






Sewer sediment deposition prediction using a two-stage machine learning solution

Marc Ribalta Gené ^{a,b,*}, Ramón Béjar ^b, Carles Mateu ^b, Lluís Corominas ^{c,d}, Oscar Esbríe
and Edgar Rubión ^a

^a Eurecat, Technology Centre of Catalonia, Unit of Applied Artificial Intelligence, Bilbao 72, Barcelona 08005, Spain

^b University of Lleida, Jaume II, 69, Lleida 25001, Spain

^c Catalan Institute for Water Research (ICRA-CERCA), Emili Grahit 101, Girona 17002, Spain

^d University of Girona, Plaça de Sant Domènec, 3, Girona 17004, Spain

^e Barcelona Cicle de l'Aigua, 08038 Barcelona, Spain

*Corresponding author. E-mail: marc.ribalta@eurecat.org

 MRG, 0000-0003-3082-9525; RB, 0000-0002-5208-685X; CM, 0000-0002-4864-0328; LC, 0000-0002-5050-2389; ER, 0000-0003-3886-3170

ABSTRACT

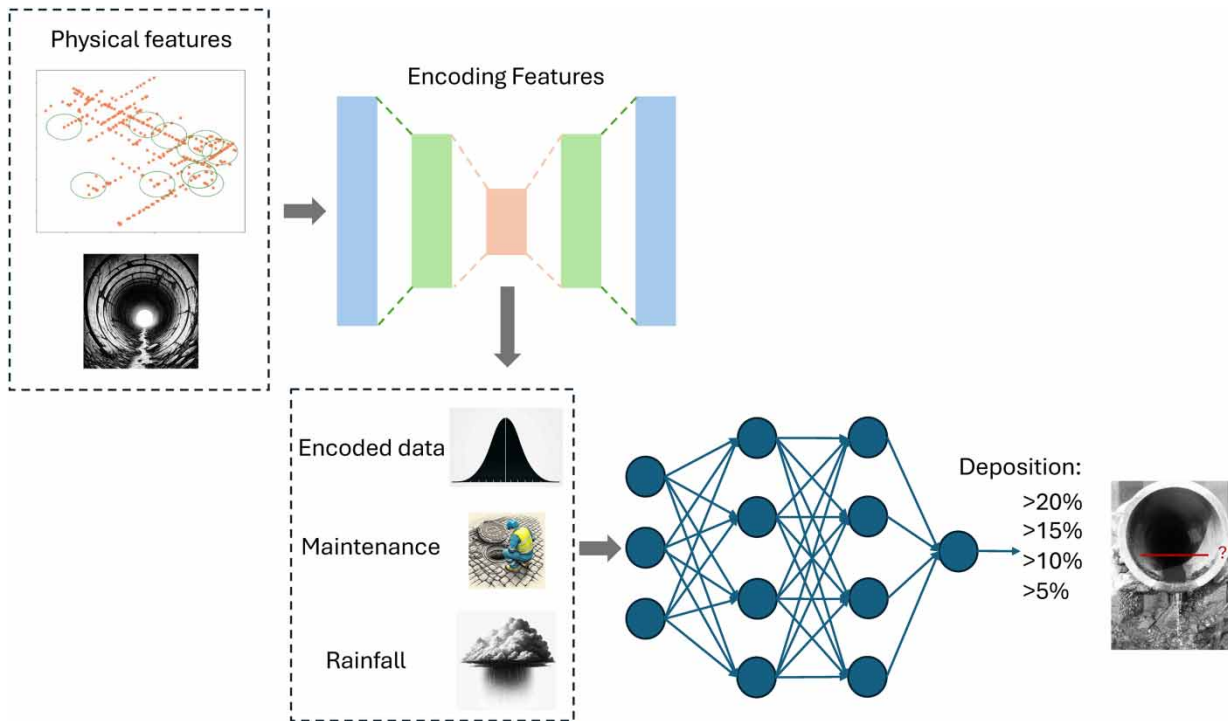
Sediment accumulation in the sewer is a source of cascading problems if left unattended and untreated, causing pipe failures, blockages, flooding, or odour problems. Good maintenance scheduling reduces dangerous incidents, but it also has financial and human costs. In this paper, we propose a predictive model to support the management of maintenance routines and reduce cost expenditure. The solution is based on an architecture composed of an autoencoder and a feedforward neural network that classifies the future sediment deposition. The autoencoder serves as a feature reduction component that receives the physical properties of a sewer section and reduces them into a smaller number of variables, which compress the most important information, reducing data uncertainty. Afterwards, the feedforward neural network receives this compressed information together with rain and maintenance data, using all of them to classify the sediment deposition in four thresholds: more than 5, 10, 15, and 20% sediment deposition. We use the architecture to train four different classification models, with the best score from the 5% threshold, being 82% accuracy, 70% precision, 76% specificity, and 88% sensitivity. By combining the classifications obtained with the four models, the solution delivers a final indicator that categorizes the deposited sediment into clearly defined ranges.

Key words: autoencoder, machine learning, neural network, sewer management, sewer sediment deposition

HIGHLIGHTS

- A predictive model based on an autoencoder and a feedforward neural network is proposed to predict future sediment deposition.
- The model's best score achieves an 82% accuracy rate for the 5% sediment deposition threshold, ensuring the identification of sediment deposition in the sewer system.
- The proposed model provides an easier-to-interpret indicator for water utilities to adapt their process accordingly.

GRAPHICAL ABSTRACT



1. INTRODUCTION

Combined sewer systems (CSSs) merge rainwater runoff, domestic sewage, and industrial wastewater in the same pipeline. Most systems transmit the combined sewage to wastewater treatment plants for treatment prior to its discharge into a water body. During the transport of wastewater, arid and solid sediment, fat, oil, and grease accumulate in some parts of the network. This event reduces the volume of water that can be carried through the pipe, and sometimes the build-up can be so large that failures or blockages can occur (Ashley *et al.* 2000, 2005). These events pose a high environmental and social risk, as they can cause retentions or flooding in the city with water containing toxic materials and substances. In addition, the sulphide derived from the retention of the wastewater will generate odour problems affecting the population and accelerate the sewage system's degradation and corrosion. Preventive routines involve a significant investment in terms of financial resources and human labour, but they are essential for maintaining the proper flow of wastewater through the sewer system by continuously monitoring its status and removing sediments as needed. For instance, the cost of maintaining and expanding the piping network in the city of Munich from 2000 to 2021 is estimated at €192.3 million (Al-Azzawi *et al.* 2022), and the US Environmental Protection Agency (EPA) estimated in 2016 that \$271 billion were needed to maintain and improve the national wastewater infrastructure. A portion of this investment, approximately \$48 billion, is allocated for correcting Combined Sewer Overflow (CSO) issues, which occur during wet weather conditions and result in the discharge of untreated wastewater and stormwater (Daguillard 2016).

Nowadays, water organizations extract and save the information generated during maintenance routines, resulting in a valuable pool of exploitable data. This information can be used to train data-driven models and develop predictive maintenance (PdM) solutions. The concept behind PdM is to use advanced analytical tools to anticipate when maintenance or repairs are required. Organizations can analyse patterns and trends in the data obtained during historical maintenance routines to predict when the infrastructure is likely to fail or require maintenance. This technique enables cost savings by allowing for the postponement or reduction of certain routines while also improving the allocation of resources dedicated to specific tasks (Kumar *et al.* 2019).

A diverse range of machine learning (ML) algorithms can be used to implement PdM solutions (Carvalho *et al.* 2019; Ran *et al.* 2019; Çınar *et al.* 2020). Determining the state of objects or infrastructures in the short or long term can be achieved

using supervised or unsupervised classification algorithms. Similarly, supervised regressive algorithms can be used to predict the remaining useful life or degradation of an object or infrastructure over time.

Studies show several ML techniques applied for sewer system condition prediction, where some of the main focuses are blockage, failure, and sediment predictions. [Bailey et al. \(2015\)](#) show the use of static and dynamic sewer data to create a predictive model based on decision trees that calculates the blockage likelihood of a sewer section. In a later study ([Bailey et al. 2016](#)), they show the addition of geographical grouping of sewer sections to improve their research. [Hassouna et al. \(2019\)](#) introduce the use of maintenance duration and sewer level monitoring data to train ensemble algorithms like a random forest to predict blockage incidents, and [Okwori et al. \(2021\)](#) explain the use of spatial heterogeneity to predict blockage propensity in sewer sections using a Poisson regression, and the prediction of blockage recurrence using random forest.

