of granular material in running water as affected by the critical bottom shear stress, density of the sediment, density of the water, gravity, and diameter of the sediment, which could all individually be considered characteristics (Shields, 1936).

A list of identified characteristics can be found in Appendix B, with Section 4.5 outlining characteristics that are modeled in our product in the form of features.

2.6 Features

In the fields of machine learning and pattern recognition, or more broadly artificial intelligence, a feature is defined as an individual measurable property or characteristic of a data set (Bishop, 2006). They are the foundational attributes used to describe a system or phenomenon that an algorithm can use to make a prediction or analysis.

Features come in many different forms (binary, numerical, ordinal…) and describe some specific property of a system or phenomenon. For example, a person's height or eye color would be features of that individual. For side channels, other features like lengths, depth, and shape would be relevant. An important distinction between features and characteristics is that features are modeled independently.

When we calculate features like flow velocity and channel length, we do *not* explicitly model any relationship between those two, even if such a relationship does exist in reality. This is because the idea behind machine learning is, in simple terms, to *learn* such relationships without needing to program them yourself.

2.7 Feature Vector

A feature vector represents an "observation" where each element in this vector represents a single feature, as shown in Figure 7. This vector represents an encoding of measured characteristics into a form digestible by a computational model or algorithm (Hastie et al., 2017).

For a classic machine learning application like a spam filter, a feature vector of an email might include the word count, the sender's email address, the distribution of certain characters, etc. Then, machine learning can be leveraged to identify relationships and patterns in the existing data to try and determine if a new, not yet seen email should be considered spam.

When developing a model to analyze side channel sedimentation, a feature vector of one channel might include width, channel length, sediment concentration, flow rate, etc. The combination of features allows for the application of mathematical and statistical methods, enabling data analysis and machine learning workflows.

2.8 Feature Space

A so-called "feature space" is the conceptual multi-dimensional space where each data point is represented as a vector, as shown in Figure 7. Each feature is represented as a single dimension of this vector. Embedding the measurements of a system or phenomenon in feature space enables various types of fundamental machine learning capabilities. Examples include supervised learning techniques like regression or classification, and unsupervised learning techniques like clustering.

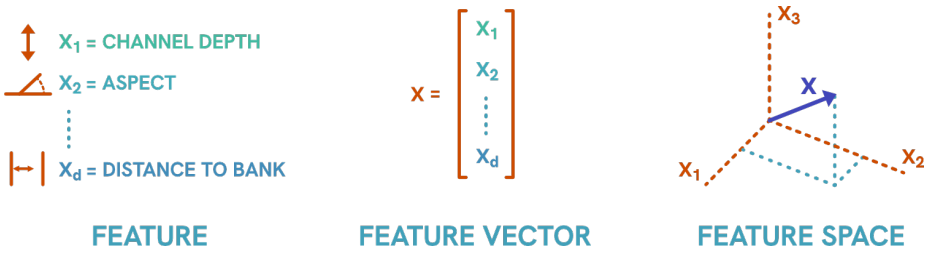


Figure 7. Features, Feature vectors, and feature space (own figure)

2.9 Feature Engineering

Feature engineering is the process of transforming raw measurements of a system into features that represent the underlying system in a way computers can access. Carefully selecting a set of features is an essential part of any machine learning approach. It is a foundational process that fits into a larger data-driven workflow, as shown in Figure 8. Often, this process even has more effect on model performance than the choice of algorithm itself (Domingos, 2012).

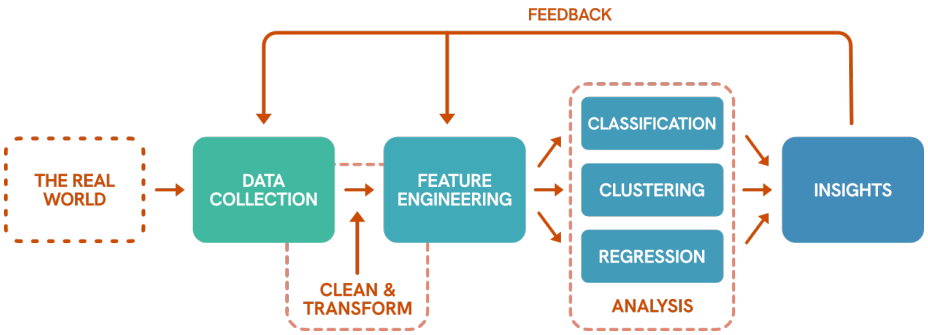


Figure 8. Data-driven analysis workflow, with feature engineering linking data with analysis (own figure)

Carefully evaluating and selecting features gives us crucial insight into which measurements capture the most meaningful information about the system we are attempting to model. This analysis can

further inform strategies for allocating resources to certain measurements or data collection methods. This may include recommendations for additional measurements to collect currently unavailable data.

Another major benefit of explicit feature engineering is the explainability of the resulting machine learning models. Unlike deep-learning approaches that often function as a “black box” (Sarker, 2021), explicit feature engineering allows for a selection of features that is more meaningful and interpretable. This explainability can then inform management decisions and data collection methods.

High-quality features are also a way to overcome other limitations of deep-learning techniques. Feature engineering enables more robust prediction and analysis on limited datasets. Deep-learning techniques can potentially extract more intricate patterns, but they require much larger datasets and more computational power (Sarker, 2021) for optimal results, neither of which we had access to in the context of this project.

2.10 Multi-layer Perceptron

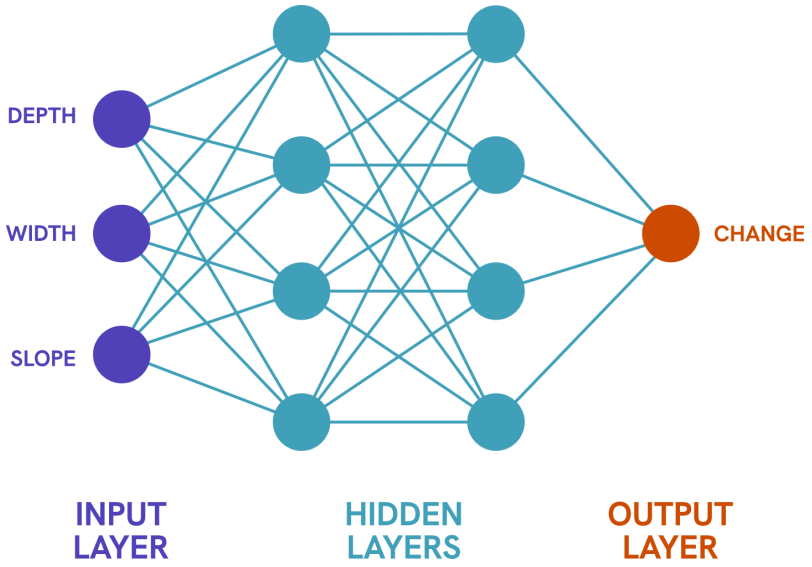


Figure 9. Structure of a multilayer perceptron (own figure)

A multi-layer perceptron (MLP) takes a set of values as input and then transforms them into output values by passing them through several hidden layers of interconnected neurons. Figure 9 shows a common diagram for the structure of an MLP. A neuron takes in values, multiplies them by the learned weights, adds a bias, and applies a non-linear activation function to capture complex relationships. Every training step (epoch), the weights and biases get incrementally updated using backpropagation to minimize the loss function. It can be trained on features to perform some kind of prediction, in our case, taking side channel features as input and predicting the sedimentation.

3 Problem Definition

3.1 Problem Summary

The sedimentation process necessitates that companies regularly dredge side channels to prevent them from closing. Determining when dredging is needed involves the collection of bathymetric measurements and expert analysis of those measurements, both of which cost time and money.

To better understand sedimentation of side channels, Van Oord now measures them yearly within the Rijntakken — the river network Van Oord is responsible for maintaining. So far, 26 side channels in this river network have been measured using a Multibeam Echo Sounder (MBES) in both 2024 and 2025.

By precisely mapping the side channels, Van Oord aims to move towards a predictive maintenance workflow. This approach should reduce maintenance costs (Zonta et al., 2020) by, for example, identifying which channels are less dynamic and therefore require less frequent surveying, or by assessing whether it would be more efficient to carry out preventive dredging in certain channels, and by streamlining the analysis process for experts by providing a prediction of future behavior.

While we have received extensive data related to these side channels, our temporal resolution is limited to only two years. This limitation prevents the use of approaches such as deep learning to predict sedimentation. Still, we wish to extract as much useful information as we can from the available data and any conclusions that can be derived from it. This means the main goal of this project is to determine *how we can describe and digitally encode meaningful features of a side channel using the limited available data*.

When the side channels are digitally encoded in feature space, this will enable more detailed numerical and/or machine learning based analysis. This could include clustering, classification, and potentially even regression-based approaches to predict sedimentation of side channels. Compared to traditional GIS analysis approaches, this can allow for instantaneous extraction of insights from the available data without any manual effort beyond adding the data to the database via our feature extraction pipeline. Thus, our secondary goal is to explore *how analysis of the encoded side channels can inform both data collection and viability for use in a predictive model to predict sedimentation.*

The outcome of the research will be a set of features that encode side channel sedimentation, along with the processing pipeline that produced them. Both of which Van Oord can build upon in the future as more data becomes available. These features will also be used to develop a proof-of-concept predictive model. Finally, based on the prototyped analysis of the features, this project will provide recommendations on the continued collection of data and potentially new collection of data.

By choosing to implement hand-crafted features and perform feature selection, the entire process becomes explainable in a way that is difficult to replicate with a deep learning approach. This allows us – in addition to simply performing prediction – to also give informed recommendations on data that could be collected in the future to improve performance. The prediction itself also becomes easier to interpret and argue for to stakeholders, given that the features are rooted in theory.

3.2 Research Questions

Following the problem definition, the main research question is:

How can features of a side channel be identified and extracted to enable predictive maintenance?

To comprehensively answer our main research question, it will be broken up into four sub-questions, which will be tackled separately.

What characteristics play a role in the sedimentation of side channels?

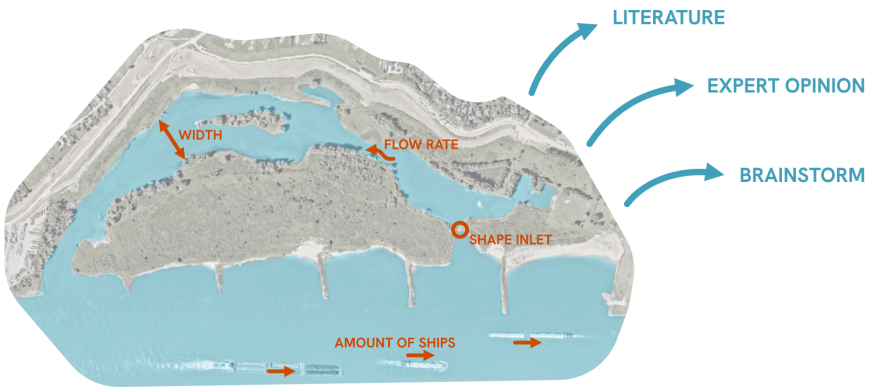


Figure 10. Example of feature identification (own figure)

To begin to develop a set of meaningful features, it is prudent to create an extensive list of characteristics that influence side channel sedimentation. By identifying which characteristics have the most impact on channel morphology, we can better prioritize which features to develop and integrate into the final model. Figure 10 showcases how such identification is based on a combination of literature study, brainstorms, and expert interviews. Once we have performed a thorough analysis, we will have a foundation from which to inform the creation of features.

What side channel features can we represent using the data we have?

Figure 11. Example of feature extraction using tools like FME, QGIS, and custom code (own figure)

Building upon the list of characteristics influencing side channel sedimentation, we will translate characteristics into concrete data based on their expected impact and effort. We have been given several datasets and will need to analyze this data to determine what features we can extract, what features we can calculate, and what features are missing. For the client, it would be ideal if the process by which features are extracted from data is as automated as possible. It thus becomes important to emphasize the creation of a pipeline that takes in “raw” data and associates it with the correct side channel, and encodes as many suitable features as possible, such as those shown in Figure 11. That way, it only needs to be set up once, and for each year it can be rerun at the click of a button, with minimal manual effort. Additionally, the processes we set up must be understandable and extendable for the client after the project has ended. This will mean investing time in high-quality code, documentation, and scalable solutions.

How can we use the encoded features to design a predictive model of future side channel sedimentation?

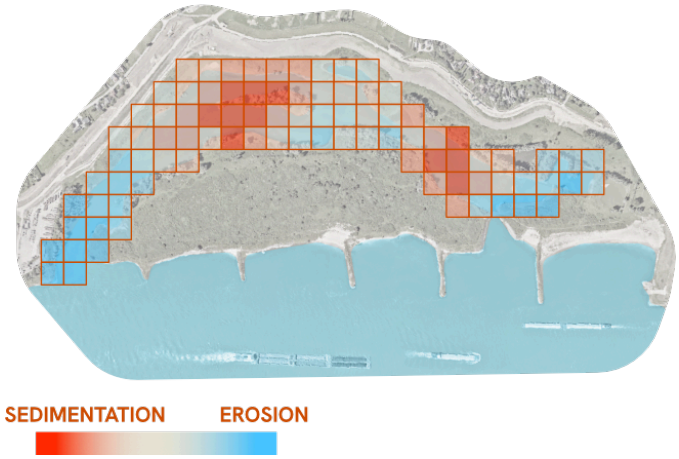


Figure 12. Example of the prediction map showing how much sedimentation or erosion occurs (own figure)

Side channels encoded in feature space provide a strong foundation for predictive modeling, for example, in the form of a machine learning pipeline. While in the temporal dimension, we currently have too few MBES measurements for a highly confident prediction, it is still valuable to prototype the predictive modeling process to investigate the predictive power of the features we do have. The outcomes of analysis, such as a sedimentation map like shown in Figure 12, could inform predictive maintenance, future data collection, etc. Furthermore, we expect that as Van Oord collects more data over time, it can be easily integrated into this system to increase accuracy and reliability.

How can we analyze the encoded features to determine driving characteristics in side channel sedimentation?

Figure 13. Example of an error map, in which pink represents areas where the prediction is lower than reality, and green represents areas where the prediction is higher than reality (own figure)

With encoded features, we can investigate the possibilities for analysis to determine, for example, which features are the most important. Do these align with what the literature describes? Are there features missing from measurements that may lead to more robust analysis?

Figure 13 showcases an error map that could be used to inform future development of the prediction pipeline. By seeing where the model currently struggles, one could direct the development of new features to try and fill gaps in the model.

4 Methodology

4.1 Overview

The project is divided into different topics, and within each topic, is further divided into corresponding tasks which have been ranked according to the MoSCoW method (must have, should have, could have, and won't have) (see Appendix D.3). The different topics are then considered in descending order of importance, starting with the must-haves.

This manner of working ensured that the team focused its efforts initially on high-priority tasks to create a minimum viable product. Additional elements were then added to the project based on their importance and remaining time.

Figure 14 shows the main steps of the project with their processes and results.

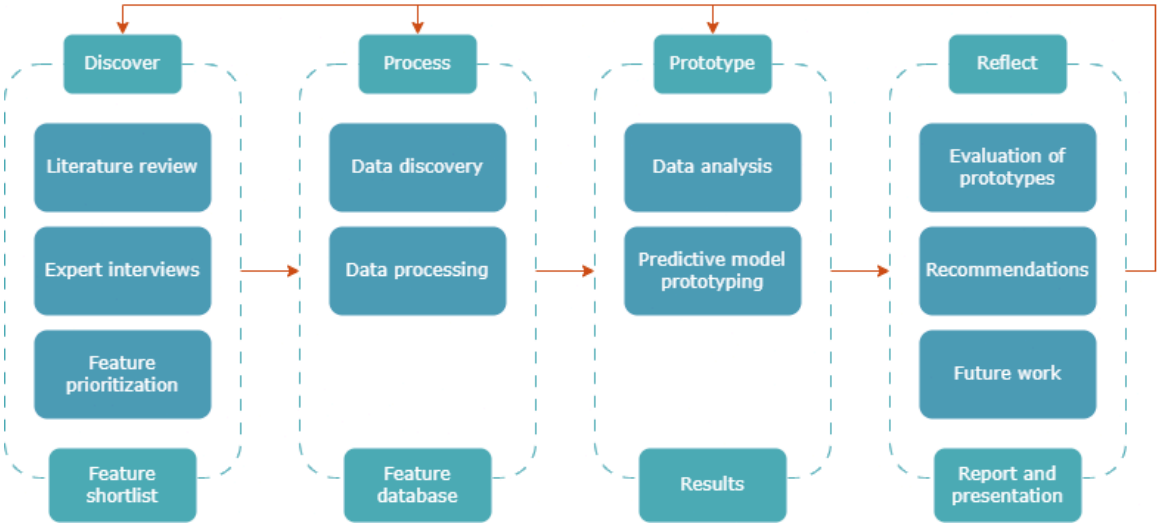


Figure 14. Flow chart outlining project methodology (own figure)

4.2 Discover

4.2.1 Literature Review

This step involved reviewing existing academic papers, reports, case studies, etc. This helped us build a theoretical framework to understand what has already been studied. For discovering side channel characteristics, we focused on papers that analyze side channel morphology, and searched within those papers for characteristics that were described as impactful. We utilized *saturation* to determine when to stop with the literature review. That is, papers were reviewed, and the mentioned characteristics were noted down. Each consecutive source added fewer novel – and more repeated – characteristics. The investigation stopped once no additional characteristics were discovered in the final paper.

The literature review identified a large number of side channel characteristics (see Table 1). The original complete list of identified characteristics is available in Appendix B.

Table 1
Number of new and repeated characteristics extracted from examined literature

Reference	New characteristics	Repeat characteristics
Thus 2025	12	0
van Denderen et al. 2025	11	6
Mosselman 2001	8	6
Riquiter et al 2017	5	8
Denderen et al. 2017	3	9
Meijer & Winden 2020	4	2
Ottevanger & Chavarrias 2019	0	4

With the list of associated features, we categorized them based on the concept of an action-priority matrix (see Figure 15). Characteristics were given a label of high, medium, or low impact; as well as extreme, high, medium, and low effort. The categorization was based on literature review and expert interviews (to determine impact), and the data availability, data quality, measurement techniques, and expected difficulty of implementation (to determine effort).

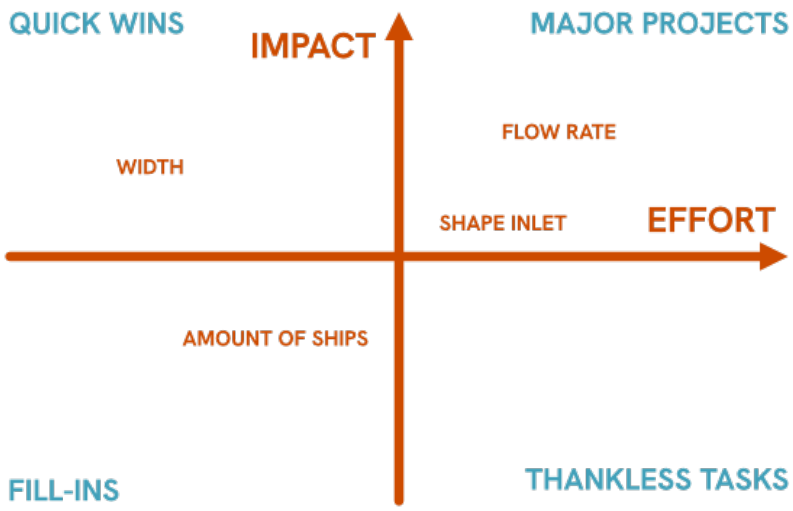


Figure 15. Example of action-priority matrix (own figure)

The combination of these two axes allowed us to identify high-priority features for implementation (see Table 2).

Table 2
Action priority matrix for discovered features

	low impact	mid impact	high impact	Total
low effort	0	2	7	9
mid effort	1	4	8	13
high effort	2	4	4	10
extreme effort	0	4	6	10
Total	3	14	25	**42**

4.2.2 Expert Interviews

To validate our findings, we had meetings with experts knowledgeable on relevant concepts such as side channels, predictive modeling, and data collection/processing. We discussed our identified characteristics, as well as potential associated features, with domain experts Pepijn van Denderen (who has performed extensive research on river morphology, morphodynamics, side channels, etc.) and Hans van der Kwast (a physical geographer focused on GIS and remote sensing, who has created the widely used QGIS plugin PCRaster).

Their inputs into the compiled features were invaluable both in terms of confirming the validity of our list, as well as emphasizing that certain characteristics had more impact than we had initially expected. For example, we initially placed groynes in the main channel as having low impact and high effort. Pepijn described that groynes, especially around the inlet structure, can have a large impact in terms of how the flow around the inlet behaves as well as how much sediment does or does not enter the channel. This was not emphasized as much in the literature we explored. Thus, we placed it instead as having an impact/effort of High/High. The adjustments made to the feature list, with commentary from experts Pepijn van Denderen and Hans van der Kwast, can be found in Appendix B.2.

4.2.3 System Architecture

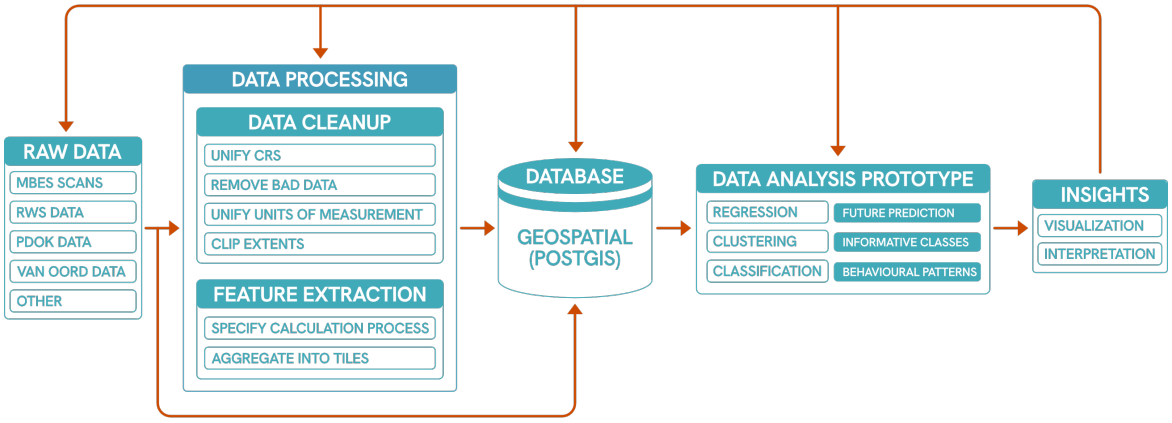


Figure 16. System architecture (own figure)

In order to facilitate the encoding of these identified features, a processing pipeline was derived (see Figure 16) that encapsulates the major necessary steps. Beginning with data ingestion, data processing and feature extraction, data storage, and a final prototype for analysis of the data via regression, classification, and/or clustering.

The data processing step involved the creation of an FME workbench that processes the raw data, performs the necessary cleanup, and then extracts the actual features that will be written to the database.

FME was chosen for this processing for a number of reasons. For one, it is widely used by the client, and they are very familiar with it. By providing the client with an FME workflow, something they understand well, it becomes much more likely that they will be able to extend and even optimize the workflow we provide them.

Secondly, it provides a very easy interface for creating precisely what we and the client desired. That is to say, an understandable, reusable, and extendable workflow that performs the processing with minimal manual intervention necessary. It can even be integrated into their company database systems easily using FME Flow. That way, once new data comes in, the client will be able to, with one click, run the pipeline for all of their channels with minimal manual intervention.

4.3 Feature Database

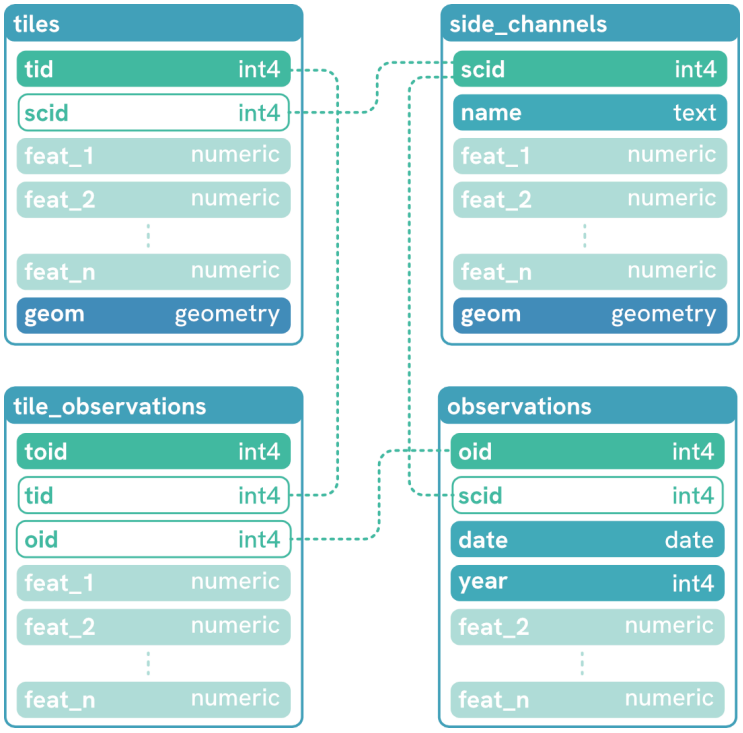


Figure 17. Database schema, primary keys in green, foreign keys outlined in green (own figure)

Once calculated, the feature values are stored in a PostgreSQL database with the PostGIS extension, hosted by Supabase. The schema is outlined in Figure 17. The database can be directly integrated with FME, QGIS, and Python, and since it is not hosted locally, can be accessed by every member of the team at any time. However, since we are using the free version, the size of our database is limited. This means it is impossible to store all datasets in the database. Only the geometry of the side channels and the tiles is stored, together with the calculated features. The raw input data, such as the MBES measurements, are stored on a shared drive.

4.3.1 Feature Scale

This project faces a data limitation, primarily in terms of the *temporal* resolution. It is quite difficult to derive a robust prediction of side channel behavior with only two measurements to draw from. However, *within* one year, we have usable measurements for 26 side channels (see Section 5.1.1 for details), which is actually quite a substantial amount of data. In order to try and bridge the gap and allow for more training data from our limited dataset, we split features into two scales: Channels and Tiles. Channel-level features encapsulate some phenomena that exist for the entire channel, for example, the length, how many inlets/outlets it has, etc. Tile-level features are local values within a 5x5m grid, for example, the bed level or flow direction. This allows for comparisons between not only all side channels, but also the local grid cells within the channels, as perhaps certain local phenomena are also comparable (e.g. the deep ditch at the inlet) and lead to similar morphological development. By combining features from these scales, we aim to extract as much predictive power as we can from the data we have access to to generate the most robust analysis.

4.3.2 Side Channels

The side channel table stores static channel-level features, as well as the geometry of the side channel, provided by RWS, in the form of a MultiPolygon. Since the boundaries of a side channel are hard to define, this was the abstraction chosen. While this representation doesn't completely reflect reality in all cases, it provides a consistent way to describe the shape of a side channel. These polygons represent the extent of the side channel we will be analyzing, rather than a fixed description of the shape of the side channel. The actual shape of a side channel depends on the water level, which changes over time and is further impacted by the processes of sedimentation and erosion.

4.3.3 Tiles

The tiles database contains the static tile-level features, as well as the tile geometry associated with the side channels. The tiles were created using QGIS and then loaded into the database using FME. In QGIS, a grid was created with 5 by 5 meter tiles, based on the RD New (ESPG:28992) coordinate system. A higher resolution could be possible, but requires more processing power, time, and database capacity than is available to us for this project. To further save on space and time, tiles were only generated for the channels that were determined to be "usable" (see Appendix C), as deriving tiles for the channels without sufficient data would likely place us over our database storage limits. RD new coordinates are already expressed in meters, so the coordinates of the corners of the tiles are always whole integers divisible by 5. This alignment was chosen because it lines up with other datasets, like the flow rate map and the bathymetry data. Only the tiles fully within the geometry of the side channels were loaded into the database to limit — but unfortunately not completely prevent — the inclusion of the transitional (interpolated) tiles between the bathymetric data of the side channel bed and the LIDAR data of the banks.

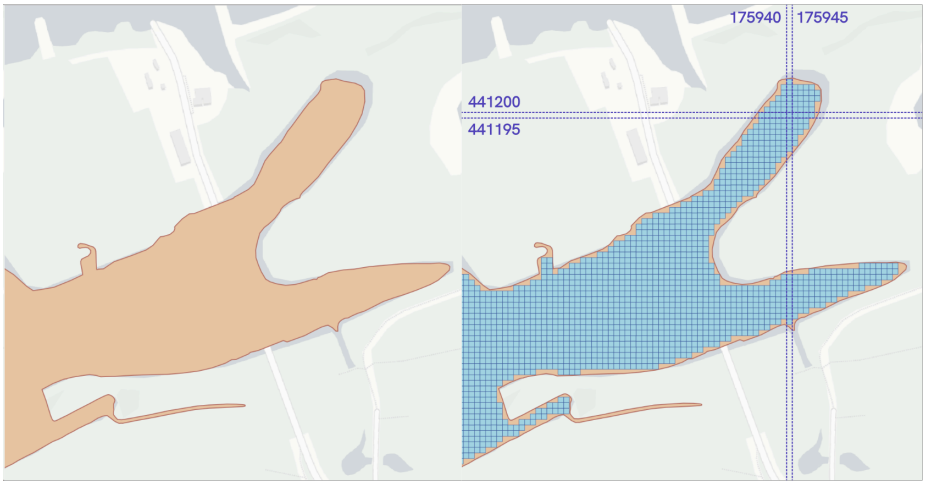


Figure 18. Left: Side Channel geometry (Schoutenwaard). Right: Tiles (own figure)

4.3.4 Static and Variable Features

Both the side channels and the tiles can have static and variable features. Static features are only written once to the database. They have either a very low update frequency or simply do not update at all. These features can be overwritten periodically if, after some years, the underlying data is updated, but if it becomes the case that this data is updated somewhat regularly (i.e. yearly), then it would be more appropriate to change these to dynamic features. These static features are stored together with the geometry in the main tables.

Variable features, on the other hand, can change every year and are therefore stored in a separate table, either `tile_observations` if the feature is on the tile-level, or `observations` if on the channel-level. These dynamic features should be updated on a yearly basis as new data comes in. For each subsequent year that you collect data, the performance of the machine learning approaches should improve in relation to the increase in training data.

A row in the `observations` table will have a year and a side channel ID, alongside all of the side channel dynamic feature values. A row in the `tile_observations` table will have a foreign key relationship to the associated row in the `observation` table (to ensure the years never get out of sync), as well as all of the tile-level dynamic feature values.

4.4 Feature Prioritization

Due to time constraints within the project, it was not feasible to implement features for all identified characteristics that could impact the sedimentation of side channels. Furthermore, due to the curse of dimensionality (Chen, 2009), it is in fact not desirable to attempt to perform analysis using such a high-dimensional feature vector, especially given how limited our data is.

Thus, from the full list of characteristics, a short list of features was identified for possible implementation during the project (see Table 3). These were features that we classified as either having a high or medium impact, while taking medium or low effort to implement in our system. The goal is to balance these factors to allow for the most robust analysis possible, given the limited number of samples.

Table 3
Short list of identified features to implement

characteristic	Feature	Type	Impact	Effort	References	Data Source
Bed level	Bed level	tile	High	Low	(Thus, 2025), (van Denderen et al., 2019), (Ottevanger & Chavarrias, 2019)	Van Oord Raster Average Layer
Length of the channel	Spine length	channel	High	Low	(Thus, 2025), (van Denderen et al., 2019), (Denderen et al., 2017)	Calculated MAT Spine
Slope	Average side channel slope	tile	High	Low	(Riquier et al., 2017), (Ottevanger & Chavarrias, 2019)	Derived from Van Oord Average Depth
Aspect	Average side channel aspect	tile	High	Low		Derived from Van Oord Average Depth
Flow direction	Side channel centerline direction	tile	High	Low		Derived from centerline
Channel connections	Pond / one-sided / two-sided	channel	High	Low	(Meijer & Winden, 2020; Riquier et al., 2017)	Van Oord + manual updates
Roughness	Standard deviation of height within cells	tile	High	Low	(van Denderen et al., 2019)	Van Oord Raster Standard Deviation Layer
Groyne(s) in main channel	(van Denderen et al., 2019)	channel	High	Med	Openstreetmap	User Input
Angle between main channel and side channel	Angle between main channel and side channel	channel	High	Med	(Thus, 2025), (Mosselman, 2001), (Riquier et al., 2017), (Denderen et al., 2017)	Derived from the MAT of the main and side channel
Bends in side channel	Straightness of channel spine	tile	High	Med	(Thus, 2025)	Derived from MAT Spine
How many days a year the channel is active	Co-flow days	channel	High	Med	(Meijer & Winden, 2020; Riquier et al., 2017)	Van Oord statistical calculation
Length relative to main channel	Side channel length as a percentage of main channel length between inlet and outlet	channel	High	Med	(Mosselman, 2001), (Riquier et al., 2017), (Denderen et al., 2017)	Ratio of MAT spines of main and side channels
Upstream discharge	Average upstream discharge	channel	High	Med	(Thus, 2025), (Meijer & Winden, 2020; van Denderen et al., 2019)	RWS Waterafvoer Data
Waves caused by ships	Average number of ships traveling by per month	channel	High	Med	(Thus, 2025), (van Denderen et al., 2019), (Mosselman, 2001), (Meijer & Winden, 2020)	AIS Data
Width	Average width perpendicular to the spine	tile	High	Med	(Thus, 2025), (van Denderen et al., 2019), (Riquier et al., 2017), (Denderen et al., 2017)	MAT Medial Ball diameter
Distance from inlet	Distance expressed as a percentage from the inlet	tile	Med	Low	(Meijer & Winden, 2020)	Downstream distance derived from MAT spine
Tidal	yes / no	channel	Med	Low		Van Oord + manual updates
Depth	Average depth	tile	Med	Med	(Mosselman, 2001), (Riquier et al., 2017), (Denderen et al., 2017)	Van Oord Raster Average Layer

Flood frequency	Number of floods per year	channel	Med	Med	(Mosselman, 2001), (Riquier et al., 2017)	Van Oord debietkaart
Side channel in an inner or outer bend of the main channel	Inner / Outer classification	channel	Med	Med	(Mosselman, 2001), (Denderen et al., 2017)	User Input
Vegetation near the channel	Relative vegetation density	tile	Med	Med	(Thus, 2025), (Mosselman, 2001)	Van Oord vegetation data

4.5 Implemented features

The short list served as a guide for the development of features. As we worked down the list, some features were removed and others added based on our experience, time limitations, and the eventual availability of data (discussed in more depth in Section 5). The final list of implemented features is as follows:

Table 4
Features that were implemented in the final pipeline

Feature	**Data Source(s)**	**Data Type**	**Static**	**Dynamic**	**Tile-level**	**Channel-level**
Bed Level	MBES Dataset	float4		✓	✓	
Channel Length	Polygons, Inlet and Outlets	float4	✓			✓
Slope	MBES Dataset	float4		✓	✓	
Aspect	MBES Dataset	float4		✓	✓	
Flow Direction	Polygons, Inlet and Outlets	float4	✓		✓	
Channel Connections	Inlets and Outlets, Manual with Satellite Reference	int2	✓			✓
Roughness	MBES Dataset	float4		✓	✓	
Inlet between Groynes	OSM Polygon Data, Manual with Satellite Reference	bool	✓			✓
Relative Length	Polygons, Inlet and Outlets	float4	✓			✓
Width	Polygons, Inlet and Outlets	float4	✓		✓	
Distance from Inlet	Polygons, Inlet and Outlets	float4	✓		✓	
Tidal	Manual from Van Oord metadata	bool	✓			✓
Inner/Outer Bend or Straight	Manual with Satellite Reference	text	✓			✓
Vegetation Near channel	RWS Vegetation Monitor	float4		✓		✓
Minimum flow Threshold	Flow Threshold Data	int8	✓		✓	
Bank	Manual with Satellite Reference	bool	✓			✓
Area	Polygons	float4	✓			✓
Perimeter	Polygons	float4	✓			✓
Distance to Bank	Polygons	float	✓		✓	
Total	**19**		**14**	**5**	**9**	**10**

The list of features balances well between tile and channel-level features for analysis, but is somewhat more imbalanced in terms of static versus dynamic features. This can be attributed largely to only having the MBES dataset as a dynamic source. Features based on that dataset made up the majority of dynamic features. There were other identified data sources, but they were not able to be used for this project. Namely, AIS data (Section 5.2.2) and water volume measurements (Section 5.2.4). Once these data sources become available and are integrated, they will balance out the static and dynamic features somewhat.

Extracting the features from the data can be done manually, but as this is time-consuming, it is important to automate this process to enable future scaling. The automated feature extraction was implemented in FME by Safe Software[3]. The resulting FME pipeline ensures feature extraction can be

[3] https://fme.safe.com/

processed efficiently and reliably for all the relevant side channels. Individual features are implemented as embedded transformers in a large FME workbench. The workbench is available to be viewed on the GitHub repository[4] of the project team.

Broadly speaking, the FME workbench is separated into three sections: input, feature extraction, and output. The input section reads raw data, as well as necessary data stored in the database, such as channel geometry, tiles, etc. The feature extraction section takes the data and calculates the actual features that will be written to the database. This step is further broken down into static and dynamic features, which are then split into tile and channel-level features. Once the features have been calculated, they are then written to the database in the output section.

The following subsections describe each feature along with the motivation for its inclusion. They also discuss the implementations of each of the features using FME.

4.5.1 Bed Level

The bed level is measured directly with a MBES. It represents the height of the channel's bed relative to the Normaal Amsterdams Peil (NAP) height. This is one of the more crucial features, as the difference between bed level values between one year and another represents the erosion and sedimentation at different locations in the side channel. Thus prediction of sedimentation is largely based on the bed levels.

Figure 19. Bed level for the Bakenhof channel (own figure)

This is a very straightforward feature to implement. First, the provided bathymetric data is clipped using a polygon representing the water area (provided by the client). Figure 19 shows the resulting raster. The pixel values are converted into points and associated with their corresponding tile. The average of each pixel is derived per tile, and that value is written as the bed level.

4.5.2 Channel Length

This feature encodes the length, in meters, of a given side channel. If channels are very short or long, it could impact the speed of the water, how likely it is that sediment manages to slow down enough to deposit, etc.

The length can be quite ambiguous, given the meandering and splitting nature of certain channels. Thus, it was decided that the length would be the shortest distance along the centerline (generated with the *CenterLineReplacer*-transformer) between the inlet and the outlet. For channels that are missing one or both of these values (one-sided channels or ponds), an artificial point on the opposite side was used as a reference (see Figure 20 and Figure 21).

[4] https://github.com/lucjonker/GEOIT1501_Synthesis_Project_VanOord/tree/main

Figure 20. Extra placed point for a one-sided connected side channel (own figure)

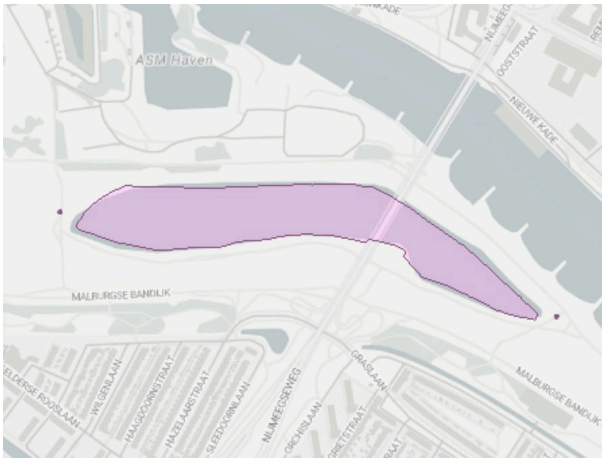


Figure 21. Extra placed points for an unconnected side channel (own figure)

The nearest points on the generated centerline to the inlet and outlet locations are identified using the *NeighborFinder*-transformer (see Figure 22). The centerline point closest to the inlet (p_{inlet}) is assigned the attribute _path_role = From, while the centerline point closest to the outlet (p_{outlet}) is labeled _path_role = To.

Figure 22. Closest point on centerline from p_{inlet} (own figure)

These p_{inlet} and p_{outlet}, together with the deaggregated centerline segments (each consisting of only two coordinates), form the input for the *ShortestPathFinder*-transformer. The resulting path length represents the channel length value for the side channel (see Figure 23).

If no valid path is found, typically when the polygon is not fully connected between the p_{inlet} and p_{outlet}, the straight-line distance between p_{from} and p_{to} is used instead (see Figure 24).

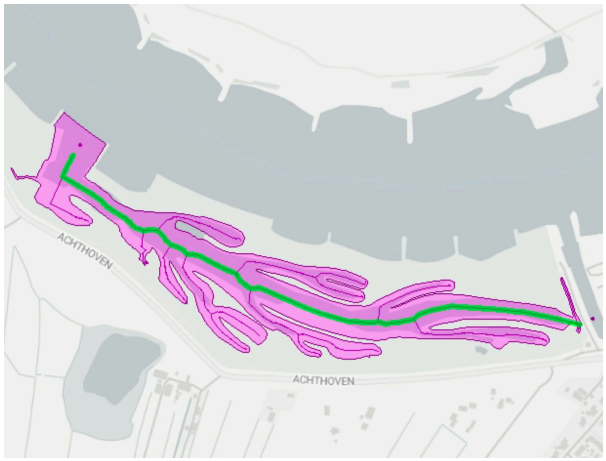


Figure 23. Length of the green line is used for the channel length (own figure)

Figure 24. Channel length, when no shortest path is found (own figure)

4.5.3 Slope

The slope of the channel's bed represents the steepness of the terrain at a given point. The motivation was that steeper segments of the channel bed are expected to experience more severe sedimentation/ erosion.

Figure 25. Slope for the Bakenhof channel (own figure)

Slope is implemented in precisely the same manner as bed level (Section 4.5.1), with the difference being that the raster slope is calculated on the bathymetric raster as the first step, and the resulting raster shown in Figure 25 is passed through the pipeline. For our calculations, we used Equation 1 of Zevenbergen and Thorne (Zevenbergen & Thorne, 1987), as it is reported to perform better on smoother surfaces, and the bottom of a river channel is generally made up of smooth hill-like structures.

$$\begin{aligned} \text{slope} &= \arctan\left(\sqrt{p^2 + q^2}\right) \\ p &= \frac{h_r - h_l}{2\Delta x} \\ q &= \frac{h_u - h_b}{2\Delta y} \end{aligned} \tag{1}$$

Where h_r, h_l, h_u, and h_b represent the right, left, upper, and bottom pixels relative to the center, respectively.

4.5.4 Aspect

The aspect of the channel's bed represents the direction that the slope is facing, represented in degrees measured from North. The motivation being that, in combination with flow direction, the model may get a sense of how the water flowing would interact with the terrain (i.e. would it flow into it or with it) to get an indication of whether it would experience much sedimentation or erosion.

Figure 26. Aspect for the Bakenhof channel (own figure)

Aspect is derived in much the same manner as slope (described in Section 4.5.3); however, instead of being calculated on the original 0.5x0.5m grid, it is first resampled to a 5x5m grid before aspect is calculated (shown in Figure 26). This is because the aspect values calculated on such a granular scale became nonsensical once averaged over the 5x5m grid cells in the same way that slope and bed level are calculated. Again, we chose the formula of Zevenbergen and Thorne, and thus Equation 2 uses the same calculations for p and q as outlined in Equation 1, with aspect being calculated as follows:

$$\text{aspect} = \arctan\left(\frac{p}{q}\right) \tag{2}$$

An important caveat is that the resampling methods provided by FME are limited to, at most, considering the 16 closest cells to the newly sampled pixel. This means for a resampling from 0.5x0.5m to 5x5m, instead of all 100 cells being considered, only the center 16 are. To mitigate this effect, first a resampling is done from 0.5 to 2.5m grid using the closest 16 points, before doing a second resampling to a 5m grid using the closest 4 cells. This will use 64 out of the 100 pixels, but was considered an adequate workaround by the client.

4.5.5 Channel Connections

This feature represents the "type" of a channel as discussed in Section 2.4, in terms of the inlets and outlets. Two-sided channels that co-flow consistently throughout the year are expected to be more dynamic and experience higher levels of erosion and sedimentation than one-sided channels, and much more than ponds.

Ponds have no inlet or outlet, so are given a value of `0`. One-sided channels generally have only an outlet and are given a value of `1`. Two-sided channels have at least one inlet and at least one outlet, and are given a value of `2`. This feature is added manually when a new side channel is added.

The value was decided by a combination of visual analysis of satellite imagery, as well as considering the Inlet and Outlet point dataset provided by Van Oord.

4.5.6 Roughness

Roughness was included with the motivation being that a generally more 'rough' portion of the river bed will exert more resistance on the water flowing above it. This will slow the water down more and lead to sediment falling more quickly.

Figure 27. Roughness for the Bakenhof channel (own figure)

Roughness is implemented in the same way as it is implemented in CGAL and QGIS (QGIS, n.d.). That is, roughness "is the largest inter-cell difference of a central pixel and its surrounding cell" (Dolan et al., 2007). Starting from the 0.5m bathymetric raster, roughness is calculated by using a 9x9 kernel to find the maximum and minimum depth around a given pixel, with roughness being defined as the difference between the max and min.

$$\begin{aligned} B_{\text{max}} &= \text{maximum Z in 9x9 window} \\ B_{\text{min}} &= \text{minimum Z in 9x9 window} \\ R &= B_{\text{max}} - B_{\text{min}} \end{aligned} \tag{3}$$

Once roughness has been derived, the raster is clipped (shown in Figure 27). The pixel values are converted into points and associated with their corresponding tile. The average of each pixel is derived per tile, and that value is written as the roughness.