In the use case of failure prediction, the data and algorithms used to generate the predictive models are similar. [Tavakoli et al. \(2020\)](#) generate a random forest using data from the mainline sewer pipes, following a methodology like the ones already mentioned. Other types of algorithms are also being researched nowadays. [Sattar et al. \(2019\)](#) compare artificial neural networks (ANNs), a support vector machine (SVM), and a non-linear regression using a dataset containing 80 years of recorded failures within a sewer network. Genetic algorithms are also used in the domain to optimize the hyperparameters of predictive models during the training process ([Robles-Velasco et al. 2021](#)). Even in studies where the use cases are based on predicting suspended sediment load on rivers, the range of algorithms tested varies from traditional ML such as Random Forest (RF) or SVM, to deep learning algorithms such as ANN and specific architectures such as long short-term memory ([Allawi et al. 2023a, 2023b](#)).

Sediment problems are usually solved by focusing on sediment transport amount or deposition level as an objective variable. [Song et al. \(2018\)](#) build a predictive model based on numerical analysis using a two-phase calculation of fluid flow analysis and sediment accumulation, using data of non-dimensional variables for inlet velocity, inlet particle volume fraction, and particle size. [Mohtar et al. \(2018\)](#) compare ANN and radial basis function models to obtain the best predictions over the incipient sediment motion in sewers using the parameters: varying sediment thickness, median grain size, and water depth, demonstrating the potential of using ANN as it is the best model in their study. The potential of ANN is also demonstrated for sediment deposition in CSSs when compared to more traditional ML algorithms ([Ribalta et al. 2021](#)). Finally, feature reduction is also a relevant topic within sewer predictions. [Bhagat et al. \(2021\)](#) propose a solution to predict sediment lead using a two-phase solution that involves a first step of variable selection using extreme gradient boosting and a second step using another ML algorithm such as ANN or SVM.

Our study stems from previous work ([Ribalta et al. 2021](#)), where we compare predictive models to classify the sediment deposition percentage level in the present and ten days ahead, and another approach that compares models to classify if a sewer section needs maintenance. The novelty in this previous study is the use of the information provided by similar neighbour sections to improve the data being fed into the algorithms and the produced predictions. However, the study has a problem with predicting sediment deposition. The obtained result pertains to present deposition within a section, which may not be as useful for water utilities since they need to schedule maintenance days in advance, and the prediction assumes that the maintenance needs to be done at that moment. Another problem arises from the data of nearby sections since utilities must access these sections to extract current deposition data, and in many cases, maintenance has already been applied to the section to be predicted. Predicting future sediment deposition increases planning time and opportunities for utilities.

In this new work, our goal is to enhance the accuracy of predicting future sediment deposition and approximate the amount that will occur before the next maintenance, which could take place in the next 3–12 months. However, this research poses a significant challenge due to the dataset's complexity, which comprises only a limited number of historical records but a high number of features – 85 features spread across 1905 records. Consequently, the dataset's structure creates uncertainty for the model, making it arduous to establish correlations between the features when there are few records available for training.

We aim to overcome this challenge by proposing a novel ML-based approach that employs a two-phase solution with a categorical objective variable, unlike the traditional continuous variable. The first model we present compresses information from 49 physical variables into 15 output variables, while the second model utilizes this output information together with 36 dynamic variables to classify sediment deposition status. To validate our models, we use data obtained from historical records of three neighbourhoods in the CSS of Barcelona.

Our proposed data model leverages not only the data from individual sewer pipes but also features from pipes that exhibit similar deposition behaviour. With this approach, we are confident that our models will deliver relevant scores in predicting

future sediment deposition while using fewer features, thereby eliminating information overload and providing a more efficient and practical solution.

2. MATERIALS

2.1. Dataset

The dataset we analysed provides network information on three neighbourhoods in the City of Barcelona, covering the period between 2017 and 2021. Table 1 presents the historical dataset's variables, including physical features, water and climate behaviour indicators, and maintenance registers for each pipe, bed, sections, or other types of structures (we use the word pipe or section to address them within the study), also known as dynamical features in this study. Notably, the dataset comprises diverse records from various sewer sections, ranging from small diameter pipes of 0.15 m to massive ones exceeding 2 m in diameter. The pipe length is also highly variable, with crosscutting pipes measuring a mere 0.5 m and main waterways extending up to 74 m. Maintenance routines are not performed on a scheduled basis but are instead planned based on urban management plans, sewer system risk prevention, and other factors, causing inspection dates to fluctuate. However, one of the most critical aspects of the dataset is the interval between maintenance routines, which can range from 2 months to over a year. This variation amplifies the uncertainty of determining the deposition level's evolution since the last inspection. Rainfall also has a significant double impact on sediment deposition, increasing the wastewater flow and, as a result, the risk of blockage or natural flushing. Therefore, the daily rainfall data collected by three rain gauges, one for each neighbourhood, provides a crucial indicator of trend changes in the deposition level, making it an essential component of our dataset.

2.2. Data model

Our data model augments each record, as presented in the previous section, appending two types of information: the state of the pipe at previous maintenance sessions, and physical and dynamic features from sections near the pipe. The result of analysing and processing the data can be seen in Table 2. The physical features are the same as described in Table 1, and most dynamic features are modified. The biggest change between the dataset and the data model is the use of data from near and similar sections. We incorporate data from similar sections to improve accuracy, selecting the five sections with the most similar sediment deposition trends and including their features in the data model. The similar (near) sections are all in the same neighbourhood, so the feature indicating it is only present for the section to be evaluated. Additionally, we enhance sediment level and cleaning features by including past values to understand better the accumulation trends. Using the sediment level in the historical data, expressed in centimetres, and comparing it to the height of the section, we compute the proportion of the section that the accumulated sediment occupies, represented as a percentage. We find that discarding the days between

Table 1 | Historical dataset of registers

Variable	Description	Unit	Range
Section	The cross-section of the pipe	dm ²	1.7–806
Height	The maximum height of the pipe zone	m	0.15–2.5
Width	The maximum width of the pipe zone	m	0.15–4.5
Perimeter	The perimeter of the pipe	M	4.7–12.4
Length	The total length of the pipe	M	0.5–74
Velocity	The mean velocity of the water during dry seasons	m/s	0–0.47
Water level	The mean water level during dry seasons	M	0–1.12
Flow rate	The mean flow rate during dry seasons	m ³ /s	0–0.04
Daily rainfall	Amount of accumulated daily rainfall	mm	0–83
Days between maintenances	The number of days between inspections	days	50–400
Cleaning applied	A Boolean to indicate if a cleaning was done during an inspection	–	0 or 1
Sediment level	The maximum deposited sediment level within a pipe	Cm	0–60
Neighbourhood	An indicator of the neighbourhood of the pipe	–	1, 2, or 3

Note: Variables for each pipe.

Table 2 | Data model used to generate the data-driven solution

Type of feature	Feature name	Details
Physical	Section	Same as in Table 1
	Height	Same as in Table 1
	Width	Same as in Table 1
	Perimeter	Same as in Table 1
	Length	Same as in Table 1
	Velocity	Same as in Table 1
	Water level	Same as in Table 1
	Flow rate	Same as in Table 1
	Neighbourhood	Only present for the section to be evaluated
Dynamic	Rainfall average	Same as in Table 1
	Rainfall standard deviation	Same as in Table 1
	Cleaning applied	Two features indicating if a cleaning was done during the last two maintenance sessions
	Sediment level	Two features of the last two deposited levels extracted and a third one is used as an objective variable
Physical	Physical features of nearer sections	The physical features for all the 5 near sections
Dynamic	Dynamic features of nearer sections	The dynamic features for all the 5 near sections

Note: Divided into the type of feature, the name of the feature used, and an explanation of how it is obtained from the historical data.

maintenance routines leads to better results, so we remove this feature. Lastly, we aggregate daily rainfall data between maintenance routines into two variables: the average rainfall and the standard deviation. The former provides an overview of the rain amount during the period, while the latter considers all days with intense rainfall. Therefore, the days between maintenance routines are also represented in the rainfall impact on the sewer section, and the sediment accumulation trends for that same section let the model assume the possible increment between maintenance sessions. Overall, the total number of registers amounts to 1905.