4.5.7 Inlet between Groynes

This feature represents whether or not the inlet of a side channel lies between two groynes built into the river. These groynes can have a large impact on the flow behavior right around the inlet structure, which therefore has a large impact on the sediment that does or does not enter the channel.

To calculate this feature first, OpenStreetMap (OSM) data was queried using the QuickOSM plugin for QGIS for the `man_made=groyne` tag. This resulted in a set of groyne polygons as shown in Figure 28. Each channel was evaluated by hand to determine if the inlet was between two groynes. Channels were assigned a `TRUE` if there was an inlet recorded and it was located between two groynes, otherwise, they were assigned a `FALSE`. This feature is added manually when a new side channel is added.

Figure 28. Groynes (blue) near Bakenhof side channel (red) (own figure)

4.5.8 Relative Channel Length

Some side channels are very straight (Figure 29), and some tend to meander or bend a bit (Figure 30) before reconnecting with the main channel. A more straight channel is expected to have different erosion patterns, energy loss, etc. than a channel that bends a lot, thus this feature expresses that behavior with a ratio between the actual length of the channel versus the straight distance between the inlet and the outlet.

To calculate the relative length of the channel (l_{relative}), the Cartesian distance between the inlet and outlet points was divided by the channel length (computed the same way as the feature *channel length*).

$$l_{\text{relative}} = \frac{l_{\text{channel}}}{l_{\text{inlet}\rightarrow\text{outlet}}} \tag{4}$$

This means the relative length is expressed as a ratio.

Figure 29. Relative length of 1.05 for side channel *Palmerswaard* (own figure)

Figure 30. Relative length of 1.28 for side channel *Klompenwaard* (own figure)

4.5.9 Width

The width is the cross-sectional width of a side channel, perpendicular to its center line, in meters. Wider sections of a channel area are expected to be calmer as the water has more space over which to disperse its energy, while Thin pieces may experience much stronger water pushing through, which could impact how quickly those sections erode.

To calculate width, a centerline is first calculated for each channel using its polygon shape, with the *CenterLineReplacer*-transformer. This centerline is then smoothed and simplified using the *Generalizer*-transformer (see Figure 31).

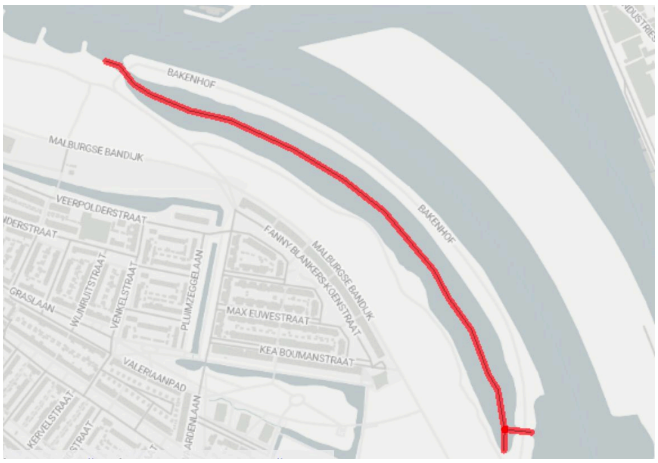


Figure 31. Generalized centerline of Bakenhof side channel (own figure)

The generalized centerline is subsequently clipped by the tile boundaries, splitting it into multiple line segments that correspond to individual tiles (see Figure 32).

Figure 32. Centerline chopped into segments corresponding to the tiles (own figure)

For each line segment, the horizontal angle ($\Delta\measuredangle_{\text{centerline},i}$) is calculated using the *HorizontalAngleCalculator*-transformer. A vertex ($p_{\text{mid},i}$) is then inserted at the midpoint of each segment, storing both a unique ID and the computed $\Delta\measuredangle_{\text{centerline},i}$.

Next, a line segment is generated at each $p_{\text{mid},i}$, oriented perpendicular according to its $\Delta\measuredangle_{\text{centerline},i}$ and sharing the same unique ID. This is achieved by, for each $p_{\text{mid},i}$, placing a point perpendicular to its horizontal angle:

$$x_{\perp,i} = x_{\text{mid},i} + 5 * \cos\left(\Delta\measuredangle_{\text{centerline},i}\right) * \frac{\pi}{180} \tag{5}$$

$$y_{\perp,i} = y_{\text{mid},i} - 5 * \sin\left(\Delta\measuredangle_{\text{centerline},i}\right) * \frac{\pi}{180} \tag{6}$$

$$p_{\perp,\,\text{i}} = \left(x_{\perp,i}, \quad y_{\perp,i}\right) \tag{7}$$

Each perpendicular line ($\Delta\measuredangle_{\perp,i}$) is then extended by the length of the channel's bounding box in both directions (see Figure 33).

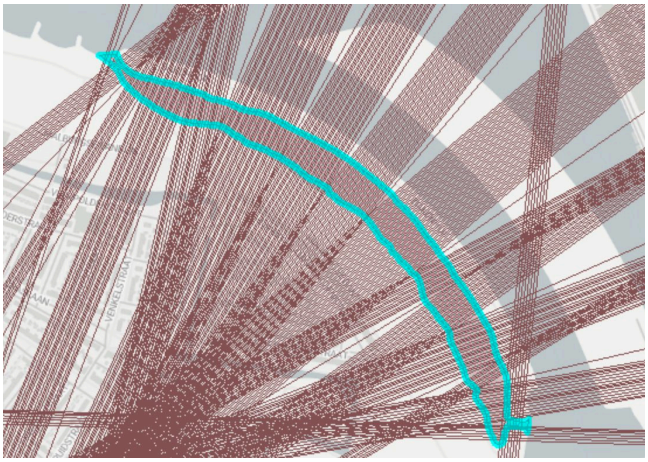


Figure 33. Extended perpendicular lines to centerline (own figure)

These extended lines are clipped by the side-channel polygon (see Figure 34 and Figure 35), and the resulting segment lengths are computed using the *LengthCalculator*-transformer. The measured length is stored as the width attribute for the corresponding $p_{\text{mid},i}$.

Figure 34. Perpendicular lines to centerline clipped to Bakenhof side channel polygon (own figure)

Figure 35. Zoomed in perpendicular lines to centerline clipped to Bakenhof side channel polygon (own figure)

In some cases, an extended line may intersect the polygon multiple times (see Figure 36). To handle this, a *Deaggregator*-transformer is used to split the line into separate parts (see Figure 37). The correct segment is then identified by testing whether its corresponding $p_{\text{mid},i}$ lies within the segment's bounding box. The length of that segment is assigned as the width value for the associated $p_{\text{mid},i}$.

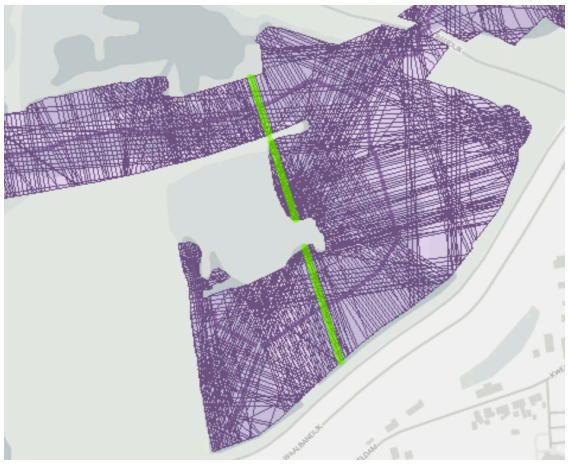


Figure 36. Clipped perpendicular lines intersect polygon multiple times (own figure)

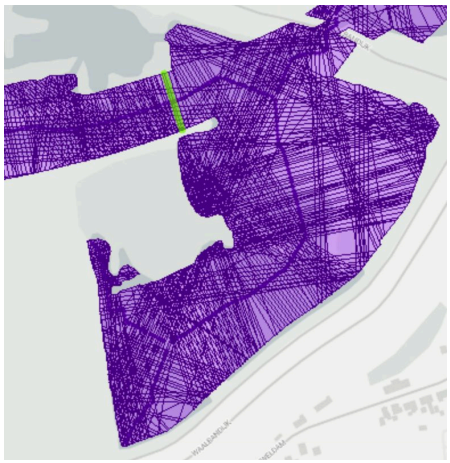


Figure 37. Deaggregated lines (own figure)

Finally, the *PointOnAreaOverlayer*-transformer transfers each $p_{\text{mid},i}$'s width value to the tile it falls within. Tiles without any points receive a width value through nearest-neighbor interpolation, performed by the *NeighborFinder*-transformer, based on Cartesian distance. Figure 38 shows the computed width values for all the tiles in the Afferden en Deest channel.

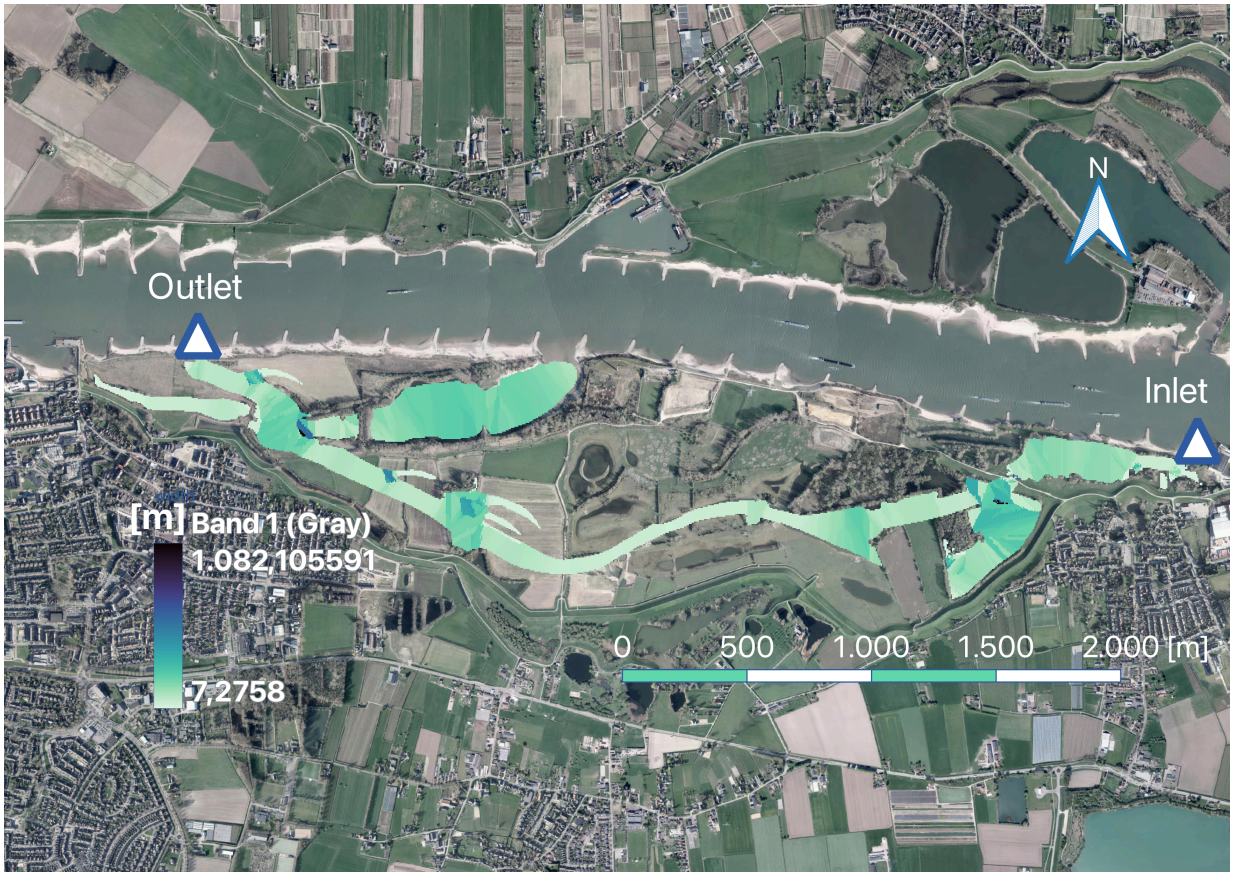


Figure 38. Width for side channel Afferden en Deest (own figure)

4.5.10 Flow Direction

Flow direction encodes the direction, in degrees, that the water is flowing in a given tile. There are certain phenomena within side channels, like the inlet hole or obstacles, that have some impact on where sedimentation occurs. We tend to see sedimentation downstream of these objects, so the hope was to encode some of that information using the flow direction.

The flow direction is computed for side channels that have both an inlet and an outlet defined in the VoxView (Van Oord's internal data viewer) of WOCU Rijntakken and is determined based on the orientation of the channel's centerline.

First, the inlet and outlet points belonging to side channels that have a corresponding polygon shapefile are manually selected and filtered from the VoxViewer object points. These filtered inlet and outlet points are then linked to their corresponding side channel polygons using the *NeighborFinder*-transformer, and then merged per side channel.

The horizontal angle between the inlet and outlet point ($\measuredangle_{\text{inlet}\to\text{outlet}}$, see Figure 39) is computed as:

$$\measuredangle_{\text{inlet}\to\text{outlet}} = 360 + \arctan(x_{\text{outlet}} - x_{\text{inlet}}, \quad y_{\text{outlet}} - y_{\text{inlet}}) * \frac{180}{\pi} \tag{8}$$

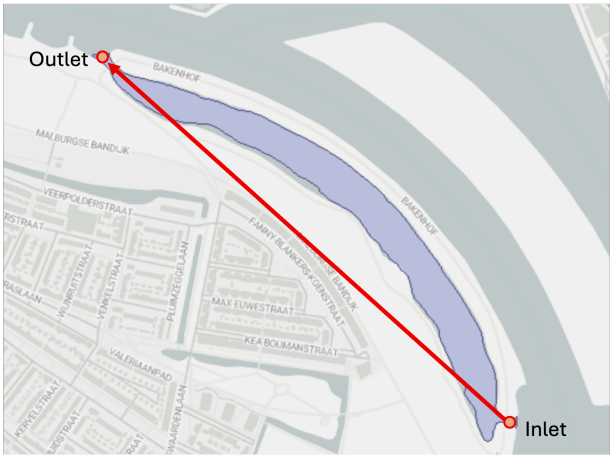


Figure 39. Horizontal angle inlet to outlet (own figure)

Next, the centerline of each side channel is generated with the *CenterLineReplacer*-transformer and clipped by the tile boundaries (see Figure 32). For every clipped centerline segment, the horizontal angle ($\Delta\measuredangle_{\text{centerline},i}$) is computed using the *HorizontalAngleCalculator*-transformer.

Since $\Delta\measuredangle_{\text{centerline},i}$ can be oriented in two directions, each computed angle is compared with $\measuredangle_{\text{inlet}\to\text{outlet}}$ to ensure a consistent flow orientation. The difference between the two angles is calculated as:

$$\text{angle difference} = \left| \measuredangle_{\text{inlet}\to\text{outlet}} - \Delta\measuredangle_{\text{centerline},i} \right| \tag{9}$$

If this difference exceeds 180°, it is adjusted as follows:

```
If angle difference > 180°:
    angle difference = 360 - angle difference
```

When the resulting difference is greater than 90°, the direction of the centerline segment is flipped by 180°:

If angle difference > 90:
    $\Delta\measuredangle_{\text{centerline},i} = \Delta\measuredangle_{\text{centerline},i} + 180$

Finally, the *LineOnAreaOverlayer*-transformer transfers each line's $\Delta\measuredangle_{\text{centerline},i}$ value to the tile it falls within. Tiles without a $\Delta\measuredangle_{\text{centerline},i}$ value receive a $\Delta\measuredangle_{\text{centerline},i}$ value through nearest-neighbor interpolation, performed by the *NeighborFinder*-transformer, based on Cartesian distance. Figure 40 shows the flow direction for all the tiles in the Afferden en Deest channel.

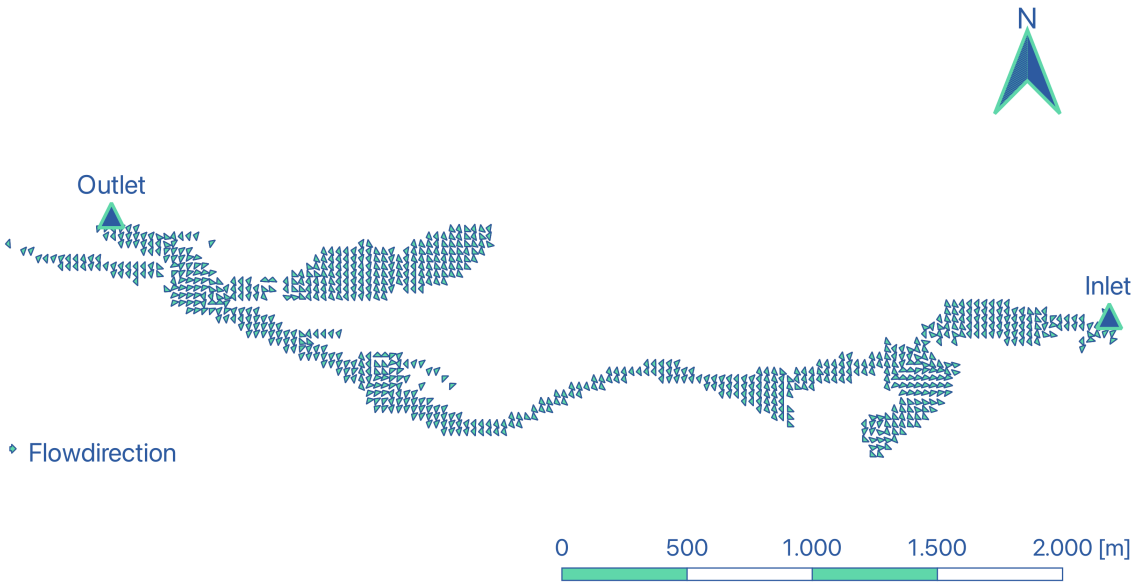


Figure 40. Flow direction for side channel Afferden en Deest (own figure)

4.5.11 Tidal

Many channels in the Netherlands, and thus also in our dataset, lie behind dikes, locks, etc. in the river that separates them from the ocean. However, some do not, and thus they are affected by tidal forces. This has a large impact on water flow behavior, speed, and potentially even direction.

The value is assigned by consulting the metadata Van Oord provided with the side channel data. The channel is assigned `TRUE` for this feature if they are tidal, otherwise, they are assigned `FALSE`. This feature is added manually when a new side channel is added.

4.5.12 Inner/Outer Bend or Straight

Side channels often branch off from the main channel in a bend of the river. As the river bends, the water will want to continue flowing in the same direction, leading to complex water behavior in different parts of the bend. Thus, whether the side channel is branching on the *inside* or *outside* of this bend can affect how the water will interact with the inlet structure, thus this feature exists to try and represent some of that behavior.

To determine whether the channel exists on an inner bend, an outer bend, or a straight segment, satellite imagery is assessed. Depending on the orientation, the channel is assigned the value of `Inner`, `Outer`, or `Straight`. This feature is added manually when a new side channel is added.

4.5.13 Vegetation near Channel

Vegetation can have quite a strong impact on how banks erode (Ahmed & Mohanadhas, 2025). Naturally, if a section of bank is supported by plants with dense root systems, it can assist in keeping it more structurally sound. The plants themselves can also help to absorb some of the energy of water passing over them. This feature thus expresses a percentage of the bank regions of the side channel that have any vegetation more dense than just sparse grass.

Figure 41 shows the classification of vegetation around the Bakenhof channel as retrieved from RWS “vegetatiemonitor” described in Section 5.2.1.

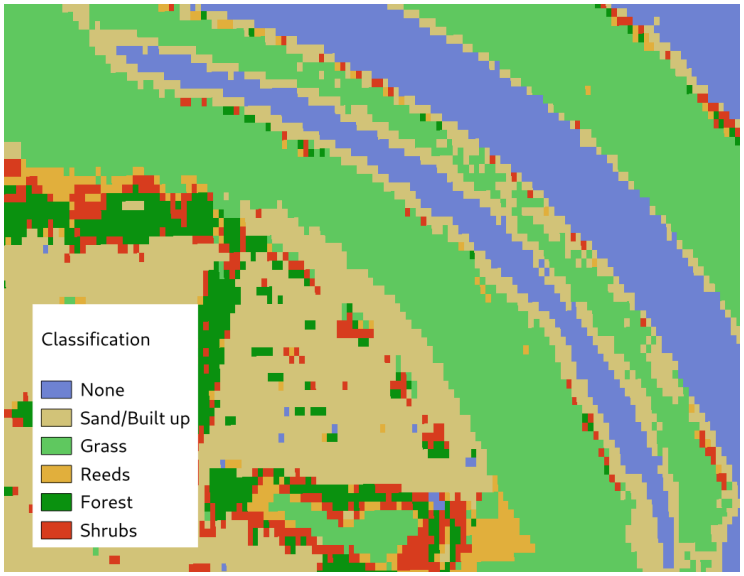


Figure 41. Different vegetation classifications, surrounding the Bakenhof channel (own figure)

The feature itself is represented as a percentage of vegetation around the side channel that is of categories “Reeds”, “Forest”, or “Shrubs”. The “Grass” category was excluded as it appears that for it to attain this classification, it needs to be only a very sparse grass, which we do not expect to have a strong effect on erosion. The percentage for the whole channel was decided because it is somewhat ambiguous how to best assign these classification values – which are on the outside of the channel – to the tiles that are on the *inside*. One could take the closest value, but for tiles close to the center of the channel, this value becomes meaningless.

To assign the values, a buffer is made of the side channel polygons to get the areas around the banks of the channels. This is used to clip the raster with the vegetation classifications. The remaining pixels are then turned into points with the *RasterCellCoercer*-transformer before being associated with the correct side channel with a *SpatialFilter*-transformer. From then on, it is a simple case of calculating the total number of classification points for the channel and taking the percentage of those with the correct classification. Once the percentage has been derived, it is written to the database.

4.5.14 Minimum flow Threshold

Not all side channels are the same in terms of when they will experience co-flow. Some channels have inlets that are built a bit higher, and some channels are simply positioned higher up overall. This will lead to those channels flowing less often than ones that are lower down, as more volume will need to flow in the river before those sections go underwater. This feature attaches such a flow threshold to each tile, representing how much volume of water needs to be flowing in the main channel before the tile will go underwater.

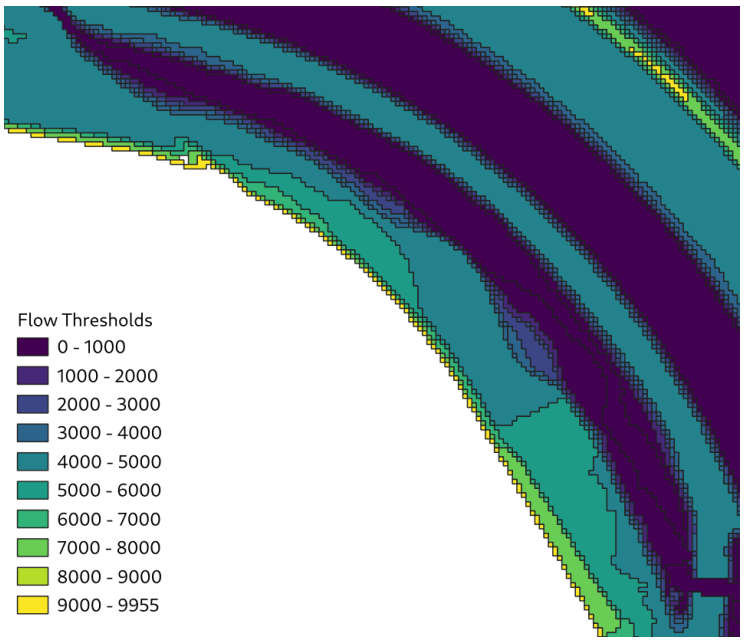


Figure 42. Different flow thresholds near the Bakenhof side channel represented in a shape file (own figure)

The implementation of this feature did involve a bit of manual effort to prepare the data. The individual layers were downloaded as shape files from Van Oord's Vox Viewer tool. In QGIS, these layers were merged into one shape file (shown in Figure 42). Then, for each polygon, an attribute was added representing the minimum flow thresholds for that category to be used in FME (previously, the shape files were only differentiated by name).

With this data processed, the FME script to apply the value to the tiles becomes straightforward. The threshold shape file is first clipped using the polygons of the side channels. Then, the center-points of the tiles are superimposed on the resulting polygons using a *PointOnAreaOverlayer* transformer. With the tiles now containing the associated minimum flow threshold value, the result can be written to the database.

4.5.15 Bank

Banks are a special type of side channel, called an 'Oevergeul' in Dutch. Instead of being an offshoot that splits from the main channel, a bank channel is built into the river itself with a man-made dam separating the two. These channels are expected to behave much differently than a traditional channel, and thus, this feature captures that information.

The value is assigned by assessing satellite imagery. The channel is assigned `TRUE` for this feature if they are bank channels, otherwise, they are assigned `FALSE`. This feature is added manually when a new side channel is added.

4.5.16 Area

This feature represents the area in square meters covered by the side channel. A larger area would likely correspond to more water volume being moved through the channel, thus, in the absence of properly usable volume data from RWS, we use this as a sort of proxy until enough data has been collected that volume can be input as a dynamic feature.

The implementation of this feature is quite trivial, as there is a built-in transformer (the *AreaCalculator*-transformer) to derive the area of a polygon. This value is calculated and then written to the database.

4.5.17 Perimeter

This feature represents the length, in meters, of the perimeter of the side channel polygon. Broadly, it is expected that the banks of the side channel lie on the borders of the polygons, thus this gives a sort of proxy for the length of banks.

The implementation of this feature is quite trivial, as there is a built-in transformer (the *PerimeterCalculator*-transformer) to derive the perimeter of a polygon. This value is calculated and then written to the database.

4.5.18 Distance to Bank

Because the banks of side channels can erode over time, depositing sediment into the side channel, this feature captures for each tile its distance from the bank. The intuition is that tiles closer to the bank may experience certain sedimentation patterns that would otherwise be difficult to model.

To calculate this feature, a point is first placed at the center of each tile using the *CentrePointPlacer*-transformer (see Figure 43).

Figure 43. Points placed in the middle of each tile (own figure)

From each of these points, the shortest distance to the corresponding side channel polygon is then computed using the *NeighborFinder*-transformer. This calculated distance represents the 'distance to bank' value for each tile as shown in Figure 44 for the Afferden en Deest channel.

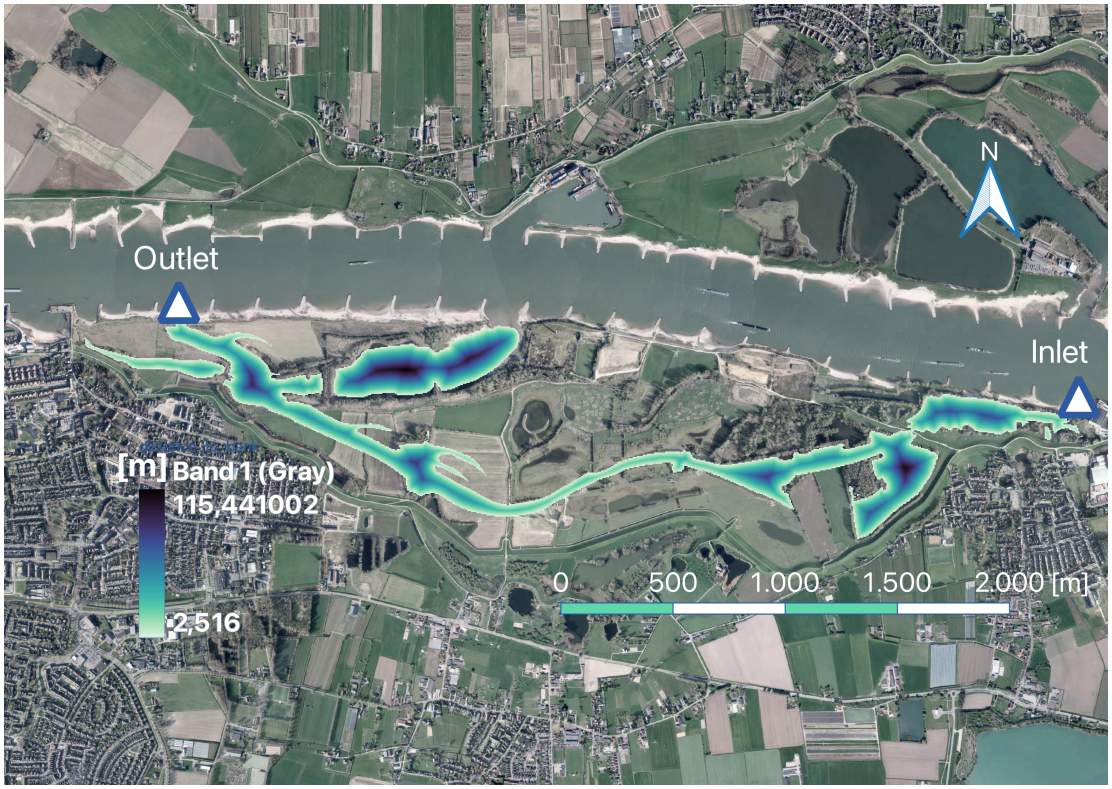


Figure 44. Distance to bank for side channel Afferden en Deest (own figure)

4.5.19 Distance from Inlet

The inlet structure is one of the more vital pieces of side channel anatomy, it dictates much of how the side channel behaves. For example, it is very common to see a deep hole right after the inlet where water pools in and digs a ditch, whose sediment is then spewed and deposited after the hole (this can sometimes even form a small sand bar). Thus, a feature describing the distance of a tile from the inlet can give an indication whether it might be in the path of these behaviors.

Distance from the inlet is computed as the shortest-path length along the centerline from the inlet point to points placed on that centerline.

First, a centerline is derived from each side-channel polygon with a *CenterlineReplacer*-transformer, then smoothed with a *Generalizer*-transformer. For every tile the generalized centerline intersects, a midpoint p_{mid} is created and tagged _path_role = To (see Figure 45). The nearest point on the centerline to the inlet point, p_{inlet}, is tagged _path_role = From. From→to-paths are then created from each p_{mid} to p_{inlet} (see Figure 46).

The centerline is then split into two-vertex line segments using a *Deaggregator*-transformer followed by a *Chopper*-transformer. These chopped segments, together with the from→to-paths, are fed into a *ShortestPathFinder*-transformer to obtain the distances from the inlet to each p_{mid}.

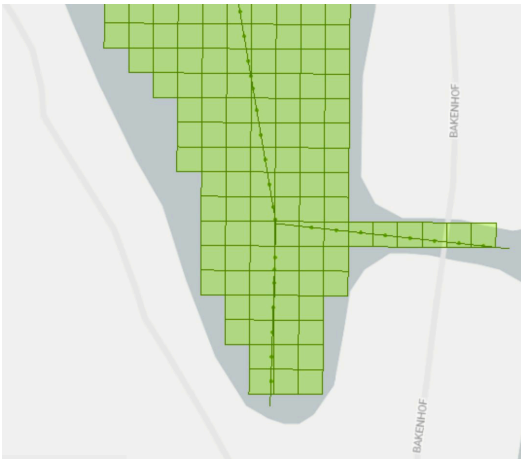


Figure 45. Midpoints p_{mid} placed on centerline for each tile it crosses (own figure)

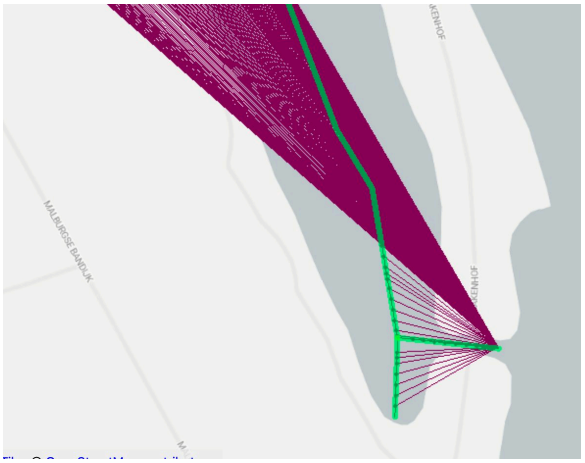


Figure 46. From→to-paths for *ShortestPathFinder*-transformer (own figure)

Finally, a *PointOnAreaOverlayer*-transformer transfers each distance value to the tile that contains its point. Tiles lacking a value receive a value using a *NeighborFinder*-transformer, based on Cartesian distance. Figure 47 shows the distance from inlet for all tiles in the Afferden en Deest channel.

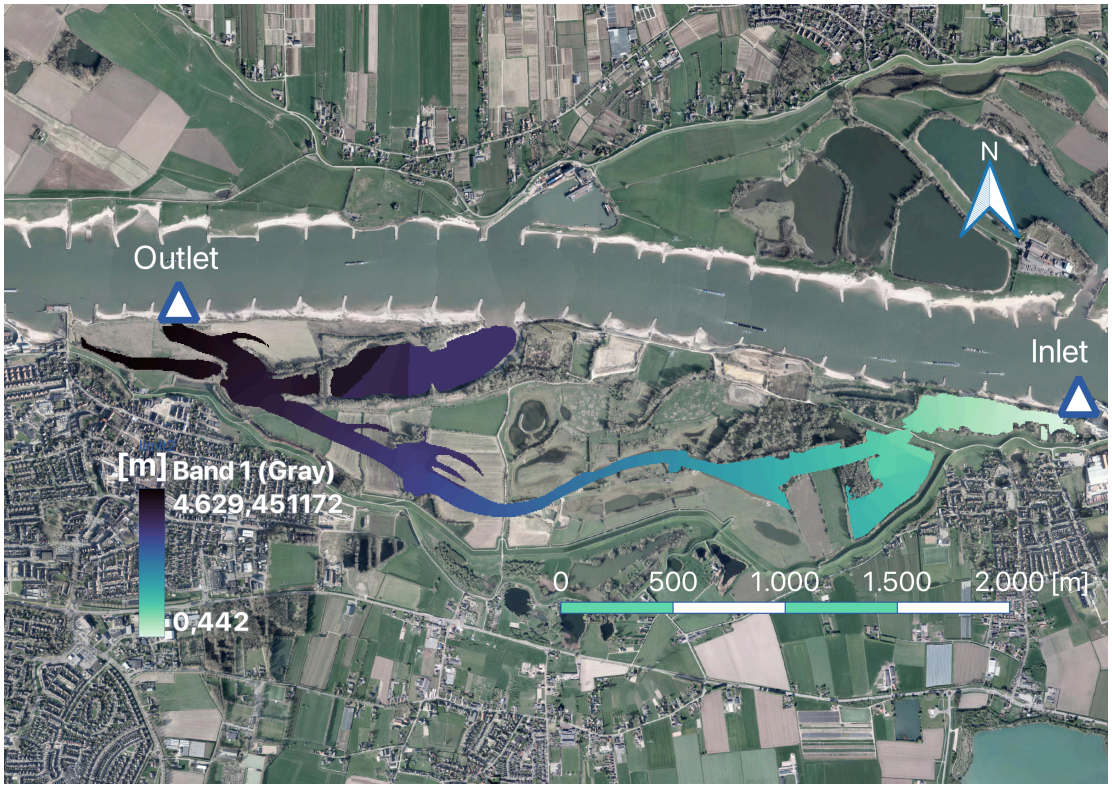


Figure 47. Distance from inlet for side channel Afferden en Deest (own figure)

4.6 Prototype

4.6.1 Statistical Analysis

Before diving into model training, it is important to first verify that the features, as we have defined them, contain useful information and are suitable for such a model. There are two main metrics that are considered. First, we calculate the correlation between each feature and the target variable to evaluate its predictive power in the form of a merit score.

The absolute Pearson correlation, or merit score, gives an indication of how much predictive power a feature holds. A higher merit score means there is a stronger correlation between the feature and the change. This means that if the merit value is greater than zero, the feature is beneficial to predicting sedimentation.

Second, we calculate the correlation and multicorrelation between features in the form of Pearson correlation, Variance Inflation Factor (VIF), and Tolerance. These metrics are then used to evaluate their benefit to the model, considering the information encoded in the full feature vector.

The Variance Inflation Factor (VIF) is a measure that is used to express how much the variance of a feature k is inflated due to multicollinearity (Kutner et al., 2005). This is achieved by running a standardized regression model where the feature to be investigated is treated as the dependent variable, and all the other features are used as predictors. This regression yields an R^2 value, which represents what part of the features' variance can be explained by the other features in the model. The VIF is defined as shown in Equation 10.

$$(\mathrm{VIF})_k = \left(1 - R_k^2\right)^{-1} \tag{10}$$

The inverse VIF or tolerance, as seen in Equation 11, is also commonly used to express the same property in a different way.

$$\mathrm{Tolerance} = \frac{1}{\mathrm{VIF}_k} \tag{11}$$

A common rule of thumb is that if the VIF exceeds 10, a feature has a high chance of causing redundancy and instability when training a machine learning model (Kutner et al., 2005).

4.6.2 Encoding Cyclical Features

Machine learning models fundamentally learn patterns by calculating distances between similar and dissimilar values. During the training process, a loss is calculated at every step based on the distance between the predicted and target variables. However, cyclical features such as the aspect angle (e.g., in degrees from 0° to 360°) pose a problem because values that are close in reality end up being far apart when represented in feature space.

To resolve this issue, a common method called trigonometric encoding (Adams & Vamplew, 1998) was used. Cyclical features are encoded as a combination of two new features, `feature_x` and `feature_y`, defined as the sine and cosine values, respectively, of the original angle. This transformation maps the cyclical feature onto a unit circle, solving the distance problem. As demonstrated in the example data for two tiles in Table 5, this encoding correctly brings formerly problematic values close together. This method allows machine learning models to properly calculate the distances and thus learn the true relationship between the values.

Table 5
Values for aspect features of two example tiles

tid	aspect_angle	aspect_x	aspect_y
35294	358.0610	0.9994	–0.0338
35259	0.0292	1.0000	0.0005

4.6.3 Bed Level Normalization

The bed level of the side channels is measured compared to the Amsterdam Ordnance Datum or Normaal Amsterdams Peil (NAP). Since side channels are spread all over the country, which means they have different base elevations. This means that the bed level needs to be normalized before they can be compared to each other. To do this, the average water height during 2024 was collected from RWS for several measuring stations located along the Rhine distributaries. To achieve the average water level for a side channel, the nearest station to upstream and downstream on the same river where located, and then their water heights were interpolated using inverse distance weighing.

4.6.4 Feature Reduction

As mentioned in Section 4.4, it is not recommended to run the model using all of the features as input, due to the curse of dimensionality. This curse states – in simplified terms – that when the amount of input features increases, a point is reached where the predictive power of the model decreases. Therefore, it is vital to manage the dimensionality of your input vector to prevent degradation of performance. This can either be solved by increasing the amount of training data or by reducing the number of input features. Since our data is limited to what we collected, a dimension reduction of the feature space is necessary. There are multiple ways to perform a feature reduction, but we will explore forward insertion and principal component analysis (PCA).

Forward insertion runs a simplified model and adds features one by one. The feature that reduces the loss the most gets added to the list of final features. In the next step, we run the simplified model again with the features you already added to the final model, together with one new feature, trying to find the feature that minimizes the loss the most. You repeat this process until you have the preestablished amount of features or the loss stops getting smaller.

A second option is performing a principal component analysis (PCA). A PCA linearly transforms the feature space into a new coordinate system along the axes that explain the most variance. Features are then remapped into new uncorrelated components. By only using the principal components — the ones that explain the most variance — the amount of input features can be reduced while keeping valuable information from all original features.

While this method might result in values that do not completely align with the actual water height at the location of the side channel, it provides a consistent benchmark that can be used across different years. This is in contrast with other approaches, like comparing the normalized bed levels for every individual channel, which might vary from year to year.

4.6.5 Predictive Model

After feature collection, it is essential to verify that the features possess enough predictive power suitable for machine learning applications. To this end, we have developed a neural network prototype that demonstrates the capabilities of our approach. However, this prototype serves as a proof of concept rather than a final model, as constructing a robust machine learning framework requires additional time and experimentation.

To evaluate the predictive capability of the features, a simple neural network in the form of a Multilayer Perceptron (MLP) was implemented using the Keras library[5] in Python. The loss is defined as the mean square error between the predicted and the real change. The dataset is split into a training and validation data subset, with 80% of the data used to train the model and 20% to validate.

An MLP has many hyperparameters that can be fine-tuned to optimize performance (Jaiswal, 2025). Determining optimal values for these hyperparameters is a lengthy process of trial and error, comparing different model setups and then making small incremental adjustments. While this is an important process, due to time constraints, only a small number of tests were performed to determine a viable model. This prototype still has room for improvement, and therefore should be viewed as a proof of concept rather than a final product.

An example of a parameter we can tune for an MLP is the depth and width. Depth refers to the number of hidden layers; a deeper model can better capture sequential patterns, but requires more data and training to achieve a decent result. A wider model can capture more complex relationships between the input variables, but increases the processing time exponentially and is prone to overfitting. To balance both of these aspects, our model consists of three hidden layers, the first two have 264 neurons, while the last one has only 64, summarized as (264, 264, 64).

In deeper networks, the weights and biases can become very small as they propagate backward through the layers, known as the vanishing gradient problem. This is especially the case with sigmoid and $\tanh$ layer activation functions, whose derivatives are very small for inputs outside of the range -1 to 1. To mitigate this, ReLu (Rectified Linear Unit) activation was chosen, which is linear for positive values and zero for negative values. The linear part always has the same derivative, which solves the vanishing gradient problem and allows deeper networks to learn effectively.

Before training, all input features are normalized to make sure they contribute equally during the training process. There are different normalization techniques, like MinMax scaling and Robust scaling, but for our data, Standard scaling performed the best as it helps to reduce the effect of outliers. Standard scaling converts the value of a feature to how many standard deviations it differs from the mean value. The mean of the standardized feature is therefore 0, and the standard deviation is 1.

Another technique applied to improve the generalization of the model is dropout. Dropout randomly deactivates a subset of neuron connections during one epoch. This is done so the model does not become overly reliant on a few pathways to obtain its result. This leads to a more robust model that is less prone to overfitting, but it does add more eccentric behavior during runs.

Lastly, the learning rate needs to be determined, as it plays a critical role in the convergence of the model. A high learning rate approaches the minima quicker, but risks overshooting and missing optimal solutions. A smaller learning rate, on the other hand, improves stability but can lead to slow convergence or getting stuck in suboptimal local minima. Because the learning rate is so important, this is determined separately for every side channel, based on the size of the training dataset and the number of variables.

[5] https://keras.io/

5 Data

5.1 Van Oord

Our client was the primary data provider for this project, their internal data portal, 'VoxViewer' was used for various datasets, in addition to other data (e.g. MBES, Soil Samples) that was sent to us directly.

5.1.1 MBES Bathymetry Data

Figure 48. Channels present in the MBES dataset from Van Oord (own figure)

Van Oord provided MBES data for a number of channels (see Figure 48). The raw data from the sensor was processed using proprietary software. This software can interpret the raw measurements and output a point cloud. The point cloud was then resampled by the client into a 0.5x0.5m raster grid, which was ultimately provided to us.

We also had access to a small subset of the original point clouds, but the number of points (upward of 80 million for some channels) made working on this data directly impractical.

The available dataset was not of uniform quality for each channel. This necessitated a more detailed investigation of which side channels were viable for analysis. For each side channel, we investigated what data was available (i.e. whether it had 2025 bathymetric data, 2024 bathymetric data, 2025 AHN combined data, and 2024 K10 combined data). The data for a given side channel is only usable if we have sufficient temporal resolution (i.e. data for *both* 2024 and 2025). Otherwise, we are missing the difference between the two years. Table 6 shows a summary of the results of this investigation.

Table 6
Total number of associated datasets for all side channels

	2025 Bathy	2025 AHN	2024 Bathy	2024 Kavel 10	RWS	Usable
Total	24	26	36	41	20	26

The provided MBES data did not have a standardized naming convention for the side channels. Sometimes different channels were grouped, while other times they were separated. For example, Heeselt Oost, Heeselt West, and Hurwenen sometimes get grouped as Heeselt, while at other times they have their own file. Many channels also had different spellings across the different datasets, for example, Dordtse Avelingen instead of Dortsche Avelingen. It was important to clearly define the names for each side channel, the full list of which can be found in Appendix C. Some channels only had bathymetry available for them for one year. Since using AHN5 or Kavel10 data should be avoided for consistency reasons, we decided to clip these combined raster layers based on the extent of the bathymetry of the other year. Ideally, the original bathymetry should be used.

5.1.2 Polygons

We were also provided with polygons that represented the shape of the side channels. These polygons were given to us by Van Oord, but were originally generated by RWS. These polygons served as a basis for many features that required a concrete perception of the side channel geometry. For example, the Medial Axis Transform (MAT) is a process that operates on the polygon to generate a centerline, which is used in many feature implementations.

Within the provided polygons, islands were stored as separate internal polygons. These were removed from the database, so none of the polygons of the side channels contain holes. Features that require islands to be calculated properly can still make use of the original dataset for calculation.