3. METHODOLOGY

3.1. Structure of the predictive solution

The available data are divided into two types of features: constant values of a pipe called ‘physical’ and variable historical values called ‘dynamic’. However, with a final count of 85 features, predicting sediment accumulation using these features can be challenging. An ML algorithm requires many records to understand the existing variables. Thus, increasing the number of variables in the dataset necessitates an increase in records (Sorzano *et al.* 2014). Fortunately, the physical features, composed of constant values, have lower overall variance within the dataset and are more redundant than the dynamic features. Therefore, we propose a methodology to pre-process the physical features using an autoencoder (AE) architecture to compress them to a smaller number of features without significant information loss. We then join these compressed physical features with the dynamic features to predict sediment accumulation using an ANN.

An AE (Kramer 1993) is a type of ANN used to learn the relation between the different features of a dataset. It consists of an encoder, which compresses the information into a lower space, and a decoder, which tries to decompress the information into its original state. It has diverse applications, such as image denoising, image reconstruction, or dimensionality reduction (Kramer 1993). AEs shine in unsupervised learning scenarios, leveraging the inherent structure of the data to learn meaningful representations without the need for labelled datasets. By encoding data into a compressed latent space and then decoding it back to its original form, they facilitate the discovery of fundamental data characteristics, creating a more concise and meaningful feature representation. This transformation is particularly beneficial for complex datasets, where the reduction of dimensionality can significantly improve the performance of subsequent predictive models. Although other methods of dimension reduction have not been tested in this study, the authors have explored other options and decided on AEs due to their theoretical and practical advantages. For instance, unlike linear methods such as principal component analysis, AEs can capture non-linear relationships in the data, which is crucial for complex datasets where linear assumptions do

not hold (Wang *et al.* 2016). AEs are capable of learning features automatically from the data, eliminating the need for manual feature engineering – a significant advantage in high-dimensional spaces where manual feature selection is impractical. Additionally, one of the main ideas is to study an ML solution that can be reused or re-implemented with ease in other case studies. Other dimension reduction methods require meticulous analysis and expert knowledge to decide the number of features to reduce, but to decide the number of features being reduced by an AE, the methodology is based on the scores and performance of the predictive model. In this study, we apply the architecture to compress the information of the physical features during the encoding phase and the result is sent to the ANN predictive model, specifically a feedforward neural network with a multi-layer perceptron. The decision to use the ANN comes mainly from the conclusions drawn in the previous study, where it is shown that the feedforward neural network trains better with our data to make future predictions (Ribalta *et al.* 2021). Nevertheless, ANNs are versatile and powerful models capable of capturing complex non-linear relationships in data, making them suitable for prediction tasks where the relationship between input features and the target variable is not linear or easily discernible.

The predictive model uses the 36 dynamic features shown in Table 2 and the compressed physical features to identify the future percentage of sediment level accumulated within a pipe. Concretely, the aim is to classify the percentage of sediment accumulation occupying the pipe, divided into five different classes: 0–5%, 5–10%, 10–15%, 15–20%, and 20–100% deposition level. The number of registers indicating an accumulation of more than 20% comprises 3% of the total number of registers available within the dataset, and since any value higher than 20% is dangerous for the water utility, we decide to classify them using the same class. The classification of these different classes, each made by a different model, can be used together to obtain a more specific indicator of the specific range in which the deposited sediment can be found. For instance, when one model indicates that sediment deposition exceeds 5% (True) and another model suggests that it does not surpass 10% (False), we can deduce that the sediment deposition falls within the 5–10% range.

The prediction of sediment deposition is usually done with regression algorithms, but in our case, a classification problem offers several advantages. First, it facilitates handling the dataset's sparsity and the high number of features by focusing on categorical outcomes rather than exact numerical predictions, which might be highly variable or uncertain due to the dataset's constraints. Second, a classification model provides actionable insights into water utilities by categorizing sediment deposition levels into predefined classes that directly inform maintenance scheduling decisions, enhancing operational efficiency and planning.

3.1.1. Generation process

To generate the two-step ML solution, we need to train and evaluate the architectures in a specific order. Table 3 shows the different steps carried out during the study to train the ML architecture. We initially define, train, and evaluate several AE architectures to compare and decide which are suitable for the data model. The decision is based on the performance of a single perceptron, which receives the output results of the AE and uses them to predict sediment deposition. Afterwards,

Table 3 | Solution generation process

AE definition
Draft AE definition
AE architectures training and evaluation
AE architectures selection
ANN definition
Draft ANN definition
ANN architectures training and evaluation
ANN architectures selection
Final model training and deployment
Select the combinations of best AE and ANN
Combination of training and evaluation
Select the best combination

the same methodology is applied to define the ANN architecture. Those are trained using the data model without applying any encoding conversion, so the results are not biased by any AE pre-processing. The final step is to build different combinations from the best AE and ANN architectures, train and evaluate the predictions, and select the most scoring one.

Neural networks have multiple parameters to configure, and the number of possible combinations is high. Training all possible neural network architectures would take a lot of time and resources, so the best workaround is to previously filter certain configurations and train only the ones with the best expectations to perform. In the article, we show a subset of architectures that obtain better performance and indicate patterns between architectures, such as the best number of layers or the range of neurons with better scoring.

A modelling stage is a process of training an architecture or algorithm to produce a predictive model. In each modelling stage, we apply the train-test process depicted in Figure 1 to ensure that the evaluation of the model's predictions is fair and unbiased. By using a portion of the data that the model has not been trained on (the test set), we can assess how well the model generalizes to new, unseen data. This helps to avoid overfitting, where a model performs exceptionally well on its training data but poorly on any new data, and ensures that the performance metrics reflect the model's true predictive capability. The dataset is divided into 5 different portions using a sliding window, with 80% of the data used for training the model and 20% used to evaluate the model's predictions. Random splitting is avoided to prevent quasi-leakage caused by records that are temporally close to those in the training set from appearing in the test set. These splits are 80–20%, 60–20–20%, 40–20–40%, 20–20–60%, and 20–80% (with the italicized percentage indicating the test dataset and the other the training dataset). After training the model, we assess its predictions using the test set, which contains data from sections that the model has not seen before, to determine its ability to adapt and generalize. Finally, we put together all the predictions made in a single set and extract the scoring values to evaluate the model's performance, defined in the next section.

3.1.2. Evaluation metrics

To determine the most effective predictive architectures, we utilize various metrics that allow us to compare trained models, identify predictive weaknesses, and evaluate overall algorithmic performance. We employ six distinct metrics to evaluate each model, as outlined in Table 4, with each metric providing a unique perspective on the performance of the algorithm:

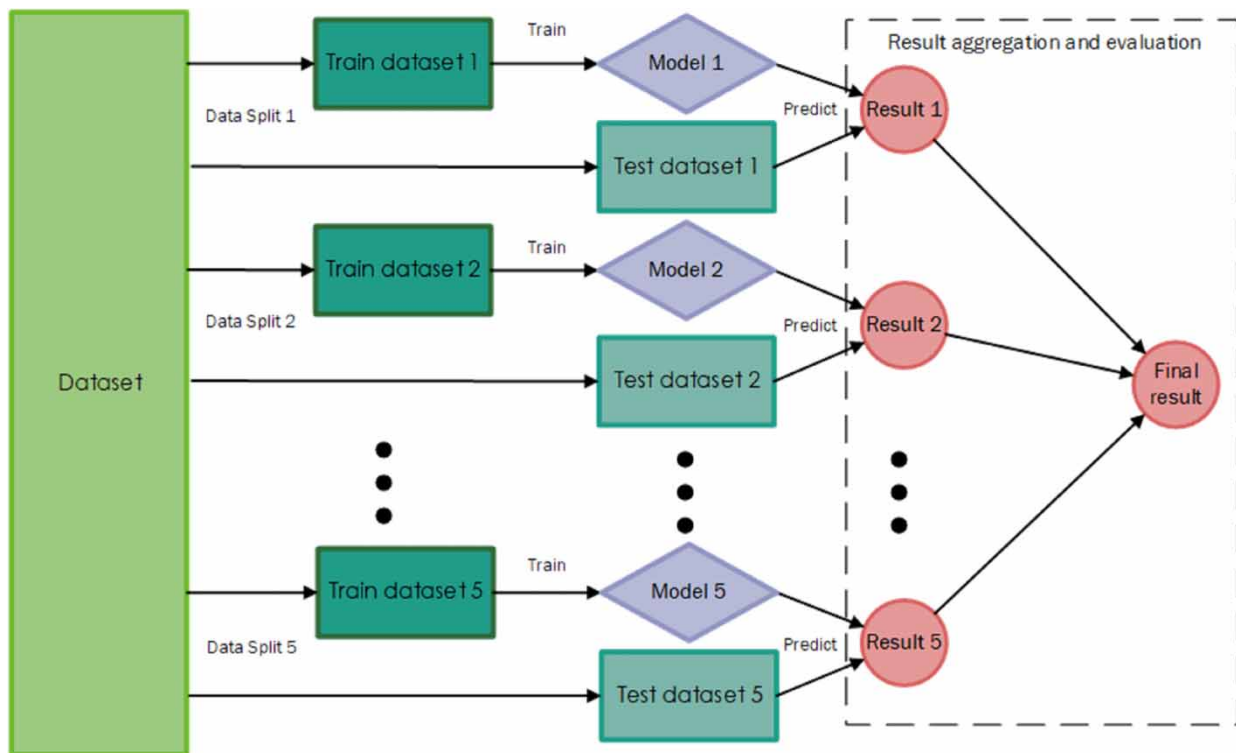


Figure 1 | Train, test, and result evaluation process.

Table 4 | Evaluation metrics

Statistical metrics	Formulas	Optimum score
Accuracy	$\frac{\text{True positives} + \text{True negatives}}{\text{Positives} + \text{Negatives}}$	1
Specificity	$\frac{\text{True negatives}}{\text{True negatives} + \text{False positives}}$	1
Sensitivity	$\frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$	1
Precision	$\frac{\text{True positives}}{\text{True positives} + \text{False positives}}$	1
AUC	The area of the ROC curve.	1
ASS	$\frac{\text{Specificity} + \text{Sensitivity}}{2}$	1

Note: A positive is a register with more deposition than the threshold and a negative is a register with less deposition than the threshold.

- Accuracy: Measures the proportion of correct classifications of sediment deposition among the total dataset.
- Specificity: Measures the proportion of real sediment deposition with less value than the threshold classified correctly by the model.
- Sensitivity: Measures the proportion of real sediment deposition with more value than the threshold classified correctly by the model.
- Precision: Measures the proportion of predicted sediment deposition with more value than the threshold classified correctly by the model.
- Area under the curve (AUC): The receiver operating characteristic (ROC) curve plots the true positive (TP) rate and the false positive (FP) rate at different classification thresholds, producing different bidimensional points ranging from (0, 0) to (1, 1) with the shape of a curve. The AUC is the entire area underneath the ROC curve, and it provides an aggregated performance of how well the model performs at various threshold settings (Bradley 1997).
- Average specificity-sensitivity (ASS): Measures the balance between the two metrics from a general perspective.

4. RESULTS

4.1. Autoencoder evaluation

The evaluation of the drafted AE architectures is shown in Table 5. The first row details the results of the predictive algorithm without using an AE to show the benefits of using the AE in a basic use case. During the first 50 epochs of training the neuron, not using an AE provides the best sensitivity out of all options. The ASS is 0.74, being the best value in epoch 50 but not the best of the three evaluated epochs, being surpassed by other architectures.

Regarding the number of layers used, the maximum specificity obtained is 0.71 when using two layers to encode the data, 0.72 when using three encoding layers, and the maximum sensitivity is 0.83 and 0.84, respectively. Table 5 displays combinations with different activation functions, showing the performance of the architectures using a relu or sigmoid activation. Overall, both perform well, but the highest performance is accomplished using the sigmoid activation function, with a maximum specificity of 0.71 and a sensitivity of 0.84. The architecture with the best specificity has 30-20-10-20-30 neurons per layer and sigmoid activation function, with 0.72 specificities at epoch 50, and the best sensitivity is offered by three different architectures scoring 0.84 at epoch 100.

To decide which architecture performs better overall, the ASS is used, selecting the highest. The results indicate two architectures with the same score: the 5-layer AE with 30-20-15-20-30 neurons per layer using the sigmoid activation function and the 5-layer AE with 40-30-10-30-40 neurons per layer using the relu function, both with an average score of 0.75.

4.2. ANN evaluation

Table 6 presents the best subset of ANN architectures, trained without using an AE and using different layer sizes and activation functions. The architectures have a common pattern of three layers each and a sigmoid as the final activation

Table 5 | AEs evaluation

Architecture ^a	Specificity			Sensitivity		
	Epoch 50	Epoch 100	Epoch 150	Epoch 50	Epoch 100	Epoch 150
No AE	0.66	0.70	0.69	0.83	0.78	0.79
25-10_relu	0.69	0.66	0.67	0.76	0.81	0.81
25-10_sigmoid	0.69	0.65	0.68	0.74	0.83	0.80
30-10_relu	0.69	0.68	0.69	0.78	0.81	0.81
30-10_sigmoid	0.71	0.68	0.69	0.77	0.81	0.79
30-20-10_relu	0.72	0.68	0.70	0.75	0.81	0.79
30-20-10_sigmoid	0.70	0.66	0.67	0.76	0.81	0.80
30-20-15_relu	0.68	0.66	0.69	0.77	0.82	0.78
30-20-15_sigmoid	0.68	0.66	0.69	0.78	0.84	0.81
35-20-10_relu	0.70	0.67	0.68	0.76	0.80	0.79
35-20-10_sigmoid	0.70	0.66	0.67	0.76	0.82	0.81
40-20-10_relu	0.70	0.68	0.69	0.78	0.81	0.81
40-20-10_sigmoid	0.70	0.65	0.68	0.77	0.84	0.81
40-20-15_relu	0.70	0.67	0.70	0.74	0.81	0.76
40-20-15_sigmoid	0.68	0.66	0.68	0.78	0.83	0.81
40-30-10_relu	0.71	0.68	0.69	0.77	0.82	0.81
40-30-10_sigmoid	0.70	0.65	0.67	0.77	0.83	0.80
40-30-15_relu	0.69	0.67	0.70	0.76	0.81	0.77
40-30-15_sigmoid	0.67	0.65	0.67	0.79	0.84	0.80

^aAll shown options are trained using a mean squared error loss. Other loss options such as the mean absolute error or Huber score are significantly worse. Only the encoder layers of the AE are shown.

Table 6 | Best ANN architectures without AE

Architecture	Specificity			Sensitivity		
	Epoch 50	Epoch 100	Epoch 150	Epoch 50	Epoch 100	Epoch 150
5-5-1_LogSigmoid-LogSigmoid-Sigmoid	0.70	0.68	0.69	0.82	0.80	0.81
5-5-1_Softplus-Softplus-Sigmoid	0.70	0.69	0.67	0.82	0.82	0.81
5-5-1_Softsign-Softsign-Sigmoid	0.73	0.77	0.74	0.83	0.79	0.78
10-10-1_LogSigmoid-LogSigmoid-Sigmoid	0.69	0.71	0.69	0.87	0.82	0.86
10-10-1_Softplus-Softplus-Sigmoid	0.72	0.70	0.71	0.85	0.85	0.83
10-10-1_Softsign-Softsign-Sigmoid	0.74	0.74	0.78	0.87	0.84	0.81
15-15-1_LogSigmoid-LogSigmoid-Sigmoid	0.73	0.70	0.71	0.84	0.85	0.83
15-15-1_Softplus-Softplus-Sigmoid	0.72	0.72	0.74	0.84	0.83	0.80
15-15-1_Softsign-Softsign-Sigmoid	0.74	0.76	0.74	0.87	0.82	0.83
20-20-1_LogSigmoid-LogSigmoid-Sigmoid	0.71	0.71	0.76	0.88	0.86	0.82
20-20-1_Softplus-Softplus-Sigmoid	0.76	0.73	0.72	0.83	0.84	0.84
20-20-1_Softsign-Softsign-Sigmoid	0.78	0.77	0.76	0.83	0.84	0.80

function. The best specificity is offered by the 20-20-1 and 10-10-1 neural networks, both using the activation function softsign, with a score of 0.78, and the best sensitivity is offered by the 20-20-1 neural network using a logarithmic sigmoid, 0.84.