5.1.3 Flow Threshold Data

Van Oord's VoxView data portal provided a flow map that contained flow volume threshold values for which each pixel would be underwater. The values are in m^3/s, and there were 10 threshold values in total (0, 1000, 1429, 2199, 3227, 4369, 5970, 6731, 8000, and 9955). The map covers the Waal, Nederrijn, and much of the Lek river. It unfortunately does cut off just short of some side channels within our usable dataset (Dordtsche Avelingen, Avelingen, Avelingerdiep, and half of Willige Langerak), and thus those tiles will be missing a feature.

The data was provided in the form of shape files for each threshold value, each individually containing the polygons that represented the areas with the corresponding value. As a preprocessing step, the individual shape files were merged into one master file, and each polygon was given an additional attribute that represented the flow threshold of its underlying layer.

5.1.4 Inlets and Outlets

For some feature calculations, the inlet and outlet points of the side channels are required. These were obtained from the ObjectPoints layer in VoxView.

However, some inlet and outlet points did not have a corresponding side channel polygon. To prevent these points from being incorrectly linked to other side channels, they were manually removed. The resulting subset—containing only inlet and outlet points associated with side channel polygons—was used for the feature calculations, distance from inlet, and flow direction.

For the features channel length and relative channel length, which do not depend on flow direction, points are needed at both ends of each side channel. Since some side channels, such as ponds or one-sided connected side channels, lack defined inlet and outlet points, additional points were manually added where necessary to enable these calculations.

5.1.5 Soil Data

Before any dredging is done, soil samples are collected from the relevant area as is required by Dutch law (Besluit activiteiten leefomgeving)[6]. This means that soil sample data should always be available for areas that have been dredged.

While Van Oord also performs the required sampling, the data is not available in a georeferenced format but only as PDF files. Because of this, we were not able to implement this feature, considering the timescale and scope of this project.

5.2 RWS

Besides the data provided by the client, we explored integrating a variety of data from various RWS data portals as well. Primarily, we investigated the various measurements found on their water info[7] and vegetation monitor[8] portals.

[6] https://wetten.overheid.nl/BWBR0041330/2025-09-20/#Hoofdstuk5_Afdeling5.2_Paragraaf5.2.2
[7] https://waterinfo.rws.nl/
[8] https://vegetatiemonitor.rijkswaterstaat.nl/#/veld

5.2.1 Vegetatiemonitor

For the past six years, RWS has published the so-called 'vegetatielegger' as open data in their vegetatiemonitor[9] web viewer. The vegetatielegger shows the designs for the type and amount of vegetation around the Rijntakken to ensure proper river flow capacity.

As part of this same web viewer, RWS also monitors the types of land use and vegetation around the Rijntakken to ensure proper management and upkeep. This data is based on satellite imagery, which is processed through a Google Earth Engine computer vision pipeline to detect 5 classes: (None/water, Sand/built up, Grass, Reeds, Forest, Shrubs). The data is openly accessible and available for download as a geo-referenced .tiff file.

5.2.2 AIS

Automatic Identification System (AIS) is a system installed on ships to ensure maritime safety when navigating. AIS enables effective communication between skippers as well as between skippers and control stations by broadcasting a ship's location in a standardized format. RWS collects and stores the location data for ships in Dutch waters to monitor shipping routes and conduct research.

However, because many skippers also live on their ships, they are legally considered residences. This makes the location data subject to strict European privacy laws, so there is no possibility for direct downloads. RWS provides anonymized or aggregated AIS datasets for researchers upon request, which would overcome this issue. Unfortunately, the AIS tracking system used by RWS was out of order for the duration of the project, so it was not possible to request the necessary data.

5.2.3 Water Height

RWS manages a number of measuring stations in and along major Dutch rivers. Among many other things, these stations measure water height in cm from NAP. Their viewer[10] displays these values and allows users to download historical measurements taken every 10 minutes.

While the measuring stations are relatively sparse, we were able to interpolate between them for use on normalizing bed level values (see Section 4.6.3).

5.2.4 Surface Water Flow

Another relevant measurement we wanted to take advantage of was the surface water flow in m^3/s. Their viewer[11] displays these values and allows for download of historical measurements taken every 10 minutes.

At the start of 2025, two new measuring stations were placed at the start of the Waal and Nederrijn rivers, which are in the study area of this project. Unfortunately, this means that there are no measurements available for 2024, and it is currently not possible to use this data for feature calculations.

5.3 OpenStreetMaps

Openstreetmap is an open data community building a freely available worldwide map (openstreetmap.org[12]). Community members record additions and continuously update the map, and often include semantic information. It is possible to query the map based on these semantic tags, as was done in this project. OSM polygon data was queried using the QuickOSM plugin for QGIS with the man_made=groyne tag. This layer was then used to assist in assigning the value for the 'inlet between groynes' feature.

[9] https://vegetatiemonitor.rijkswaterstaat.nl/#/veld
[10] https://waterinfo.rws.nl/publiek/waterhoogte
[11] https://waterinfo.rws.nl/expert/Afvoer?parameters=Debiet___20Oppervlaktewater___20m3___2Fs&view=map
[12] https://www.openstreetmap.org/about

6 Results

6.1 Feature Analysis

We performed statistical analysis on the tile-level features of two channels, the Bakenhof (Figure 49) and Afferden en Deest(Figure 50) side channels. These are chosen to see how the values differed between a morphologically simple channel like Bakenhof and a more complicated example like Afferden en Deest. The analysis serves as a proof of concept and sanity check that the features have valuable predictive power. Given that the analysis takes place for models trained on one channel, only the tile-level features are included for analysis as the channel-level features would be the same for all tiles.

Figure 49. Bakenhof side channel (own figure)

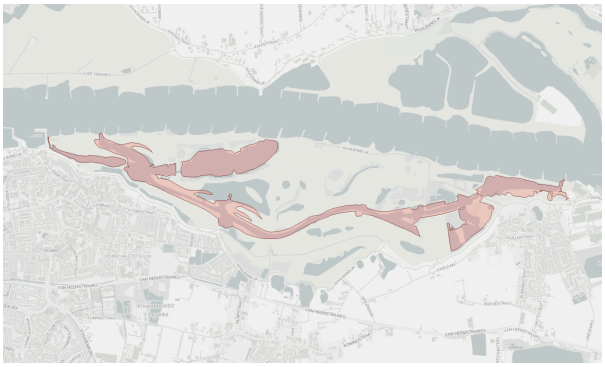


Figure 50. Afferden en Deest side channel (own figure)

Given the relatively large sample size of $N = 1917$, the statistical significance threshold for the Pearson correlation coefficient (r) is low. Any absolute correlation value $|r|$ greater than or equal to 0.0448 is considered statistically significant at the $\alpha = 0.05$ level. This means almost all observed relationships between the features are statistically significant. However, it is important to note that, while statistically significant, many of these correlations are very weak.

6.1.1 Merit

If the merit value is greater than 0.0448, that indicates that the feature is significantly beneficial to predicting sedimentation. Table 7 shows that most implemented features are suitable for predicting sedimentation in the Bakenhof and Afferden en Deest channels. Table 7 also shows that, depending on the channel, different features contribute in different amounts to the prediction. For example, roughness in the Bakenhof channel has a merit score that is more than three times that of the same feature in the Afferden en Deest channel. Similarly, the merit of `distance_from_inlet` is not statistically significant for the Bakenhof channel but is for the Afferden en Deest channel.

Table 7
Merit of different features for Bakenhof and Afferden en Deest

Feature	**Bakenhof**	**Afferden en Deest**
bed_level	0.151644	0.117676
slope	0.196017	0.080107
roughness	0.219961	0.068184
aspect_x	0.154710	0.053130
aspect_y	0.151217	0.039375
width	0.091637	0.018358
min_flow_threshold	0.205412	0.273423
flow_direction_x	0.020612	0.003091
flow_direction_y	0.014172	0.059818
distance_from_inlet	0.010258	0.259906
distance_to_bank	0.047583	0.079002

6.1.2 Pearson Correlation Coefficient

The correlation matrix in Figure 51 demonstrates that several features are correlated with each other. While this is expected to some extent, high correlation can lead to redundancy and instability in predictive models. Specifically, the strong correlation between roughness and slope is identified as problematic in both Figure 51 and Figure 52. To mitigate the negative effects of this multicollinearity, Principal Component Analysis (PCA) will be employed as a solution in the subsequent modeling prototype.

In Figure 51, the correlations between components of cyclical features (aspect and flow_direction) are noted for the Bakenhof channel, though this relationship is inherent to their definition as components of a single feature. Figure 52 does not show the same relationship between components of cyclical features. This difference can likely be explained by the difference in the shape of the two channels, Bakenhof being a lot simpler and straighter than Afferden en Deest. When channels become more complex and include many more angles and aspects, the relationship between the features of all the tiles becomes much more complex as well.

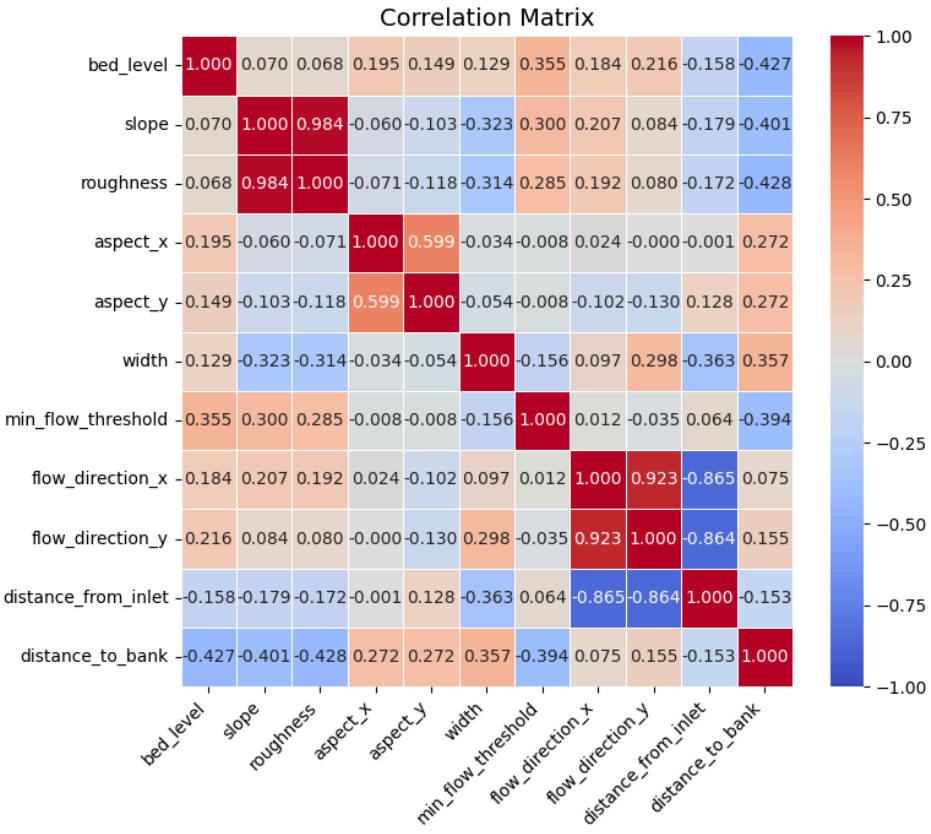


Figure 51. Correlation matrix of dynamic tile-level features for Bakenhof (own figure)

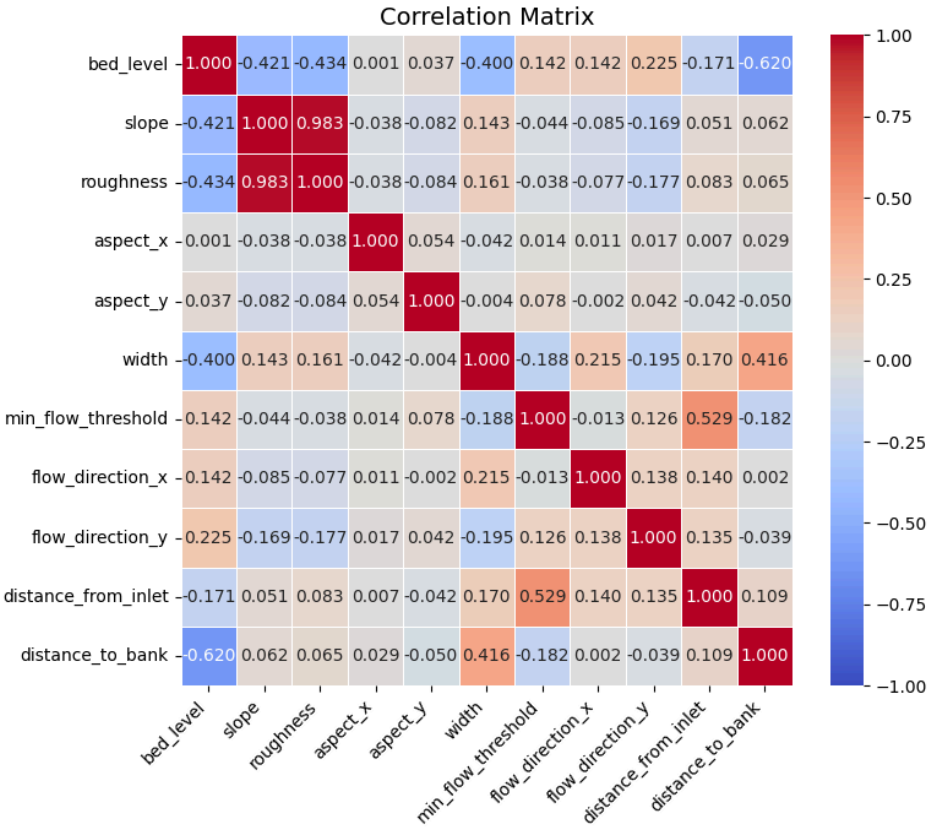


Figure 52. Correlation matrix of dynamic tile-level features for Afferden end Deest (own figure)

6.1.3 Variance Inflation Factor and Tolerance

Table 8 and Table 9 show that the calculated VIF values are sufficiently low ($\ll 10$) and tolerance values significantly high ($\gg 0.1$) for most features, indicating that multicollinearity should not pose a significant problem for the majority of the dataset (Cohen et al., 2003). The notable exceptions are roughness and slope, which exhibit high VIF values (36.07 and 35.66, respectively), confirming the issues identified in the correlation analysis. As previously stated, this issue will be addressed through the application of PCA in the prototype model.

In Table 8, aspect_x and aspect_y have VIF values around 1.75, which shows the expected correlation but is low enough that they are not expected to cause instability during the model training process. Additionally, flow_direction_x and flow_direction_y have high VIF values of 10.3 and 12.9, respectively. In accordance with the pairwise correlation in Figure 51

Table 8
Feature multicollinearity statistics for Bakenhof

Feature	VIF	Tolerance
bed_level	2.056726	0.486210
slope	36.666858	0.027273
roughness	36.997474	0.027029
aspect_y	1.771861	0.564378
aspect_x	1.751459	0.570952
width	2.446079	0.408818
min_flow_threshold	1.331975	0.750765
flow_direction_y	10.310373	0.096990
flow_direction_x	12.968219	0.077112
distance_from_inlet	6.798673	0.147088
distance_to_bank	2.809345	0.355955

Table 9
Feature multicollinearity statistics for Afferden en Deest

Feature	VIF	Tolerance
bed_level	2.467080	0.405337
slope	36.659545	0.027278
roughness	37.429641	0.026717
aspect_y	1.034960	0.966221
aspect_x	1.009159	0.990924
width	1.518738	0.658442
min_flow_threshold	1.797235	0.556410
flow_direction_y	1.169949	0.854738
flow_direction_x	1.171139	0.853869
distance_from_inlet	1.933817	0.517112
distance_to_bank	1.931996	0.517599

6.2 Multilayer perceptron

The prototype MLP model, despite its relative simplicity, does still require some tuning for the performance to reach acceptable levels. This section describes some of the involved processes.

6.2.1 Dimension Reduction

As explained in Section 4.6.4, we will first have to reduce the number of input features. In Figure 53, the learning curve is shown for a model that was run with all features present. While the loss of the training set continues to go down, it has no impact on the test loss. This indicates the model is overfitting on the noise present in the training data set, showing the need for dimension reduction. Both forward insertion and PCA were explored.

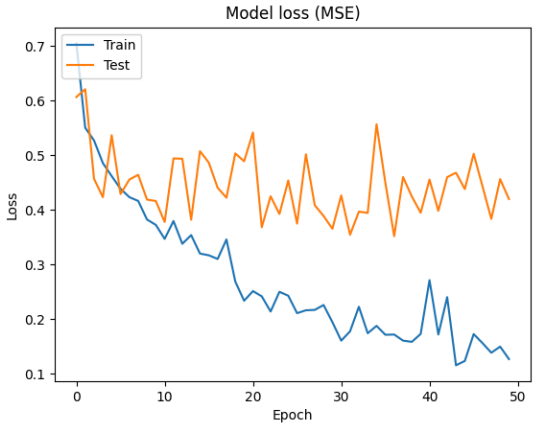


Figure 53. Learning curve of the MLP trained on all features for Bakenhof side channel (own figure)

Forward insertion resulted in six features that gave the best result for Bakenhof: aspect, minimum flow threshold, distance from inlet, width, bed level, and distance to bank. For PCA, 5 components explaining 86% of the variance resulted in the best model.

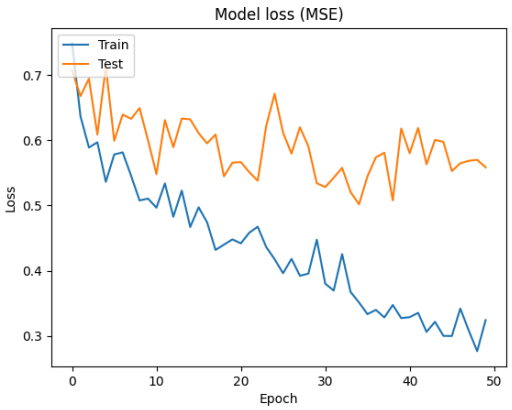


Figure 54. Learning curve using forward insertion with 6 features (own figure)

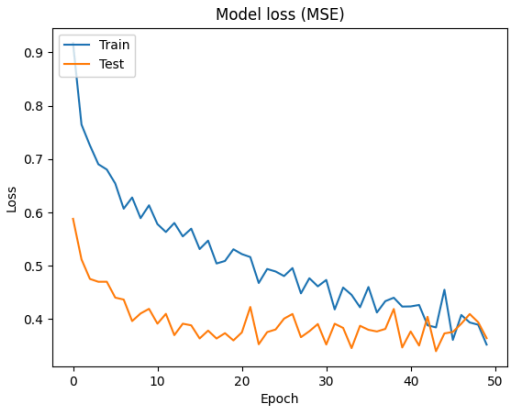


Figure 55. Learning curve using PCA with 5 components (own figure)

Figure 54 and Figure 55 highlight that PCA reduces overfitting and yields better results for the overall loss of the validation training set. Thus, in further models, PCA will be utilized for dimensionality reduction, and forward insertion will not be further considered.

6.2.2 Single Side Channel

The MLP – defined in Section 4.6.5 – was initially run for a single side channel at a time. The features were first reduced using PCA, of which five components were included. The explained variance is also recorded. To increase the stability of the results, the model was run five times for each side channel, and each of the results were averaged. The combination loss is the MSE (Mean Squared Error) of all tiles between the real change and the average prediction of the five runs. The learning rate may be different for each channel, depending on which value gave the best result. Table 10 shows the result of the model for a select number of side channels. For many one-sided connected side channels, like Schoutenwaard and Lexmond, no flow direction and distance from inlet was calculated and therefore not included in the model.

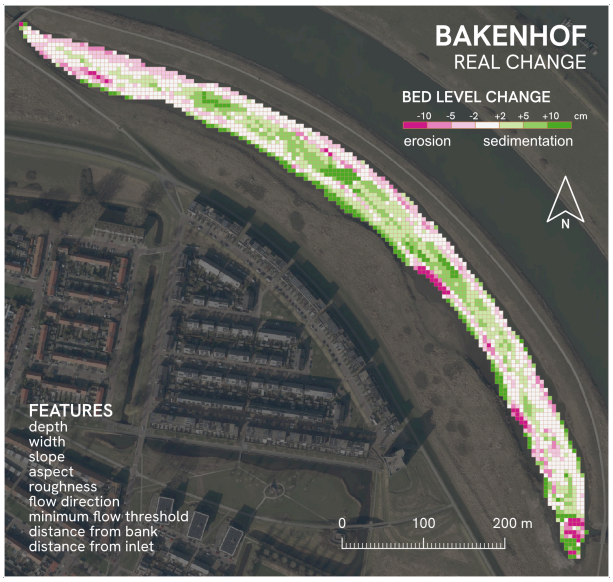
Table 10
Model results for select side channels. Loss is defined as the mean squared error. Centimeter values describe the percentage of tiles with such an error. 'Direction' is the percentage of tiles predicted in the correct direction (positive or negative)

Side Channel	**Learn Rate**	**Var Exp**	**Train Loss**	**Test Loss**	**Comb Loss**	**≤1cm**	**≤2cm**	**≤5cm**	**Direction**
Afferden en Deest	0,00001	73,27%	0,618	0,591	0,738	17%	34%	66%	71%
Bakenhof	0,0001	85,94%	0,394	0,372	0,300	36%	60%	88%	84%
Beneden Leeuwen	0,00001	76,17%	0,568	0,980	0,814	21%	39%	71%	69%
Klompenwaard	0,0001	79,40%	0,346	0,264	0,253	22%	41%	73%	82%
Lexmond	0,0002	83,23%	0,323	0,459	0,257	29%	51%	82%	82%
Schoutenwaard	0,0005	87,90%	0,411	0,355	0,273	55%	81%	95%	88%
Waalse Waard	0,0001	88,11%	0,287	0,274	0,212	34%	58%	87%	86%

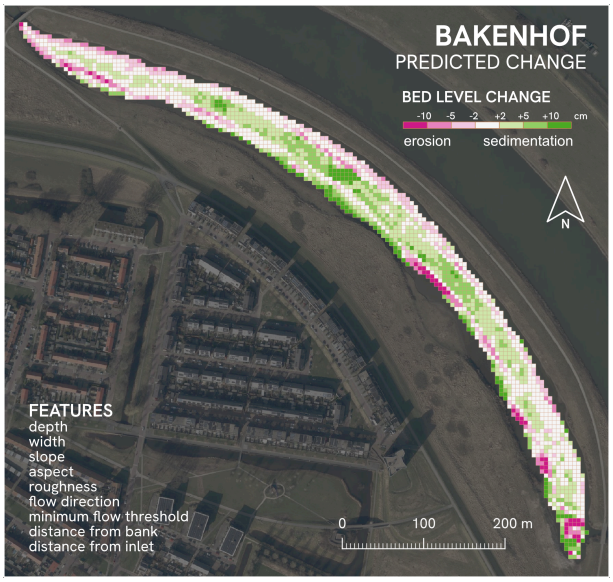
The model yielded good results for most of the side channels. In general, we see better performance on smaller side channels with a simple morphology, which indicates that less complex side channels are easier to predict. While a channel like Schoutenwaard appears to perform the best when considering errors over 5 cm, its loss values suggest a weaker performance compared to some other side channels. This can be explained by the limited variability in the bed level change of Schoutenwaard. As a result, its standard deviation is smaller, increasing the mean square error of the standardized change, used for the loss function. In other words, after standard scaling, one standard deviation in the Schoutenwaard corresponds to fewer centimeters than in other channels.