Two architectures offer the best score, the 10-10-1 and 20-20-1 neural networks using a softsign activation function, both with an ASS of 0.8 at epoch 50. However, the 20-20-1 neural network has the same score at epoch 100, being the decisive factor to select it as the best neural network.

4.3. Final architecture evaluation

The final step is to determine which combination of AE and ANN offers the best results. Table 7 shows the scores of the top two AE combined with the best ANNs. The ANN is trained during 50 epochs, uses a softsign activation function, and uses a mean squared error loss, but each AE is trained using various settings to maximize the final score. Table 7(a) displays the results of training a 5 neurons per layer ANN, and the AE with a 40-30-10_ReLU architecture outputs a 0.8 ASS score, improving the ASS by 0.02 from training without an AE. Table 7(b) shows the results for a 10 neurons per layer ANN, but in the case of using an AE to boost the train outputs the same ASS, 0.8. Table 7(c) presents the results for a 15 neurons per layer ANN, and the 40-30-10_ReLU scores the best ASS, 0.82, which is a 0.02 increase from training the ANN without AE. Lastly, Table 7(d) displays the result of using an AE to boost the training of the 20 neurons per layer ANN. The best ASS is achieved by the 30-20-15_Sigmoid AE, increasing from 0.81 to 0.82.

We extend the analysis beyond the integration of AEs and ANNs by comparing their performance against traditional ML techniques, specifically focusing on how these methods fare when utilizing the outputs generated by the AE. The models selected for this comparative analysis were previously employed in an earlier study (Ribalta *et al.* 2021). However, due to the evolution of our data model, which now incorporates additional variables such as rainfall, a retraining process was necessary to ensure their current applicability. To identify the optimal configuration for each model, we conducted a grid search. This process involved experimenting with various parameter combinations to determine which settings yield the most favourable outcomes. The output of this process produced the following models and configurations:

- Gradient Boosting: 200 estimators, maximum depth of 3, and learning rate of 0.1.
- AdaBoost: 200 estimators and learning rate of 0.1.
- RF: 200 estimators, without restricting the maximum depth.
- Extra Trees: 200 estimators and maximum depth of 20.
- Logistic Regression: Inverse of regularization strength, C, of value 10.

To train and evaluate these models, we used the same methodology that has been used with the neural networks, and the results are shown in Table 8. The models have higher scores in specificity than the neural networks, but in the case of sensitivity and the ASS metrics, the scores are lower. Of the models shown, the one with the best scores is the Gradient Boosting, with a final ASS value of 0.74, being 0.08 lower than the best ANN configuration. Therefore, the ANN provides better results than any of the traditional algorithms presented.

Table 7 | Best AE combined with (a) 5 neurons architecture, (b) 10 neurons architecture, (c) 15 neurons architecture, and (d) 20 neurons architecture

AE architecture	Specificity	Sensitivity	ASS	AE epochs	Loss criteria
(a) ANN 5-5-1_Softsign-Softsign-Sigmoid					
40-30-10_ReLU	0.7	0.9	0.8	1,000	MSE
30-20-15_Sigmoid	0.74	0.85	0.79	500	Huber
(b) ANN 10-10-1_Softsign-Softsign-Sigmoid					
40-30-10_ReLU	0.77	0.84	0.80	1,000	MSE
30-20-15_Sigmoid	0.77	0.84	0.80	1,500	Huber
(c) ANN 15-15-1_Softsign-Softsign-Sigmoid					
40-30-10_ReLU	0.77	0.86	0.82	1,000	Huber
30-20-15_Sigmoid	0.77	0.85	0.81	1,500	MSE
(d) ANN 20-20-1_Softsign-Softsign-Sigmoid					
40-30-10_ReLU	0.78	0.84	0.81	250	MSE
30-20-15_Sigmoid	0.76	0.88	0.82	500	MSE

Table 8 | Traditional ML algorithms evaluation using the 30-20-15 AE architecture

Algorithm	Specificity	Sensitivity	ASS
Gradient Boosting	0.82	0.66	0.74
AdaBoost	0.83	0.58	0.70
RF	0.83	0.59	0.71
Extra Trees	0.83	0.59	0.71
Logistic Regression	0.82	0.54	0.68

The final architecture is displayed in [Figure 2](#) to enable other researchers or engineers to reproduce the model. The AE uses its first three layers to compress or encode the 49 physical features into 15 reduced features. This process of encoding effectively reduces the dimensionality of the data, capturing the most relevant information in fewer features. After the dimensionality reduction, the 15 encoded features are combined with 36 dynamic features and sent to the ANN as input. The ANN (a multi-layer perceptron) processes these features to predict an output value that ranges between 0 and 1, which is then converted into a Boolean value (true or false, or 1 or 0). This step is done by applying a threshold where outputs above 0.5 are considered true (1) and those below it false (0).

To ensure the validity of the proposed architecture across different classification thresholds, [Table 9](#) presents the evaluation of four independent models using various scoring metrics. An important factor to consider when assessing the results is the balance of positive values in the dataset, or, said differently, the percentage of registers that are higher than the evaluated threshold. The dataset with the lowest threshold (>5%) exhibits a balance of 39%, which decreases as the threshold increases, to 14, 4.82, and 3.25% for the thresholds >10%, >15%, and >20%, respectively. This imbalance significantly impacts on precision and sensitivity scores, as they tend to decrease in value due to the high penalty incurred by the low number of real positives to be predicted compared to the high likelihood of producing FPs. Conversely, accuracy and specificity benefit from the imbalance due to the increase in true negative (TN) values and the model's ability to correctly predict a negative value.

The results of the scoring metrics reveal a possibility for overfitting in each of the models. To ensure the robustness of the models, [Figure 3](#) presents four confusion matrices, which illustrate the prediction results of the models trained with the objective variable at the different thresholds, with the upper row depicting the 5 and 10% thresholds and the lower row representing the 15 and 20% thresholds. In all cases, the FP rates are lower than the TN rates, the false negative (FN) rates are lower than the TP rates, and the FN rates are lower than the TN rates. However, the FP rates are only lower than the TP rates at the 5% threshold, and the disparity between the two increases with each subsequent threshold increment.

In the final step of our analysis, we evaluate the performance of the models when used in conjunction to predict future sediment deposition through a simple decision-making process. Specifically, we merge the predictions of each model by assuming that the highest predicted deposition value is correct. For instance, if one model predicts a deposition of 5% and another predicts 20%, we select 20% as the final prediction. To evaluate the accuracy of this approach, we present a confusion matrix in [Figure 4](#), which compares the actual sediment deposition classifications with the predicted ones across five categories: no deposition, greater than 5%, greater than 10%, greater than 15%, and greater than 20%. By comparing the actual classifications with the model's predictions across these categories, the confusion matrix in [Figure 4](#) provides a comprehensive view of the model's performance. It helps identify not just overall accuracy but also specific types of errors – such as over- or under-estimating the amount of sediment deposition – which are crucial for refining the model and improving future predictions. The primary goal is to determine how well the predictive approach matches reality.

When examining the real sediment depositions that exceed 15 and 20%, it is interesting to note that our models never predict them as having no deposition, and instead, mostly predict a deposition of over 10%. The significance of this observation lies in the models' ability to distinguish between no deposition and significant deposition events. Similarly, for increases greater than 10%, only a handful of cases were predicted as having no deposition. As the actual deposition values decrease towards the threshold for no deposition, we see an increase in false predictions, with over 100 instances of FNs at the 5% threshold. Looking at FPs, we observe that nearly 400 records were wrongly predicted as having a sediment deposition, with many instances of >5% predicted as higher thresholds. Particularly, the >15% threshold suffered the most, with only 9 correct predictions. It is worth evaluating the extent of erroneous predictions. In the case of >20%, the worst false prediction would be predicting no deposition, and the best would be predicting >15%. False predictions are often adjacent to the