For others, like Beneden Leeuwen and Afferden en Deest, the channels were more complex as seen in the lower explained variance of the PCA. This made it harder for the model to converge and caused the final outcome to be less reliable. Other than providing more features or data, a different model setup, for example, a different width and depth, might result in a better prediction.

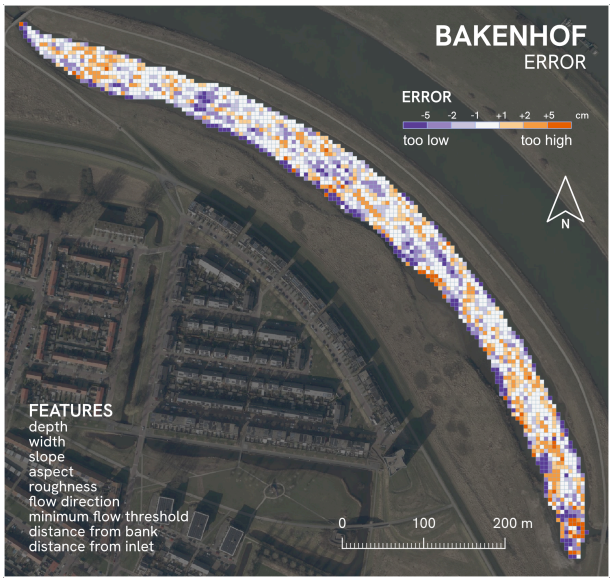
When looking at the spatial distribution of the error, visualized for Bakenhof in Figure 56 and for Schoutenwaard in Figure 57, certain regions inside the side channel appear harder to predict than others. Two main problem areas can be identified, namely, dynamic zones and river banks.



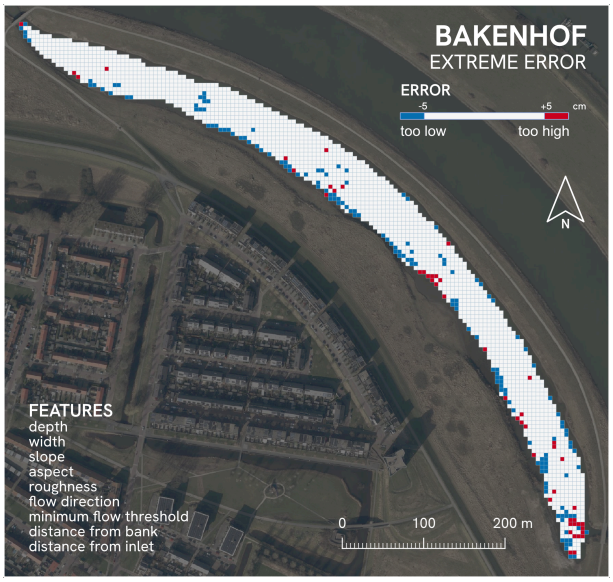
a. Real observed change of the Bakenhof side channel



b. Predicted change of the Bakenhof side channel



c. Real observed change of the Bakenhof side channel

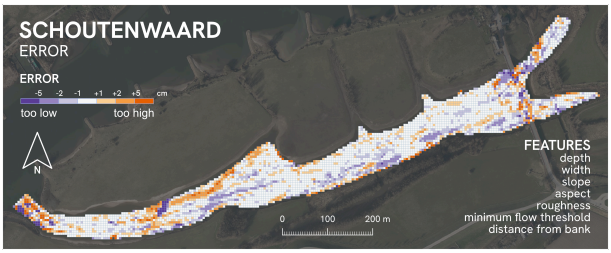


d. Predicted change of the Bakenhof side channel

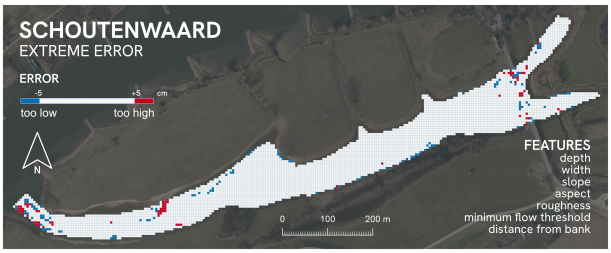
Figure 56. Visual representation of the MLP results of the Bakenhof side channels (own figure)

a. Real observed change of the Schoutenwaard side channel

b. Predicted change of the Schoutenwaard side channel



c. Real observed change of the Schoutenwaard side channel



d. Predicted change of the Schoutenwaard side channel

Figure 57. Visual representation of the MLP results of the Schoutenwaard side channels (own figure)

Dynamic zones exhibit high variability of bed level and feature complex flow patterns, which make it harder for the model to understand the changes. This is especially clear in areas where high erosion and sedimentation occur in close proximity to each other. An example of this is the inlet area, where turbulent flow and steep slopes, such as in the erosion hole, increase uncertainty in the prediction.

River banks are also more likely to produce errors. They often feature the highest rates of sedimentation or erosion, depending on which side of the channel they are on. While the model generally captures these trends, it tends to underestimate the magnitude. This effect could be due to the use of standard scaling, which reduces the influence of extreme values, or due to the spatial aggregation of data, where boundary effects and outliers are more common.

6.2.3 Multiple Side Channels

Training the MLP on multiple side channels at once might increase robustness and generalization, while also giving access to more data for small side channels that have trouble running on their own. To keep the training set balanced, however, the side channels that are grouped should be roughly equal in size and have the same number of channel connections, which narrows down possible groupings.

Even when channels meet these initial criteria, their behavior can still vary a lot. Channels with similar sizes and equal connection count can still display very unique behavior, adding more noise to the model rather than enforcing patterns. When deciding to run the model on different channels, the selection should be made carefully so as to still obtain good results.

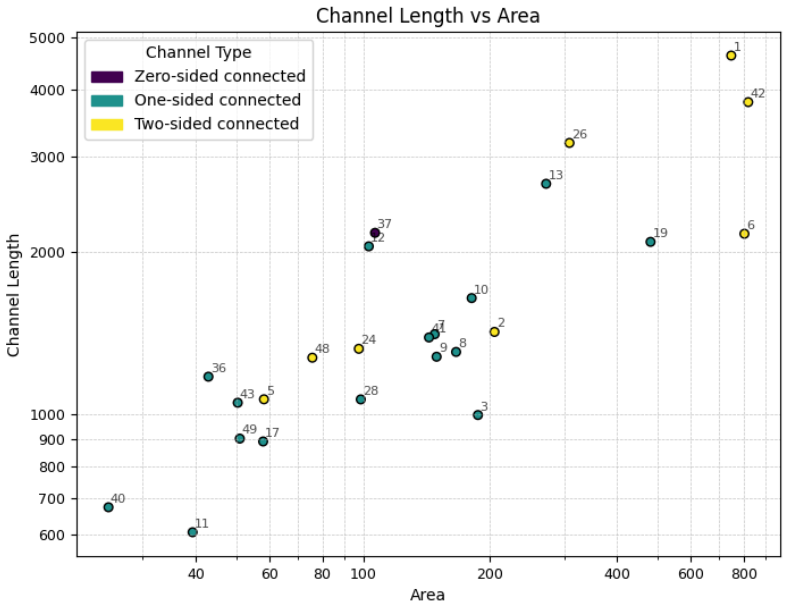


Figure 58. Plot of length, area, and connections labeled with side channel id (log scale) (own figure)

If we graph the side channels based on channel length and area, as shown in Figure 58, we can get an idea of what channels might be compatible to be trained by the same MLP. As expected, there is a large variation in side channels. Of the two-sided connected side channels, Bakenhof (5) appears to be most compatible with Klompenwaard (24) and Waalse Waard (48), which also showed good results when trained with a similar model in Section 6.2.2.

Since the model is being run with more data, it should be able to use more features without overfitting. Different amounts of components were tested, the one with the best result is shown in Table 11 with the number of PCA components listed. Two variations were tested, one where the global features are included and one where they are excluded.

Table 11
Model results for select side channels, as well as grouped channels with and without global features. Loss is defined as the mean squared error. Centimeter values describe the percentage of tiles with such error. 'Direction' is the percentage of tiles predicted in the correct direction (positive or negative)

Name	**Learn Rate**	**PCA**	**Var Exp**	**Train Loss**	**Test Loss**	**Comb Loss**	**≤1cm**	**≤2cm**	**≤5cm**	**Direction**
Bakenhof	0,0001	5	85,94%	0,394	0,372	0,300	36%	60%	88%	84%
Klompenwaard	0,0001	5	79,40%	0,346	0,264	0,253	22%	41%	73%	82%
Waalse Waard	0,0001	5	88,11%	0,287	0,274	0,212	34%	58%	87%	86%
Combo no global	0,0001	6	86,63%	0,377	0,359	0,274	29%	51%	81%	83%
Combo with global	0,0002	8	95,11%	0,363	0,356	0,285	30%	52%	81%	83%

Overall, the model still performs well when training with multiple side channels at once, but does not appear to offer any significant improvements over running the model on a single side channel. Most metrics end up somewhere between the values for the individual side channels, neither surpassing nor falling behind them. The inclusion of global variables does not appear to significantly improve the model either. The variation in these variables is probably too small for the model to extract patterns from. Maybe when training with more side channels, these features could have more predictive power, even though currently these groupings do not appear very viable, due to a limited number of channels and high variety among them.

The grouping of the smaller one-sided channels (11, 17, 36, 40, 43 & 49) did not result in a viable model, either due to the limited amount of data or the high variance between side channels. Using multiple side channels to obtain a robust model for small side channels might still be possible, but the ones currently available are not similar enough for this to be achievable.

7 Discussion

In this chapter, the results of the research are analyzed. In addition, it outlines recommendations and future work for our client, as well as a discussion of the limitations of the research.

7.1 Data Analysis

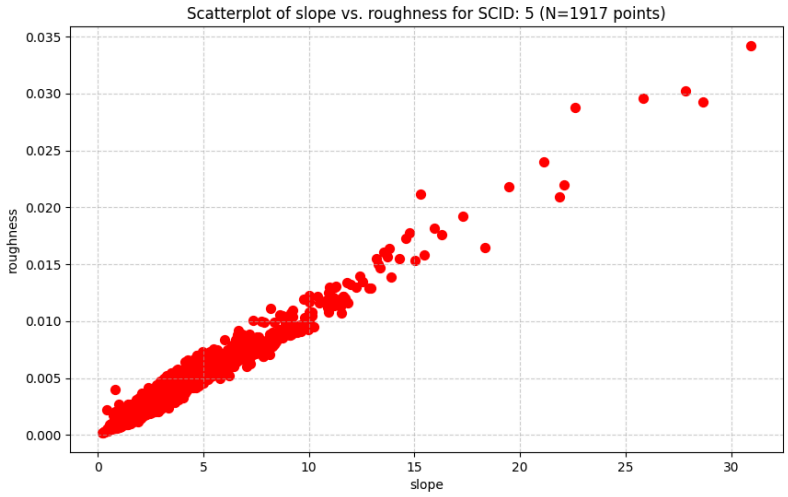


Figure 59. Plot of the slope and roughness features for the bakenhof channel (own figure)

Based on the merit and correlation analyses presented in Section 6, several adjustments to the final feature set can be considered to optimize model performance, reduce data redundancy, and improve computational efficiency.

The analysis, visualized in Figure 59, clearly indicated that slope and roughness are highly correlated, with high Variance Inflation Factor (VIF) values that suggest potential instability issues during model training. This correlation is likely because both features are calculated using similar kernel-based methods on the underlying bed-level data. Currently, this multicollinearity is addressed through PCA, which effectively merges or drops redundant information.

While effective, alternative approaches could be considered. One approach could be to simply drop one of the features from the database entirely. Given their redundancy, this would likely result in relatively little loss of predictive power while saving storage space and processing time. Another option would be to modify the calculation of one of the features. This could involve changing the scale of the kernel used for its calculation, aiming to make the two resulting features more differentiable and less correlated, or considering a new calculation for roughness than the currently implemented formula.

Features like flow direction and distance from inlet have low merit scores for the Bakenhof channel, showing they probably contribute less to the prediction of sedimentation compared to other features. It would be valuable to further investigate the added benefit of these features within model prototypes. This investigation should extend beyond the Bakenhof and Affenden en Deest study area to include other channels. If the contribution remains low across many channels, their exclusion could lead to a simpler, more efficient final database without a significant reduction in accuracy.

7.2 Model Performance

The model showed a strong performance for the prediction of certain individual side channels. Some channels, however, were still harder to predict, but this could be solved by adding more data from future measurements.

Other options that showed promise during testing included creating more local features like the curvature (second-derivative of the bed level) or change in width (derivative of width), and experimenting with more optimal model setups via more extensive testing.

Running the model on multiple side channels did not result in either a strong improvement or a reduction in performance. While this approach is not completely impractical, the amount of troubleshooting combined with the limited amount of side channels and high variety of behavior between them, makes this approach less ideal than single-channel models. Thus, if this single-channel model strategy is emphasized heavily in the future, one might consider dropping channel-level features entirely in favor of more and more varied tile-level features.

Currently, the model does not perform well on smaller side channels, due to the limited amount of tiles to train on. Initially, the tile size was standardized among all channels to facilitate the comparison between them. However, since this approach proved sub-optimal, a higher resolution could be applied to these channels to increase data and reduce the error due to aggregation.

7.3 Recommendations and Future Work

One of the major goals of the project was to derive a set of recommendations for Van Oord regarding how they should continue work on this product after we hand it off to them. Ten weeks were obviously enough time to implement everything we came up with, but it did give us time and insight to discover many potentially helpful next steps that can improve the performance of the pipeline. Here, we collect those insights and present them as paths for the client's future with this product. They are split into short-term recommendations concerning data management and long-term recommendations that encompass broader and more impactful changes.

7.3.1 Short Term

1) **Continue collecting MBES data and adding it to the database.**

At the moment, we have a single-year time step for only 26 channels. As you continue collecting data, you will begin to have a proper time series of data that can be analyzed to find more complex patterns than might be possible otherwise. More years of data collection will also unlock more side channels for analysis than the 26 we currently have.

That being said, at a certain point, we hope that this model can help you to decide moments when it is actually reasonable *not* to perform data collection for a given year. As the model becomes better trained on more data it can, at a certain point, ideally be used to perform a prediction for a channel to determine if it is reasonable to skip collecting MBES data that year, which would then save Van Oord time and money.

2) **Dredging activities**

While not relevant for the specific measurements so far (2024 and 2025), this is an extremely important characteristic. If this is not tracked, the model will be unable to account for the large, sudden changes. It will be important for the continued accuracy of the model to account for dredging when it takes place. Currently, in the `observations` table, there is a feature prepared (`dredged_in_last_year`). However, it has not been put to use in the final model as it was unclear when and where dredging may have taken place within the project scope.

If dredging does occur, you can make use of this feature to mark those measurements. Then, in the Python code, ensure that data before and after dredging are not used to train the model as if it were a "natural" erosion.

3) **Continue implementing and testing features for other identified characteristics**

Table 4 outlines the features that we had time to implement in the data processing pipeline. Thus, those features also form the basis on which the machine learning and prediction were based.

While we believe they make up a strong set of features that possess impressive predictive power, there are *many* others that we did not have time to implement, both in Table 3 and Appendix B. Some of these features may have very strong predictive power if you can find the time to implement them in the pipeline, which could further improve the performance of the model.

Some specific (non exhaustive) examples that we noted down during the project are as follows:

- Adding more *local*, tile level features, as they seemed to provide more predictive power than the global features.
- Collecting and using an "inlet type" feature that captures the inlet construction. We could not find data for this feature, but have encountered it repeatedly as an important characteristic.
- Implementing upstream discharge once the 2026 data is available.
- Implementing a feature for waves caused by ships once AIS is back up and RWS has had time to collect the needed data.
- Side channel geometry derived from satellite imagery. This would make the boundary polygons dynamic, which could allow for more accurate calculation of features like area or length.
- Integrate soil type data. If you can get soil data that indicates types of sediment or the diameter of the soil in a geo-referenced format, it could prove to be a strong predictor for sedimentation behavior.
- A discussion with expert Pepijn van Denderen revealed the importance of river groynes. Currently, we do model whether or not the inlet of a river is situated between two groynes, but we did not have time for the more sophisticated metric suggested by Pepijn. That being a relative position of the inlet between the two groynes (i.e. close to one side, in the center, etc.) This could prove to be a powerful feature if implemented well.

7.3.2 Long Term

1) **Consider more sophisticated machine learning models**

With only ten weeks to implement this project, there was only time for us to make a prototype machine learning model. The prototype model was a simple multi-layer perceptron.

What would be interesting to investigate is testing performance with some traditional machine learning approaches, like a random forest, as well as more sophisticated neural networks like convolutional neural networks (CNNs). The latter of which are trickier to design properly, but will likely showcase better performance on spatially/temporally correlated data that we are working with in this project.

Once enough data has been collected, you can also consider a deep learning approach. Such an approach was not chosen for this research because of poor performance when data is limited, as well as a lack of explainability. However, if after some years there is enough data, and explainability turns out not to be as valuable to stakeholders, then deep learning could be used to perform prediction with potentially even higher accuracy than the current approach.

2) **Compare the predictions and measurements to simulations done by a hydrologist**

Hydrologists would try to make predictions about side channel behavior by using their extensive knowledge of hydrological processes to model how they would expect all of the various parts of the system to behave.

Our research attacks the problem the other way around, aiming to predict the behavior of side channels purely in a data-driven manner. That is to say, we do not need to model explicitly how the features interact with each other, or concern ourselves with the complex hydrological formulas that dictate the processes we aim to predict. Naturally, this is very advantageous for us as Geomatics students who do not have any specific hydrology knowledge.

That being said, it may be worth investigating how our model compares to a more "traditional" approach, perhaps just for one side channel. There may even be ways to integrate hydrological simulation data into our model in the form of features to improve the performance.

7.4 Limitations

Over the course of the project, some limitations became clear both in terms of the implementation of the product and how the product itself performs.

7.4.1 Scope

To begin, the term 'side channel' in fact describes a very wide range of different structures, that are each individually *incredibly* complex, open-ended systems. A given one-sided connected channel and a given two-sided connected channel may (and often do) behave in vastly different ways. Even two

side channels that are relatively similar in terms of number of connections, length, etc. might behave very differently. Trying to make a system that is generic enough to encompass such different channels was a distinct challenge, and without a doubt means that the overall performance is going to vary from channel to channel.

In addition to different behavior, even something as simple as the shape of the channels being so different and unique led to some features being worse suited for certain channels than others.

7.4.2 Data

Another major limitation was that of data. We had access to only two years of usable data for most channels. This meant only *one* comparison from year to year. This makes any sort of robust prediction incredibly difficult. That being said, after encoding the data we had into features and testing the performance, we were pleasantly surprised with the quality of the output. We are confident that with another few years of data to train on, it might become quite a strong and reliable predictive tool.

This can also perhaps be achieved even more quickly if some more data sources are able to be implemented. We had hoped to implement features involving soil type, ship data, and river water volume in the pipeline, but for each of these data sources, there were issues (see Section 5) that led to us not being able to integrate them during the scope of the project. However, they all contain valuable information, and we hope that with more time, the client can make use of the data that we could not.

7.4.3 Technical

In addition to limited data, there was limited time and space. Time in the form of how much time we could spend implementing features before we needed to move on to analysis, and limited space in the form of the database used to store said features. The 5x5m grid was largely a practical choice made due to the limited hardware availability for processing the feature values, but primarily because the free database we used had strict space limitations. Luckily, our client has their own database solution, and thus are free to explore much more powerful, granular analysis than our student laptops and free database tiers could support.

7.4.4 Implemented features:

Although the implemented features were carefully designed, some limitations and potential improvements remain in their implementation.

Most features are currently computed using the side channel polygon shapes. However, these shapes vary with changes in water level. Furthermore, the side channel polygons were obtained from RWS, but the exact procedure used to extract these polygons is not known.

The applied spatial resolution of $5 \times 5m$ may also not optimally represent the data. Features such as *Bed level* and *Slope* tend to smooth out at this resolution, leading to loss of detail. Unfortunately, higher-resolution features could not be used due to restrictions on database access.

The feature *Channel length* is derived from the shortest path between inlet and outlet points. In cases where inlet or outlet points were missing, for example, for ponds or one-sided connected side channels, they were added manually. While in most instances the channel ends were clearly identifiable, this was more challenging for side channels with multiple branches. In such cases, the furthest and/or widest arm was selected as the endpoint.

For side channels that consist of disconnected parts, preventing the computation of a shortest path, the Cartesian distance between the inlet and outlet was used instead. Although this will be an underestimation of the channel length, we still decided to include it since the average *Relative channel length* = 1.15 with a standard deviation of $\sigma = 0.12$, suggesting that the Cartesian distance falls reasonably within the expected range of shortest-path channel lengths.

However, using the Cartesian distance for a unconnected side channel polygon as channel length results in a *Relative channel length* of 1, which is incorrect. This occurs in the side channels ‘Kleine Willemspolder’ and ‘Langsdam Wamel-Dreumel’.

The *Flow direction* feature is now based on the idea that the water flows from inlet to outlet, following the general shape of the side channel. Depth information was not included. This method may introduce

errors, particularly in areas with sharp bends or steep slopes. Flow direction algorithms based solely on gravitational forces, which base flow direction on the bathymetric dataset, were considered unsuitable, as they indicate to where water flows (downslope) but not how it moves within a flow system. Similarly, computational fluid dynamics was also not an option for us, since we want our model to be usable for our client, it is out of scope for this project, and also, we want a 2D feature rather than a 3D simulation, which we would get from computational fluid dynamics.

The Width feature is computed as the length of the line perpendicular to the centerline. However, unusually high width values occur near centerline branches (see Figure 60 → 1). To mitigate this, nearest-neighbor interpolation is applied to assign width values to tiles that do not intersect the centerline. Nevertheless, this may result in some tiles inheriting width values from unrelated centerline segments (see Figure 60 → 2). Consequently, the implementation performs less reliably for side channels with strongly branched and/or disconnected centerlines.

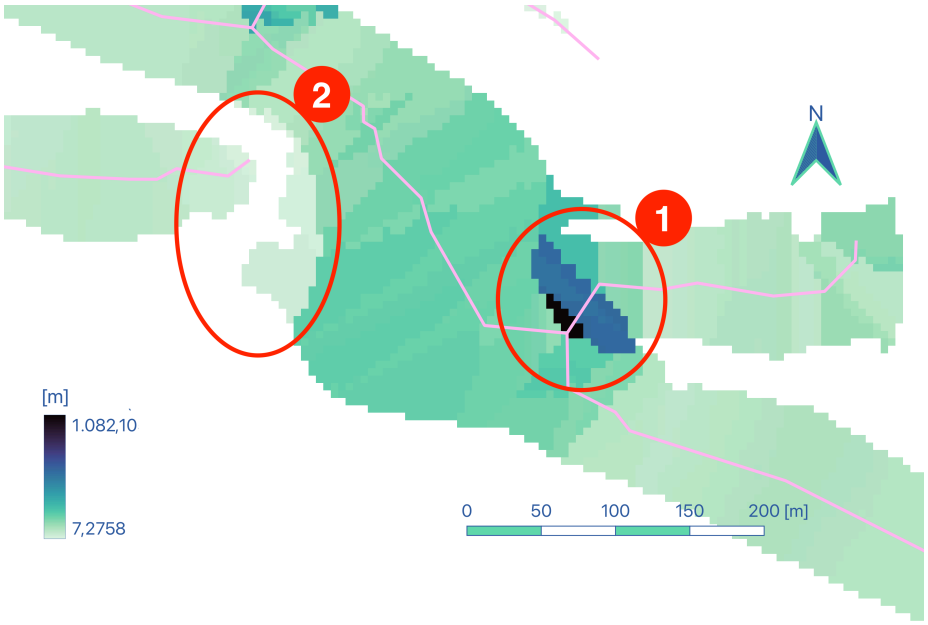


Figure 60. High width values at centerline branches (own figure)

The Distance from inlet feature computes distances from the inlet to points placed along the centerline, using nearest-neighbor interpolation to fill missing tile values. This approach has two main drawbacks. First, when the inlet point lies far from the centerline (as in 'Beneden Leeuwen'), it may yield unrealistically high distance values near the inlet (see Figure 61). Second, for disconnected polygons, where no valid path can be determined, interpolated values may not be meaningful (see Figure 62).

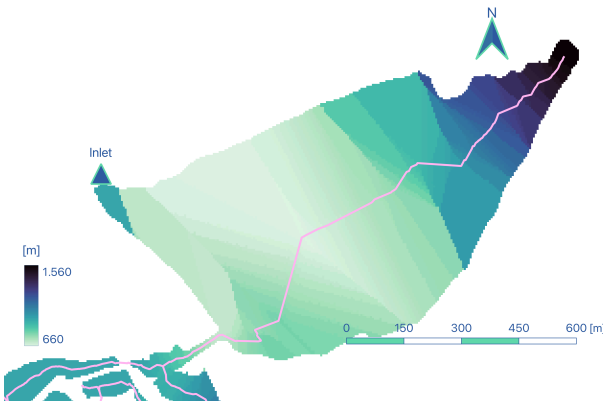


Figure 61. Inlet far from centerline resulting in overestimated *Distance from inlet* values (own figure)

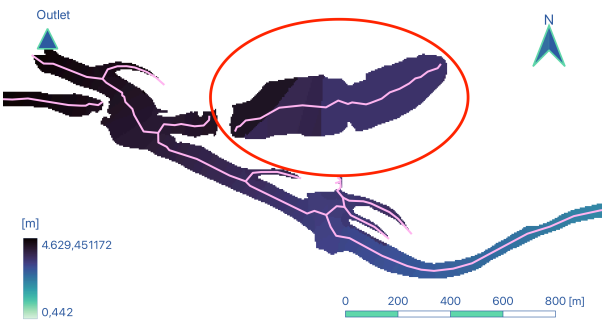


Figure 62. Disconnected polygons resulting in false *Distance from inlet* values (own figure)

For the Minimum flow threshold feature, the flow map from VoxView was used. However, for the side channels ‘Dordtsche Avelingen’, ‘Avelingen’, ‘Avelingerdiep’, and part of ‘Willige Langerak’, sections were cut off. Consequently, some values are missing for these areas.

Finally, the implementation of the *Aspect* feature requires that the bathymetry dataset be resampled to the $5 \times 5m$ grid before aspect computation. This is crucial because aspect values are non-scalable due to:

1) Averaging angular data produces invalid results (see Figure 63 vs Figure 64)

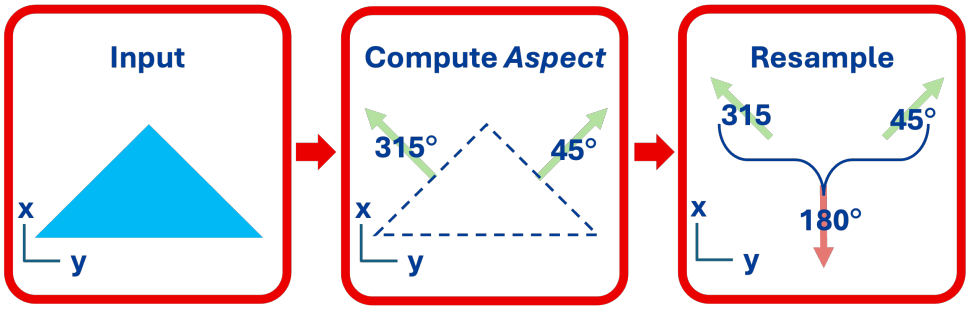


Figure 63. Wrong aspect computation when first computing the aspect and then resampling (own figure)

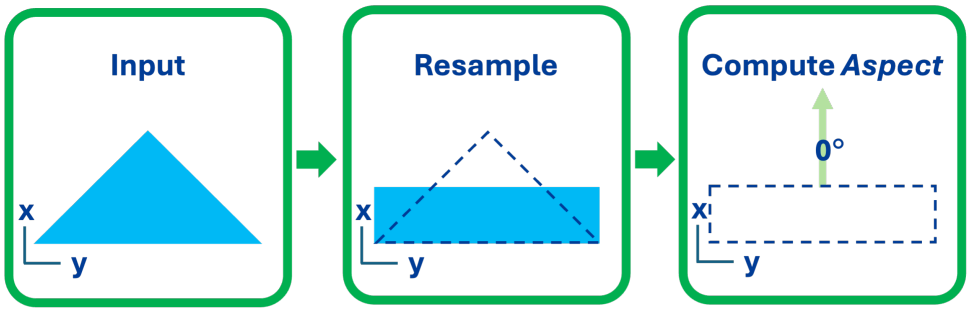


Figure 64. Correct aspect computation when first resampling and then computing the aspect (own figure)

2) and aspect does not account for slope steepness (see Figure 65 vs Figure 66)

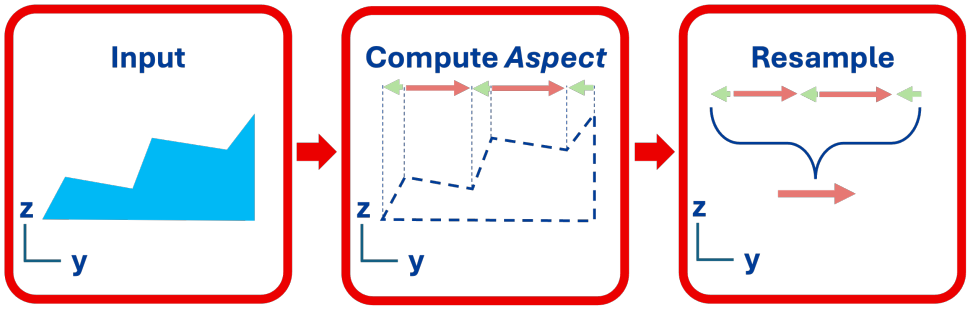


Figure 65. Wrong aspect computation when first computing the aspect and then resampling (own figure)

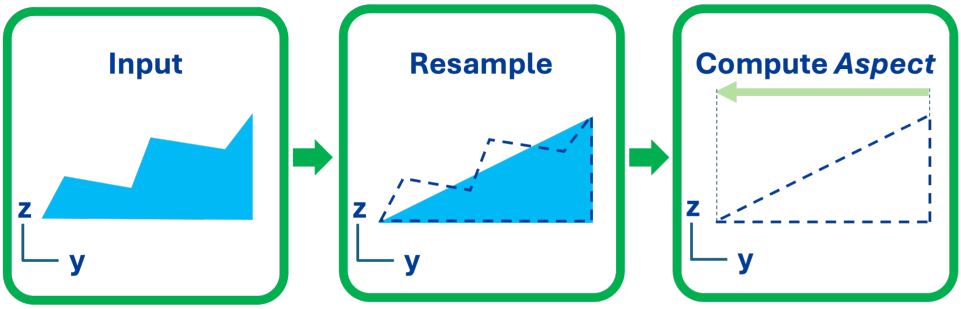


Figure 66. Correct aspect computation when first resampling and then computing the aspect (own figure)

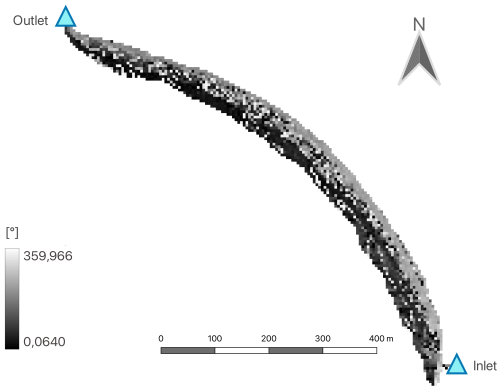


Figure 67. Wrong aspect computation when first computing the aspect and then resampling (own figure)

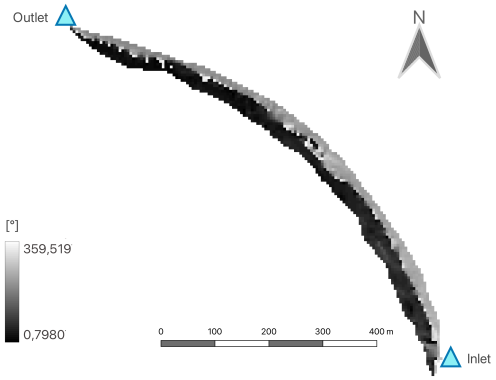


Figure 68. Correct aspect computation when first resampling and then computing the aspect (own figure)

Therefore, resampling is performed before the aspect calculation. Figure 67 and Figure 68 show the difference caused by the ordering of operations. However, FME restricts kernel size to a maximum of 4x4, whereas resampling from a $0.5 \times 0.5m$ grid to a $5 \times 5m$ grid ideally requires a 10x10 kernel. As a result, only 64 of 100 cells are used during resampling.

8 Conclusion

In this project, we set out to answer the question "***How can features of a side channel be identified and extracted to enable predictive maintenance?***". Our goal was to investigate the viability of extracting predictions *solely* from the available data, without needing any in-depth knowledge of the phenomena themselves. This would be accomplished by leveraging feature engineering and machine learning rather than the traditional approach of developing convoluted models based on the physical equations governing the phenomena, and using those for prediction.

Despite some limitations, such as the temporal availability of data, limited time to implement a solution, and the generally complicated nature of river-side channels, we implemented a pipeline that satisfies the requirements of the client and should only grow more reliable in the future as more data becomes available.

The FME workbench encapsulates the data extraction process that translates raw data into usable, interpretable features that are stored in a geospatial database, and the data analysis prototype, despite being based on a relatively simple multi-layer perceptron, can make impressively accurate estimations on the sedimentation of side channels. For example, the model trained on just the Bakenhof channel is able to predict 88% percent of tiles correctly within a 5-centimeter margin. 84% of all tiles that are predicted, are predicted to move in the correct direction. For only having two years of data to work from as training, this performance was impressive.

The product we deliver to our client Van Oord, is built with the expectation that over time, they can iterate upon our work to improve performance by adding new and novel features, adding data from more years of measurements to refine prediction results, and implement more sophisticated machine learning solutions such as convolutional neural networks to extract even more predictive power from the provided features.

Using this product, various doors are opened for the client's predictive maintenance workflows that otherwise may remain closed. Over time, as predictions become more reliable, predictions for the following years can be used for decision makers to, for example, decide to skip a year of MBES measurements, which saves the company quite some money. It can also be used for experts within the company to get an idea of when side channels may be at risk of breaching RWS guidelines in the near future, which can inform decision makers about upcoming preventative dredging earlier than would be possible otherwise.

Furthermore, the conceptual process of a data processing and feature extraction pipeline to encode information about some physical phenomena, followed by a machine learning prediction step, could even be applied to many other phenomena in order to enable predictive maintenance. For example, another task of Van Oord in the WOCU Rijntakken project is that of mowing and maintaining vegetation on the river banks. With appropriate data collection and feature encoding, it is possible that a similar prediction engine could be developed to assist stakeholders in deciding when and where it is most important to perform maintenance on vegetation.

Lastly, as students, we successfully managed to synthesize knowledge from much of our first year of education together in one cohesive research task as can be seen in Appendix D.2. This project provided a practical example of how to put our various skills to the test and taught us all a lot about geospatial data handling, data analysis, FME, geospatial databases, etc. We are all very glad to have had the opportunity to work on this project, and hope Van Oord can make good use of it in the future.

References

Adams, A., & Vamplew, P. (1998). Encoding and Decoding Cyclic Data. *The South Pacific Journal of Natural Science*, *16*, 55. https://www.researchgate.net/publication/316961463_Encoding_and_Decoding_Cyclic_Data

Ahmed, M. N., & Mohanadhas, B. (2025). Review of the effect of river vegetation on sediment transport and river morphology and erosion prevention. *ISH Journal of Hydraulic Engineering*, *31*(2), 398–408. https://doi.org/10.1080/09715010.2025.2471362

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer. https://www.microsoft.com/en-us/research/wp-content/uploads/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf

Chen, L. (2009). Curse of Dimensionality. In L. Liu & M. T. Özsu (Eds.), *Encyclopedia of Database Systems* (pp. 545–546). Springer US. https://doi.org/10.1007/978-0-387-39940-9_133

Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (p. 423). Lawrence Erlbaum. https://eli.johogo.com/Class/CCU/SEM/_Applied%20Multiple%20Regression-Correlation%20Analysis%20for%20the%20Behavioral%20Sciences_Cohen.pdf

Denderen, P. van, Schielen, R., Blom, A., Hulscher, S., & Kleinhans, M. (2017). Morphodynamic assessment of side channel systems using a simple one-dimensional bifurcation model and a comparison with aerial images. *Earth Surface Processes and Landforms*, *43*, 1169–1182. https://doi.org/10.1002/esp.4267

Dolan, M., Connell, Brown, C., Guinan, J., & Grehan, A. (2007). Multiscale Terrain Analysis of Multibeam Bathymetry Data for Habitat Mapping on the Continental Slope. *Marine Geology*, *927*, . https://doi.org/10.1080/01490410701295962

Domingos, P. (2012). A Few Useful Things to Know About Machine Learning. *Commun. ACM*, *55*, 78–87. https://doi.org/10.1145/2347736.2347755

Hai, X., Liu, G., Liu, P., Zheng, F., Zhang, J., & Hu, F. (2017). Sediment transport capacity of concentrated flows on steep loessial slope with erodible beds. *Scientific Reports*, *7*, 2350. https://doi.org/10.1038/s41598-017-02565-8

Hastie, T., Friedman, J., & Tisbshirani, R. (2017). *The elements of Statistical Learning: Data Mining, Inference, and prediction*. Springer. https://doi.org/https://doi.org/10.1007/978-0-387-84858-7

Jaiswal, S. (2025, April). *Multilayer perceptrons in Machine Learning: A comprehensive guide*. https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning

Kutner, M. H., Nachtsheim, C., Neter, J., & Li, W. (2005). *Applied Linear Statistical Models* (pp. 406–410). McGraw-Hill Irwin. https://users.stat.ufl.edu/~winner/sta4211/ALSM_5Ed_Kutner.pdf

Mahato, P. K., Singh, D., Bharati, B., Gagnon, A. S., Singh, B. B., & Brema, J. (2022). Assessing the impacts of human interventions and climate change on fluvial flooding using CMIP6 data and GIS-based hydrologic and hydraulic models. *Geocarto International*, *37*(26), 11483–11508. https://doi.org/10.1080/10106049.2022.2060311

Meijer, D., & Winden, A. van. (2020, April). *Morfologische Ontwikkeling Van Nevengeulen*. Kragten. https://www.stroming.nl/nl/overzicht/morfologische-ontwikkeling-van-nevengeulen

Ministerie van Infrastructuur en Waterstaat. (2025, April). *Ruimte voor de Rivieren*. https://www.rijkswaterstaat.nl/water/waterbeheer/bescherming-tegen-het-water/maatregelen-om-overstromingen-te-voorkomen/ruimte-voor-de-rivieren

Mosselman, E. (2001). *Morphological development of side channels*. IRMA-SPONGE, Delft Cluster.

NOAA Photo Library. (2016,). *Artist's conception of multibeam sonar on NOAA Ship NANCY FOSTER*. https://commons.wikimedia.org/wiki/File:Fis01334_(27555144884).jpg

Ottevanger, W., & Chavarrias, V. (2019). *Morphological development of the bifurcation at Pannerden*. https://publications.deltares.nl/11203682_007.pdf

QGIS. (n.d.). *24.2.1. raster analysis*. Retrieved October 1, 2025, from https://docs.qgis.org/3.40/en/docs/user_manual/processing_algs/gdal/rasteranalysis.html#roughness

Ripple, W. J., Wolf, C., Mann, M. E., Rockström, J., Gregg, J. W., Xu, C., Wunderling, N., Perkins-Kirkpatrick, S. E., Schaeffer, R., Broadgate, W. J., Newsome, T. M., Shuckburgh, E., & Gleick, P. H. (2025). The 2025 state of the climate report: a planet on the brink. *Bioscience*. https://doi.org/10.1093/biosci/biaf149

Riquier, J., Piégay, H., Lamouroux, N., & Vaudor, L. (2017). Are restored side channels sustainable aquatic habitat features? Predicting the potential persistence of side channels as aquatic habitats based on their fine sedimentation dynamics. *Geomorphology*, *295*, 507–528. https://doi.org/https://doi.org/10.1016/j.geomorph.2017.08.001

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, *2*(6). https://doi.org/10.1007/s42979-021-00815-1

Shields, A. F. (1936). *Application of similarity principles and turbulence research to bed-load movement* [Doctoral dissertation]. https://resolver.tudelft.nl/uuid:a66ea380-ffa3-449b-b59f-38a35b2c6658

Simons, J. H., Bakker, C., Schropp, M. H., Jans, L. H., Kok, F. R., & Grift, R. E. (2001). Man-made secondary channels along the River Rhine (The Netherlands); results of post-project monitoring. *Regulated Rivers: Research & Management*, *17*(4–5), 473–491. https://doi.org/https://doi.org/10.1002/rrr.661

Team Rijntakken. (2023, August). https://rijntakken.nl/team-rijntakken/

Thus, D. (2025). *Morphological development of secondary channels*. preprint.

van Denderen, R. P., Schielen, R. M., Westerhof, S. G., Quartel, S., & Hulscher, S. J. (2019). Explaining artificial side channel dynamics using data analysis and model calculations. *Geomorphology*, *327*, 93–110. https://doi.org/https://doi.org/10.1016/j.geomorph.2018.10.016

Vendrig, K. (2001). Nevengeulen biodiversiteit in Waal, verhogend?. *RWS, RIZA*. https://open.rijkswaterstaat.nl/@197727/nevengeulen-waal-biodiversiteit/

Zevenbergen, L. W., & Thorne, C. R. (1987). Quantitative analysis of land surface topography. *Earth Surface Processes and Landforms*, *12*(1), 47–56. https://doi.org/https://doi.org/10.1002/esp.3290120107

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, *150*, 106889. https://doi.org/https://doi.org/10.1016/j.cie.2020.106889

A Glossary

Term	Dutch	Definition
AIS		Automatic Identification System is a tracking system used on ships to identify and locate vessels
Bed shear stress	bodemschuifspanning	The frictional force per unit area that a flowing fluid exerts on the riverbed
Bifurcation	Splitsing	The point where a river's main channel splits into two or more separate channels, which then continue downstream
Characteristic	-	Some real-life phenomena pertaining to side channels that have an affect on their morphology. For example, the flow rate and direction of the water within the channel.
Co-flow	Meestroom	A side channel is said to co-flow if water flows from the river into an inlet, and out from an outlet back into the river. This does not happen all year round for many channels, but only when the water level reaches a certain threshold height.
Discharge	Waterafvoer	This is the quantity of water in m^3/s that flows through a channel. For our purposes it usually describes the water flowing upstream from the main channel.
Feature	-	Measured or calculated attributes used to describe a system that an algorithm can use. For example the slope of the bed within a 5x5m grid.
Feature Engineering	-	The process of transforming raw measurements into features computers can access.
Feature Space	-	The conceptual multi-dimensional space where each data point is represented as a vector.
Fluvial flooding	Fluviale overstroming	Fluvial flooding, also known as river flooding, occurs when a river or stream overflows its banks due to excessive rainfall or snowmelt in its catchment area.
Groyne	Krib	A rigid aquatic structure built perpendicularly from a river bank limiting the movement of sediment.
MBES	-	Multibeam Echo Sounder is a type of sonar that uses multiple sound waves to create a high-resolution, 3D map of the bed of a body of water.
Machine Learning	-	Machine learning is a process by which a computer system can learn or adapt to a certain task without explicitly being designed to do so. By leveraging certain algorithms and statistical approaches, these techniques can extract patterns and predictions from the data that it is trained on.
Main channel	Hoofdgeul	The main waterway or river side channels can branch off off (e.g. Rhine, IJssel)
MLP	-	Multilayer Perceptron. A type of neural network using multiple layers of interconnected neurons, with weights and biases, to predict complex patterns in the training data
Morphology	Morfologie	The study and analysis of the shape, form, and spatial arrangement of the sea- or riverbed

PCA	-	Principal component analysis. A feature reduction technique that remaps features into new components along the axes of most explained variance.
RWS	Rijkswaterstaat	Department of Waterways and Public Work is the executive agency of the Ministry of Infrastructure and Water Management of the Netherlands.
Side Channel	Nevengeul	A secondary channel that branches off and runs alongside a main channel.
USV	-	Unmanned surface vehicle sometimes called a drone boat or drone ship
VIF	Variantie-inflatiefactor	The Variance Inflation Factor (VIF) is a measure that is used to express how much the variance of a feature is inflated due to multicollinearity
VoxView	-	Van Oord's internal data viewer, used as a data source for some datasets (e.g. inlets and outlets)
Weir	Stuw	A low dam built across a river to raise the level of water upstream or regulate its flow
WOCU	-	Waardengedreven OnderhoudsContract Uiterwaarden