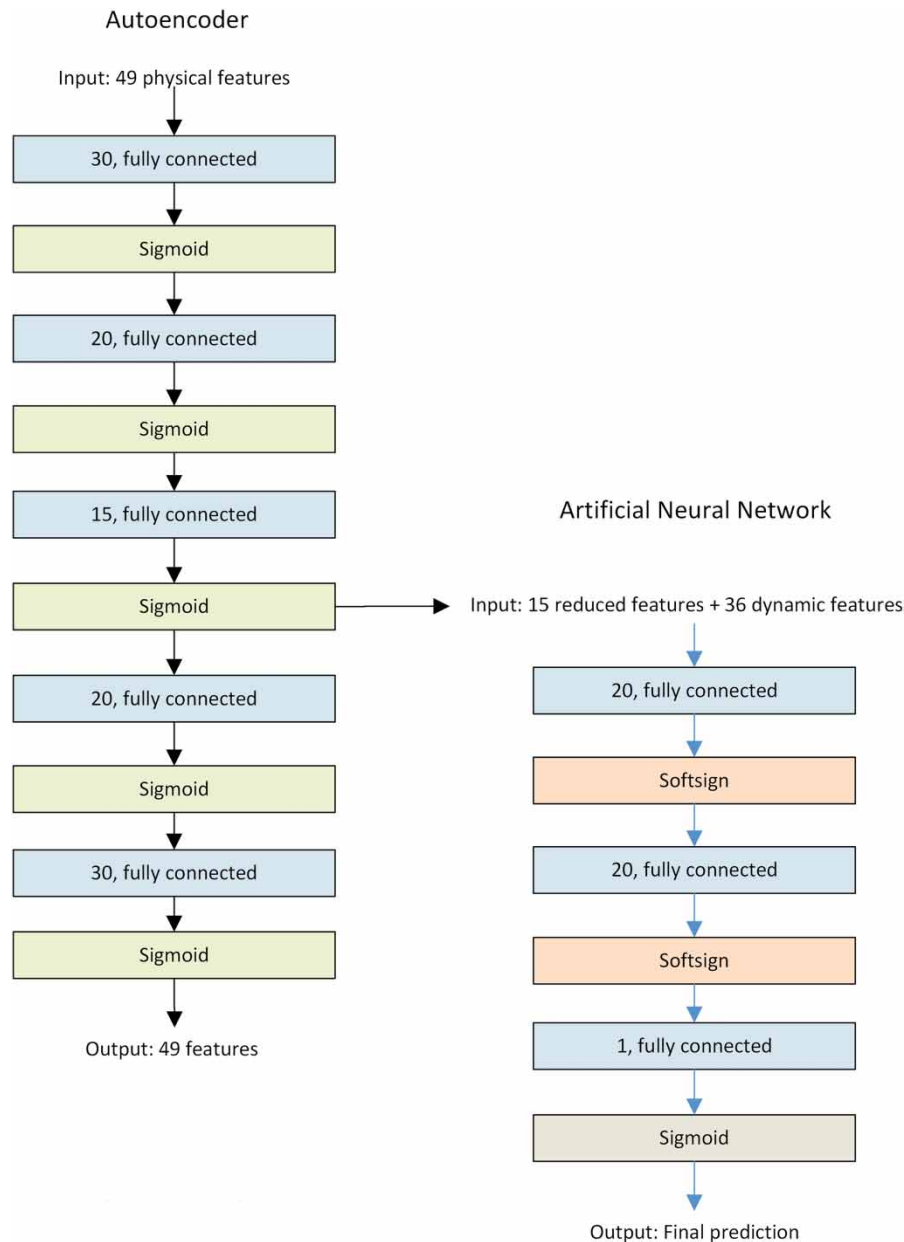


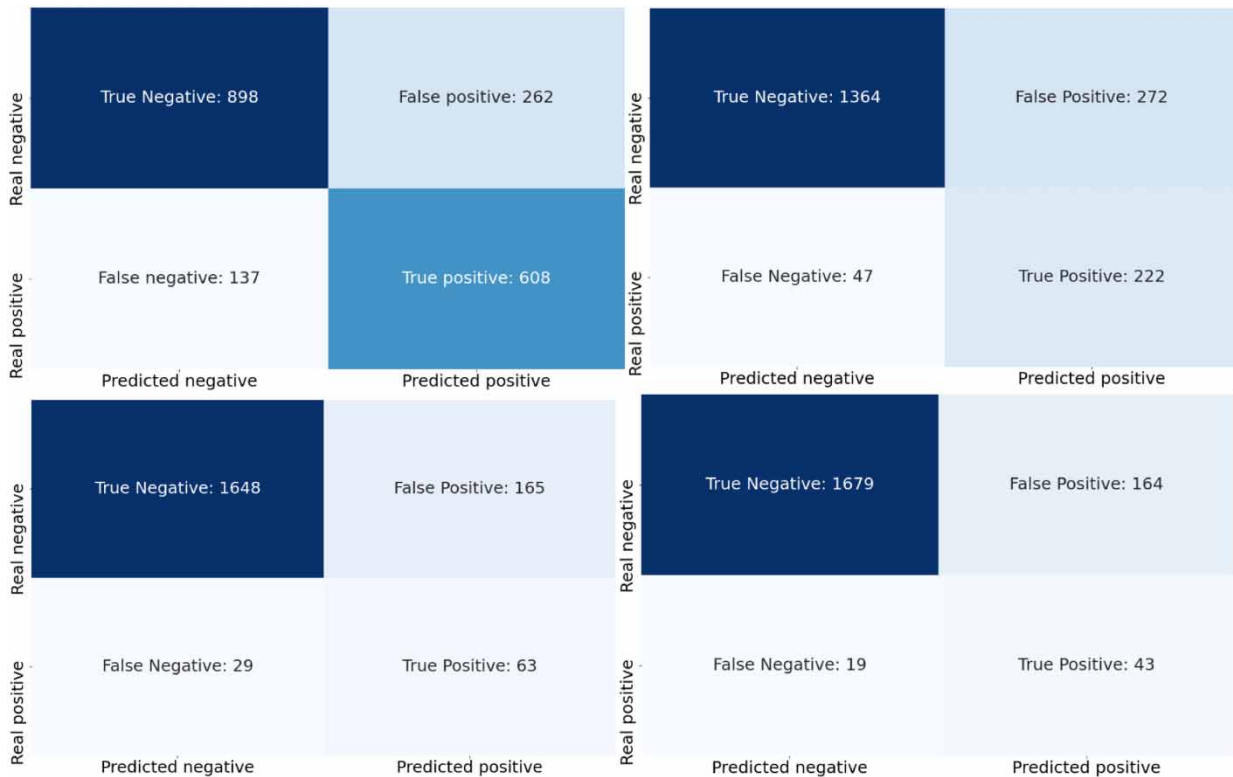
Figure 2 | Final architecture.

threshold being predicted within the confusion matrix. The $>15\%$ threshold produced the best results, whereas false predictions were typically in $>10\%$ and $>20\%$.

To understand better the sediment classifications, we need to compare the predictions using real sediment deposition. Figure 5 shows a boxplot of the different thresholds and the deposition values assigned to each. The Y axis indicates the real deposition value and the X axis indicates the result of (1) doing the individual future classification by each model and (2) merging the predictions using the strategy already indicated to obtain Figure 4. These plots focus on different aspects of the model's predictions. The key purpose of using boxplots in this evaluation is to provide a broad, overall view of how the predictive models behave across different sediment deposition thresholds. Unlike the confusion matrix, which gives a detailed breakdown of prediction accuracy and errors, the boxplot aggregates the data to show trends and variations in prediction accuracy and the actual values' distribution within each predicted category. This helps in understanding the models' general tendencies, such as consistent overestimation or underestimation within certain thresholds.

Table 9 | Results of the final architecture for different classification thresholds

Classification type	Specificity	Sensitivity	Accuracy	AUC	Precision	Balance of positive values
>5%	0.76	0.88	0.81	0.82	0.70	39%
>10%	0.80	0.85	0.81	0.83	0.43	14%
>15%	0.85	0.73	0.85	0.80	0.26	4.82%
>20%	0.9	0.68	0.89	0.79	0.21	3.25%

**Figure 3** | Confusion matrix results for the different classification types. Upper left: 5%. Upper right: 10%. Lower left: 15%. Lower right: 20%.

The <5% threshold holds most values within the correct range, with a maximum value of 7 and a minimum value of 0, and some higher outlier values. The >5% threshold has most values between 3 and 8, with a median value of 5.5, a maximum value of 15.5, and a minimum of 0. The >10% has most values between 4 and 11, which matches with the 226 under threshold classifications indicated in Figure 4, and has a maximum and minimum value of 18 and 1. The >15% has most values between 6 and 17, with a median value of 12, a maximum of 33, and a minimum of 3. Finally, the >20% encompasses a longer range of values, between 5 and 20, with a maximum of 40 and a minimum of 1.