B List of Identified Characteristics

B.1 Characteristics List

Table 12 shows the original list of channel characteristics generated during the 'Discover' step of our project as desscribed in Section 4.2

Table 12
All identified side channel sedimentation characteristics with references and commentary on their use

Characteristic	References	Scale	Impact	Effort	Comments
Depth	(Mosselman, 2001), (Riquier et al., 2017), (Denderen et al., 2017)	tile	High	Low	Bed level data is provided by client, depth can be easily calculated
Length of the channel	(Thus, 2025), (van Denderen et al., 2019), (Denderen et al., 2017)	channel	High	Low	Easily extracted from centerline of channel
Slope / Aspect	(Riquier et al., 2017), (Ottevanger & Chavarrias, 2019)	tile	High	Low	Easily calculated from the depth information
Channel Classification	(Meijer & Winden, 2020; Riquier et al., 2017)	channel	High	Low	Data is provided by client
Angle between main channel and side channel	(Thus, 2025), (Mosselman, 2001), (Riquier et al., 2017), (Denderen et al., 2017)	channel	High	Med	Requires some calculations between main and side channel, but is not especially complicated
Bends in side channel	(Thus, 2025)	tile	High	Med	Can be derived from the shape of centerline
How many days a year the channel is active	(Meijer & Winden, 2020; Riquier et al., 2017)	channel	High	Med	Data provided by client
Length relative to main channel	(Mosselman, 2001), (Riquier et al., 2017), (Denderen et al., 2017)	channel	High	Med	A simple ratio between length of side channel and corresponding length of main channel
Upstream discharge	(Thus, 2025), (Meijer & Winden, 2020; van Denderen et al., 2019)	channel	High	Med	Data provided by RWS
Width	(Thus, 2025), (van Denderen et al., 2019), (Riquier et al., 2017), (Denderen et al., 2017)	tile	High	Med	Requires calculations but can be derived with centerline and polygons of side channels
Bed level	(Thus, 2025), (van Denderen et al., 2019), (Ottevanger & Chavarrias, 2019)	tile	Med	Low	Bed level data provided by client
Distance from inlet	(Meijer & Winden, 2020)	tile	Med	Low	Distance from inlet via centerline
Roughness	(van Denderen et al., 2019)	tile	Med	Low	Can be approximated using the standard deviation, which is provided by the client
Flood frequency	(Mosselman, 2001), (Riquier et al., 2017)	channel	Med	Med	Data provided by client
Side channel in an inner or outer bend of the main channel	(Mosselman, 2001), (Denderen et al., 2017)	channel	Med	Med	Can be easily input manually
Vegetation near the channel	(Thus, 2025), (Mosselman, 2001)	tile	Med	Med	Data provided by client
Waves caused by ships	(Thus, 2025), (van Denderen et al., 2019), (Mosselman, 2001), (Meijer & Winden, 2020)	channel	Med	Med	Data provided by AIS
Obstacles (bridge/island) in the channel	(van Denderen et al., 2019), (Mosselman, 2001)	tile	High	High	Might be possible with a relatively complex derivation to identify what is an obstacle (hand labelled?), and whether a given cell is upstream or downstream from it, then a distance.
Threshold Height	(Meijer & Winden, 2020)	channel	High	High	How do we determine? Possible with available data but would take some work.
Design of inlet structure	(Thus, 2025), (Riquier et al., 2017)	channel	High	Extreme	Complete lack of available data, difficult to classify

Flow Capacity	(Riquier et al., 2017)	channel	High	Extreme	Would need to perform simulations at different main channel discharges, out of scope
Inlet Speed	(Meijer & Winden, 2020)	channel	High	Extreme	Would need measurements of flow velocity at the inlet
Sediment transport within the main channel	(van Denderen et al., 2019)	channel	High	Extreme	Lack of data, how to relate to side channels
Side channel discharge / Percentage of total discharge	(Thus, 2025), (van Denderen et al., 2019), (Meijer & Winden, 2020; Mosselman, 2001)	channel	High	Extreme	Likely requires simulations, depends on threshold height, etc.
Transverse bed slop (Bed level difference between main and side channel)	(Denderen et al., 2017), (Ottevanger & Chavarrias, 2019), (Mosselman, 2001)	channel	High	Extreme	Missing bathymetric data for the main channels.
Bank Erosion	(van Denderen et al., 2019), (Denderen et al., 2017)	tile	Med	High	Difficult implementation with given data, would potentially need to use high-resolution satellite imagery?
Bends in the main channel	(Thus, 2025)	channel	Med	High	Effect of this characteristic very difficult to model
Orientation angle (defined as the average angle between the side channel and floodplain flow lines during a flood)	(Mosselman, 2001)	tile	Med	High	Difficult to determine floodplain with given data, too low of impact for the effort of implementation
Sediment Size	(van Denderen et al., 2019), (Denderen et al., 2017), (Ottevanger & Chavarrias, 2019)	channel	Med	High	Lack of data, would probably need to measure by hand
Bed Shear Stress	(van Denderen et al., 2019), (Riquier et al., 2017)	tile	Med	Extreme	Incredibly difficult to derive
Difference in channel slope	(Denderen et al., 2017)	channel	Med	Extreme	Would require bathymetric scans of main channel near side channel, difficult to localize
Flow Velocity	(van Denderen et al., 2019), (Mosselman, 2001)	tile	Med	Extreme	Flow velocity data would need to be taken for each channel in multiple locations, RWS data is far too sparse to be useful
Vegetation inside the channel	(Mosselman, 2001), (Denderen et al., 2017)	tile	Med	Extreme	Very difficult to measure, would need detailed scans for each side channel to associate with tiles
Soil type	(van Denderen et al., 2019)	channel	Low	Med	How to derive a feature from this data? Not many samples so not useful to apply to all side channels.
Waves caused by wind	(Meijer & Winden, 2020)	channel	Low	Med	Missing detailed data, not very impactful
Bend flow upstream of the bifurcation	(Denderen et al., 2017)	channel	Low	High	Difficult to derive and define for a given side channel, lack of available data
Groyne(s) in main channel	(van Denderen et al., 2019)	channel	Low	High	Would perhaps need to be implemented with computer vision on satellite imagery? Too low impact for the effort.
Weirs	(van Denderen et al., 2019), (Mosselman, 2001), (Riquier et al., 2017)	channel	Low	High	Lack of data, how to classify (yes/no?), low impact

B.2 Characteristics List Updates

Table 13 shows the updates to the list of features reached via collaboration with experts Pepijn van Denderen and Hans van der Kwast.

Table 13
Updates to the characteristics based off of expert opinion. Bold entries in impact or effort were changed from the original list, and expert comments are noted on the far right column.

Characteristic	Impact	Effort	Expert comments
Bed level	**High**	Low	Probably a better feature than depth, as depth is difficult to define, and bed level is already provided
Length of the channel	High	Low	Probably correlated with global slope
Roughness	**High**	Low	Probably higher impact
Waves caused by ships	**High**	Med	Especially impactful on sedimentation of the Rhine as there is relatively low flow velocity
Angle between main channel and side channel	High	Med	Confirmed important in alignment with literature
Upstream discharge	High	Med	Confirmed important
Groyne(s) in main channel	**High**	Med	Upstream groynes cause complicated flow patterns. Distance to the groine will effect the type of incoming sediment
Side channel in an inner or outer bend of the main channel	Med	Med	Quite important however most side channels in the Netherlands are in the outer bend
Vegetation near the channel	Med	Med	Could have an effect on the bank stability, and thus erosion in the channel
Depth	**Med**	**Med**	Probably has very high overlap with bed level and isn't as straightforward to define
Obstacles (bridge/island) in the channel	High	High	Relevant for "units" approach
Threshold Height	High	High	Actually part of the inlet structure, and is indeed a valuable piece of data
Sediment Size	**High**	High	Important for transport mechanism
Design of inlet structure	High	Extreme	Confirmed very important, be sure to include the threshold hight at which the channel co-flows
Soil type	**Med**	High	Strongly related to sediment size

C Side Channel MBES Data Availability

Table 14
Side channels present in the provided data, along with associated datasets

Side Channel	2025 Bathy	2025 AHN	2024 Bathy	2024 Kavel 10	RWS	Usable
Afferden en Deest	✓	✓	✓	✓		✓
Avelingen	✓		✓	✓		✓
Avelingerdiep	✓	✓	✓	✓	✓	✓
Baarsemwaard				✓	✓	
Bakenhof		✓	✓	✓		✓
Beneden Leeuwen	✓	✓	✓	✓	✓	✓
Blauwe Kamer	✓	✓	✓	✓		✓
Bossenwaard	✓	✓	✓	✓		✓
Brakel Oost	✓	✓	✓	✓	✓	✓
Brakel West	✓		✓	✓	✓	
Dordtsche Avelingen	✓	✓	✓		✓	✓
Elster Buitenwaarden		✓		✓		✓
Everdingenwaard	✓	✓	✓	✓		✓
Ewijkse Waard			✓	✓		
Gamerense Waard			✓	✓		
Goilberdingen						
Graafsche Waard	✓	✓	✓	✓	✓	✓
Groesplaat						
Heeselt Oost	✓	✓	✓	✓		✓
Heeselt West			✓	✓		
Hondswaard				✓		
Hurwenen			✓	✓	✓	
Kleine Willemsplaat				✓	✓	
Klompenwaard	✓	✓	✓	✓	✓	✓
Kwelgeul				✓		
Langsdam Ophemert	✓		✓	✓		
Langsdam Wamel-Dreumel		✓	✓	✓		✓
Lexmond	✓	✓	✓	✓		✓
Loenense Buitenpolder			✓	✓	✓	
Lunenburgerwaard						
Meinerswijk (Groene Rivier)	✓	✓	✓			✓
Meinerswijk (Plas van Bruil)			✓	✓		
Millingerwaard				✓	✓	
Noordoever Lek			✓			
Opijnen			✓	✓		
Palmerswaard	✓	✓	✓	✓	✓	✓
Pannerden	✓	✓	✓	✓	✓	✓
Passewaaij		✓			✓	
Pontwaard	✓	✓	✓	✓	✓	✓
Salmsteke		✓	✓	✓		✓
Schoutenwaard	✓	✓	✓	✓	✓	✓
Spiegelwaal	✓	✓	✓	✓	✓	✓

Stadswaard	✓	✓	✓	✓	✓	✓
Steenwaard						
Varikse Plaat				✓		
Vianen		✓	✓	✓		✓
Vogelzang			✓	✓		
Waalse Waard	✓	✓	✓	✓	✓	✓
Willige Langerak	✓	✓	✓	✓		✓
Total	**24**	**26**	**36**	**41**	**20**	**26**

D Planning and Process

D.1 Task Division

Table 15 shows how the tasks were (roughly) divided over the course of the project. While all members had their roles and responsibilities, it is important to note that we all participated in developing solutions for this project.

Table 15
Task division

Name	**Report Responsibilities**	**Task(s) performed**
Yair Roorda	2.1, 2.6 - 2.9 4.2, 4.4.1, 4.4.2 5.2.1, 5.2.2, 5.2.4, 5.3 6.1 7.2 A D.2	Quality control duties (setting up report, review writing and figures) Literature study, compiling feature list, implement features (area, perimeter, vegetation, draft of length and relative length, manual features), feature statistical analysis
Luc Jonker	2.3 - 2.5 3 4.1, 4.2.2, 4.3.1, 4.3.2.1, 4.4 5.1.2, 5.1.3, 5.2.3, 5.2.4 7.3, 7.4, 7.5 8 D .3	Technical manager duties (Setup architecture, maintain repo) Literature study, feature effort impact analysis, setup feature template, implement features (bed_level, slope, roughness, aspect, minimum flow threshold, manual features), assemble aggregated FME workspace
Vincent Vanderheeren	4.3.2, 4.4.3 - 4.4.5 6.2 C D.4	Chairman duties (agendas for meetings, shared calendar) Creating visualizations, set up spatial database, implement FME to database pipeline, machine learning prototype and analysis
Michel Beeren	4.3.4.2, 4.3.4.8 - 4.3.4.10, 4.3.4.18, 4.3.4.19 5.1.4 7.5.1	Communications and data management duties (emailing with clients, experts, supervisors, maintaining onedrive with data.) Initial QGIS feature extraction + analysis, implement features (relative length, length, width, distance from inlet, distance to bank, flow direction)

D.2 Relevant Courses

Table 16
Connection between Geomatics courses and project topics

Topic / Course	**Relevance & Application**
Sensing Technologies	• Proposing ways to measure unknown features. • Understand benefits and limitations of current data collection methods (e.g., Sonar, satellite imagery, Lidar).
GIS and Cartography	• Calculate slopes and aspect. • Use QGIS/FME for feature calculation.
Python Programming for Geomatics	• Writing Python code for data processing and automation.
Geo Database Management Systems	• Storing and accessing features in a geospatial database.
Digital Terrain Modelling	• Aspect and slope theory • Interpolation
Positioning and Location Awareness	• Managing coordinate reference systems (CRS). • Utilizing GNSS for precise positioning.
Geo-information Governance	• Integrating into corporate data governance (e.g., Van Oord). • Combining and processing data from many sources. • Designing storage and access systems.
Machine Learning for the Built Environment	• Feature engineering for predictive models. • Developing a predictive model prototype. • Using PCA for feature evaluation and reduction.
Applied Spatial Analytics	• Spatial unit design. • Feature discovery from spatial data. • Advanced cartography and map making.
Geoweb Technology	• WMS / APIs. • Interacting with the VoxView webservice.
3D Modelling and the Built Environment	• Medial Axis Transform for centerline calculations.

D.3 MoSCoW Analysis

Table 17
MoSCoW analysis per project phase. Objectives that are finished are bold.

	Discover	Process	Prototype	Reflect	Present
Must	• **Identify relevant features from literature**	• **Calculate relevant features for a single channel** • **Store the calculated features in a format that integrates into VanOord's data governance**	• **Use the calculated features for data analysis**	• **Evaluate the relative importance of different features**	• **Produce a clear and relevant report and presentation** • **Provide up-to-date documentation and metadata**
Should	• **Come up with additional features with the project team and VanOord** • **Rank features based on availability and relevance**	• **Scale the calculations to multiple channels** • **Automate the feature creation process** • **Quantify relationships and variance of features** • **Create a geodatabase for storing and querying the features**	• **Use the calculated features to train a predictive machine learning model** • **Use the calculated features to categorize different sites using machine learning**	• **Recommend surveying methods for additional unknown features** • ~~Compare computed features with features in literature~~	• **Produce an executive summary of the report and presentation** • **Write a technical report explaining how to extend our methodology in the future**
Could	• **Come up with additional features by reaching out to experts in the field**	• **Calculate features for all 52 sites, both 2024 and 2025** • Run basic CFD simulations for a case study	• Predict 2026 timestep using machine learning	• **Run statistical analysis to quantify relationships and significance** • **Interview end users about their experience**	• Produce a scientific poster presenting the main takeaways of the project
Won't	• Capture additional data	• Run advanced CFD simulations for multiple sites	• Use deep learning for predictive modeling	• Use detailed hydrological engineering knowledge to judge prototypes	• Develop a web service or app • Deliver a GUI or production-ready product

D.4 Rich Picture

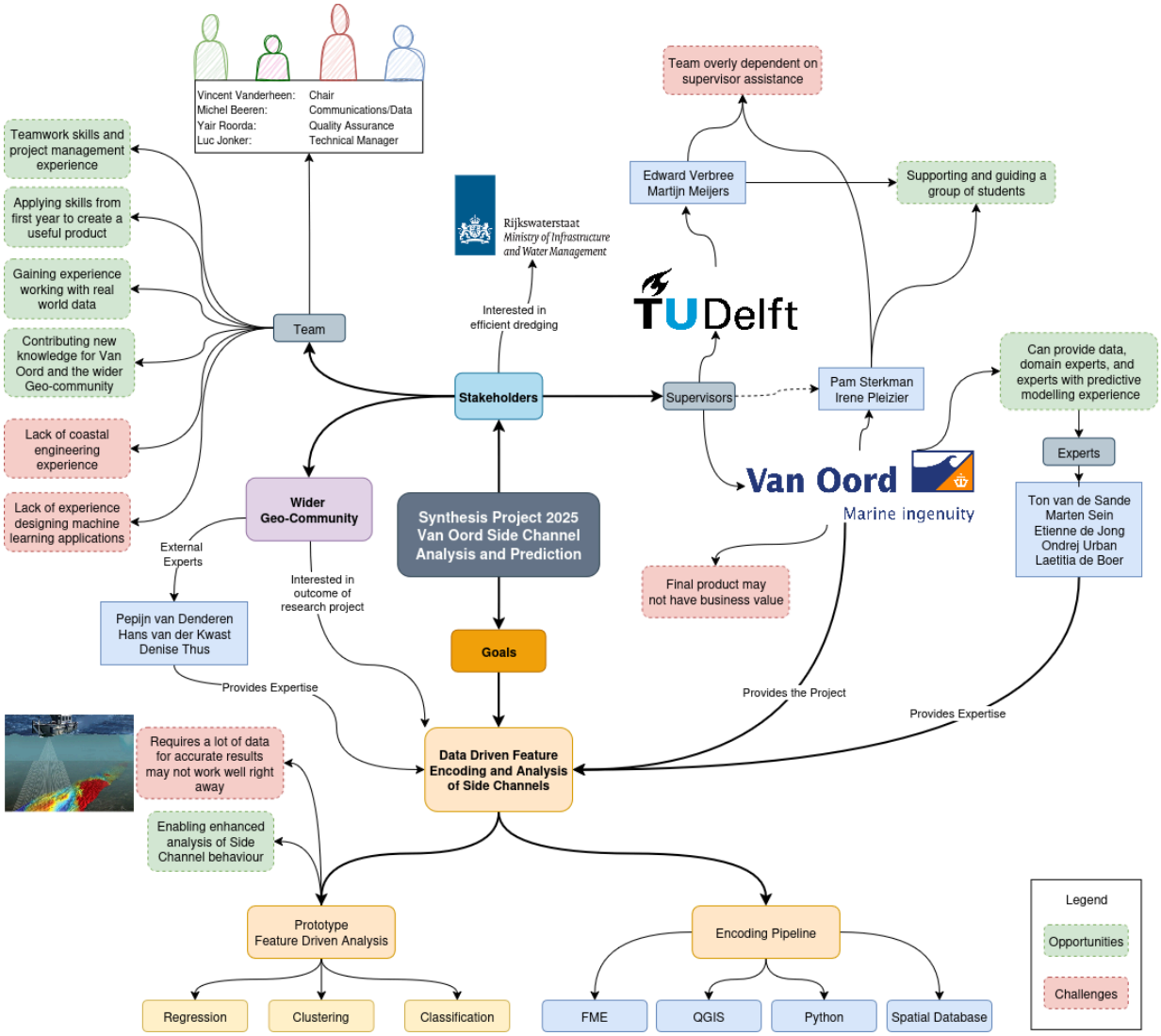


Figure 69. Rich picture describing project (own figure)

D.5 Gantt Chart

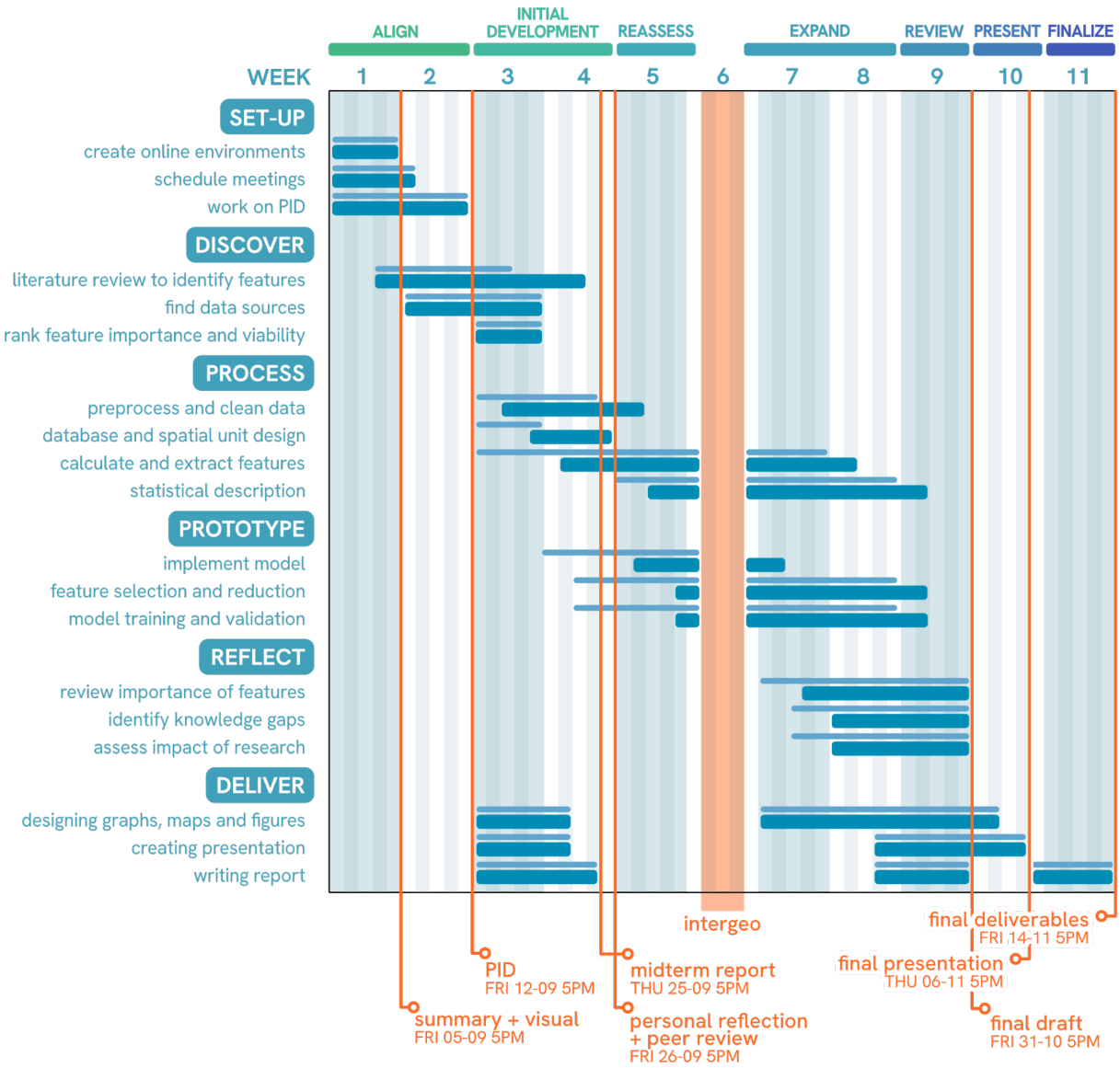


Figure 70. Gantt Chart (own figure)