When comparing the various thresholds, it becomes clear that the <5% threshold can correctly classify the range of values, but the other thresholds tend to include some positive values that are falsely classified within the 0–5% range. The >5% and >10% thresholds tend to have most values falling within a similar space, but the maximum and minimum values classified differ, with the latter having a higher range. Moving on to the >15% and >20% thresholds, these two have a much wider range of values, but the >20% threshold seems to suffer from a large number of FPs due to the decision-making process defined earlier. Overall, the thresholds tend to show an increasing deposition value range, with higher deposition values being classified in higher thresholds.

	<5%	>5%	>10%	>15%	>20%
<5%	758	147	127	54	70
>5%	114	165	99	38	60
Real >10%	9	26	89	23	30
>15%	0	4	11	9	10
>20%	0	2	15	8	37
	<5%	>5%	>10%	>15%	>20%
	Predicted				

Figure 4 | Confusion matrix of all possible classes.

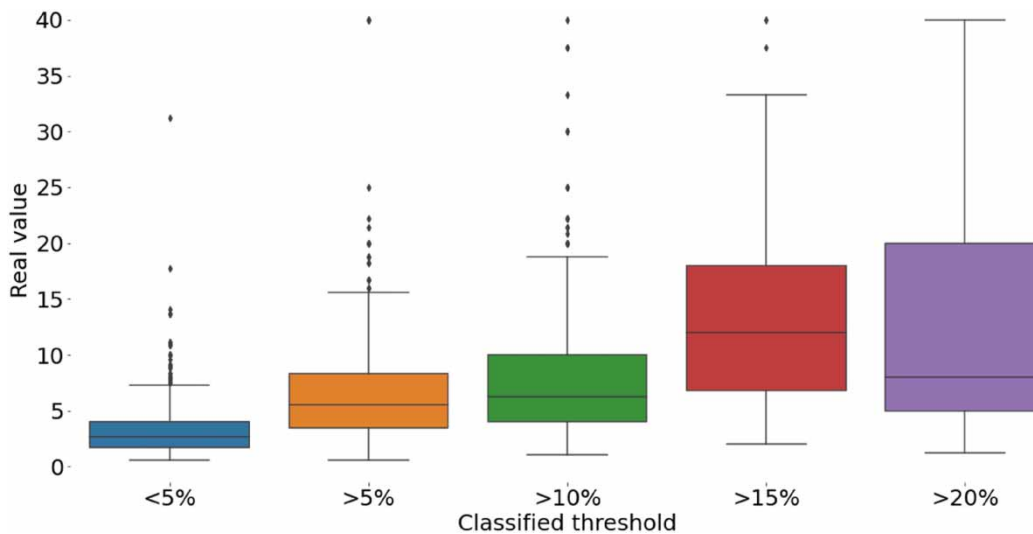


Figure 5 | Boxplot of the sediment deposition values classified to a threshold.

5. DISCUSSION

Sediment deposition estimation in a sewer section is a challenge that has been researched for a long time (Ribalta *et al.* 2022), studying predictions in real time, short term, and long term. However, attempting to build a predictive solution without sensor data along the entire sewer requires a dataset that represents the behaviour of the sewer at a general scope and further optimization of the features to extract all possible details. The complexity of the problem lies in the structure of the information since generating a descriptive data model requires enough features to indicate the physical properties of the infrastructure, the applied maintenance, and the environmental effects that can influence the accumulation process. A high number of features need to be followed by a high number of registers to enable enough learning information for the algorithms to comprehend the inner behaviour of each parameter and the generic relationship between all of them and the objective variable. This fact directly impacts the type of algorithm to be used to train since many state-of-the-art architectures require many

records to train correctly (Sevilla *et al.* 2022). Furthermore, the number of articles on the research topic yearly is not high (Ribalta *et al.* 2022), and there is no opportunity to compare a wide range of algorithms applied on multiple occasions within the domain of sewer sediment accumulation. While past studies use data-driven models to predict sediment deposition at the present (Jiang *et al.* 2021; Ribalta *et al.* 2022; Vanegas *et al.* 2022), the proposed model predicts sediment deposition in the future maintenance session. Additionally, as demonstrated in a study by Rosin *et al.* (2022), incorporating rainfall information into the predictive model is beneficial and might be necessary for future studies of this type.

In this study, we use AEs to reduce the high number of input features and refine the final prediction, but its benefits to the predictive model are not big. The data of this case study have an imbalance between the number of features and the total number of registers, so obtaining more maintenance registers of the studied sections would benefit the training process of the predictive model. Therefore, there is still room to improve and demonstrate the AEs' performance when the number of registers is consistent, not being affected by the curse of dimensionality.

Predicting future sediment deposition in this study is subject to certain challenges. The ideal scenario would involve an exact and regular time interval between maintenance sessions, but this approach is being replaced by the dynamic optimization of maintenance schedules (Tscheikner-Gratl *et al.* 2019). In most sewer sections in this study, the interval between maintenance sessions varies, making it difficult for the model to interpret and understand seasonal variations in sediment deposition. Nonetheless, this study proposes an architecture capable of predicting a range of future deposition within various time horizons, which can be replicated in other case studies. The objective variable is an essential component of the predictive solution, providing a small range of deposition values that can be effectively utilized by water utilities to aid in their maintenance sessions, despite not offering the exact deposition value.

The predictive ranges presented in the study are selected from the historical maintenance data, considering the number of registers available for each range and category. For this study, the range with more registers to offer is the <5% threshold, with 1,156 registers, and is also the one with the best predictions. As the threshold increases, the dataset becomes increasingly unbalanced, with fewer registers above the threshold, leading to a decrease in the quality of predictions, as it happens in many other ML problems (Thabtah *et al.* 2020; Ghosh *et al.* 2022). Nevertheless, the model predictions keep providing reliable information to the water utility since the percentage of TNs is higher in most cases.

Using the trained models in the four possible thresholds gives more information as a final output, but the results are also more difficult to understand. We provide a simple interpretation of the results, prioritizing the predictions with a higher sediment deposition, although there are other potential ways to improve this process. When the models predict, the output value ranges between 0 and 1, indicating the confidence value of the model towards predicting if the deposition will meet the threshold or not. In this case, a final function could be defined to decide which of the models' outputs has sufficient confidence to make the most likely prediction. This function could be defined using a new data-driven model, receiving the output of the models and deciding which of the thresholds is correct.

Water utilities can leverage the models as decision-support tools to plan future maintenance sessions for each section. The results indicate several conditions met by the models that can be used in some scenarios. When using all the models, the classifications into the class <5% are mostly correct, providing a clear indication of the sections with minimal deposition. While utilizing each model individually, one must be aware of the implications. The >5% model can be employed in all situations due to the stability in the predictions and good behaviour, but the other models produce a high number of FPs, rendering them useful only when applying maintenance to sections with a lower expected deposition that has no adverse impact. Additionally, the models can accurately classify situations where the result should be positive, with a low number of FNs.

6. CONCLUSION

In this study, we show the application of an ML architecture to predict the future sediment deposition in sections of a sewer system. Using real data from the city of Barcelona, we present an architecture that handles a dataset with the curse of dimensionality, objective variable imbalance, and long-term estimations.

In addressing the curse of dimensionality, our architecture incorporates a data reduction technique through the use of an AE, which adeptly compresses variables that represent the static physical properties of sewer sections. This approach streamlines the dataset and enables the predictive model to more effectively discern the relationship between predictive and objective variables with less historical data. Furthermore, the AE output is used by an ANN, but there are more algorithms

and architectures to be tested in other case studies since feature reduction may be beneficial to algorithms that require a low number of variables. The current architecture is being tested using a rather low number of historical registers and doing it with a high number would let the community know if the impact on data reduction positively increases the deposition classification. We also compare the performance of the architecture with traditional ML algorithms within the study. Combining the AE and the ANN, we obtain a score of a specificity of 0.76, a sensitivity of 0.88, and an average of 0.82 between them (ASS), and the combination of AE with the best-performing algorithm (Gradient Boosting) results in a score of a specificity of 0.82, a sensitivity of 0.66, and an average (ASS) of 0.74, meaning that within our study, the ANN performs better.

These tests have been carried out using the limit that considers sediment deposition greater than 5%, but the chosen architecture is evaluated in four different classification models, ranging from 5 to 20% sediment deposition. This method reduces the complexity of estimating the future sediment deposition value by using a category that englobes a range of possible values instead of focusing on predicting the exact value, enabling long-term predictions of sediment deposition. In our study, we use four ranges, but future studies might have a different data structure with a different balance of positive values that may enable the study of higher thresholds. Those cases would need to redefine the categorical ranges by looking into the balance of positive and negative registers and verifying that the presented architecture works properly using our scoring methodology.

Although the study presents solutions to prevalent issues within sewer system datasets and outlines strategies for long-term prediction, there is a need for further exploration into several areas. Among these is to enhance the dataset diversity. Demonstrating the architecture's efficacy across various deposition scenarios necessitates increasing the volume of data, particularly data reflecting higher sediment deposition percentages. Expanding our dataset would facilitate the examination of different thresholds and identify which yields the most insightful information and which can be most accurately predicted using this architecture. It is important to emphasize that the objective of this study is not to simulate the actual behaviour of the sewer system with the predictions but rather to enhance sewer management decision-making. Therefore, the necessary data should not be related to real-time sediment behaviour, such as sediment transport.

To enhance the effectiveness of future studies in this field, it is crucial to identify the relative importance of different features in the model prediction process. This would boost the data preparation process, allowing more time to be devoted to other crucial steps in model development. In addition, a thorough understanding of feature importance can reduce the time spent to decide which data need to be collected, thereby optimizing the overall research process. An important technique that could be used in future studies to identify this importance is explainable artificial intelligence, which analyses the impact of the different features in the predictions done by the ML models.

FUNDING

The work described in this paper has been conducted within the project SCOREwater. This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751. Marc Ribalta acknowledges funding from AGAUR DI-2019-066.

DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

CONFLICT OF INTEREST

The authors declare there is no conflict.

REFERENCES

- Al-Azzawi, M. S. M., Gondhalekar, D. & Drewes, J. E. 2022 *Neighborhood-scale urban water reclamation with integrated resource recovery for establishing nexus city in Munich, Germany: Pipe dream or reality?* *Resources* **11** (7), 64. doi:10.3390/resources11070064.
- Allawi, M. F., Abdulhameed, U. H., Adham, A., Sayl, K. N., Sulaiman, S. O., Ramal, M. M., Sherif, M. & El-Shafie, A. 2023a *Monthly rainfall forecasting modelling based on advanced machine learning methods: Tropical region as case study.* *Engineering Applications of Computational Fluid Mechanics* **17** (1). doi:10.1080/19942060.2022.2073565.
- Allawi, M. F., Sulaiman, S. O., Sayl, K. N., Sherif, M. & El-Shafie, A. 2023b *Suspended sediment load prediction modelling based on artificial intelligence methods: The tropical region as a case study.* *Heliyon* **9** (8), e18506. doi:10.1016/j.heliyon.2023.e18506.
- Ashley, R. M., Fraser, A., Burrows, R. & Blanksby, J. 2000 *The management of sediment in combined sewers.* *Urban Water* **2**, 263–275. doi:10.1016/S1462-0758(01)00010-3.

- Ashley, R., Bertrand-Krajewski, J. L. & Hvitved-Jacobsen, T. 2005 Sewer solids-20 years of investigation. *Water Science & Technology* **52** (3), 73–84. doi:10.2166/wst.2005.0063.
- Bailey, J., Keedwell, E., Djordjevic, S., Kapelan, Z., Burton, C. & Harris, E. 2015 Predictive risk modelling of real-world wastewater network incidents. *Procedia Engineering* **119**, 1288–1298. <https://doi.org/10.1016/j.proeng.2015.08.949>.
- Bailey, J., Harris, E., Keedwell, E., Slodoban, D. & Kapelan, Z. 2016 Developing decision tree models to create a predictive blockage likelihood model for real-world wastewater networks. *Procedia Engineering* **154**, 1209–1216. doi:10.1016/j.proeng.2016.07.433.
- Bhagat, S. K., Tiyasha, T., Awadh, S. M., Tung, T. M., Jawad, A. H. & Yaseen, Z. M. 2021 Prediction of sediment heavy metal at the Australian Bays using newly developed hybrid artificial intelligence models. *Environmental Pollution* **268**, 115663. doi:10.1016/j.envpol.2020.115663.
- Bradley, A. P. 1997 The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*. **30** (7), 1145–1159. doi:10.1016/S0031-3203(96)00142-2.
- Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. P., Basto, J. P. & Alcalá, S. G. S. 2019 A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering* **137**, 106024. <https://doi.org/10.1016/j.cie.2019.106024>.
- Çınar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M. & Safaei, B. 2020 Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability* **12** (19), 8211. <https://doi.org/10.3390/su12198211>.
- Daguillard, R. 2016 EPA Survey Shows \$271 Billion Needed for Nation's Wastewater Infrastructure. Available from: <https://www.epa.gov/archive/epa/newsreleases/epa-survey-shows-271-billion-needed-nations-wastewater-infrastructure.html> (accessed January 2023).
- Ghosh, K., Bellinger, C. & Corizzo, R., Branco, P., Krawczyk, B. & Japkowicz, N. 2022 The class imbalance problem in deep learning. *Machine Learning*. doi:10.1007/s10994-022-06268-8.
- Hassouna, M., Reis, M., Fairuz, M. & Tarhini, A. 2019 Data-driven models for sewer blockage prediction. In *2019 International Conference on Computing, Electronics & Communications Engineering*, pp. 68–72. doi:10.1109/iCCECE46942.2019.8941848.
- Jiang, Y., Li, C., Zhang, Y., Zhao, R., Yan, K. & Wang, W. 2021 Data-driven method based on deep learning algorithm for detecting fat, oil, and grease (FOG) of sewer networks in urban commercial areas. *Water Research* **207**. doi:10.1016/j.watres.2021.117797.
- Kramer, M. A. 1993 Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal* **37** (2), 233–243.
- Kumar, A., Chinnam, R. B. & Tseng, F. 2019 An HMM and polynomial regression based approach for remaining useful life and health state estimation of cutting tools. *Computers & Industrial Engineering* **128**, 1008–1014. doi:10.1016/j.cie.2018.05.017.
- Mohtar, W. H. M. W., Afan, H., El-Shafie, A. & Bong CHJ, G. A. A. 2018 Influence of bed deposit in the prediction of incipient sediment motion in sewers using artificial neural networks. *Urban Water Journal* **15** (4), 296–302.
- Okwori, E., Viklander, M. & Hedström, A. 2021 Spatial heterogeneity assessment of factors affecting sewer pipe blockages and predictions. *Water Research* **194**, 116934. doi:10.1016/j.watres.2021.116934.
- Ran, Y., Zhou, X., Lin, P., Wen, Y. & Deng, R. 2019 A survey of predictive maintenance: Systems, Purposes and Approaches. Arxiv. doi:10.48550/arXiv.1912.07383.
- Ribalta, M., Mateu, C., Bejar, R., Rubión, E., Echeverria, L., Varela Alegre, F. J. & Corominas, L. 2021 Sediment level prediction of a combined sewer system using spatial features. *Sustainability* **13** (7), 4013. doi:10.3390/su13074013.
- Ribalta, M., Mateu, C., Bejar, R. & Rubión, E. 2022 Machine learning solutions in sewer systems: A bibliometric analysis. *Urban Water Journal* **20** (1), 1–14. doi:10.1080/1573062X.2022.2138460g.
- Robles-Velasco, A., Cortés, P., Muñuzuri, J. & Onieva, L. 2021 Estimation of a logistic regression model by a genetic algorithm to predict pipe failures in sewer networks. *OR Spectrum* **43**, 759–776. doi:10.1007/s00291-020-00614-9.
- Rosin, T. R., Kapelan, Z., Keedwell, E. & Romano, M. 2022 Near real-time detection of blockages in the proximity of combined sewer overflows using evolutionary ANNs and statistical process control. *Journal of Hydroinformatics* **24** (2), 259–273. doi:10.2166/hydro.2022.036.
- Sattar, A. M. A., Ertugrul, O. F., Gharabaghi, B., McBean, E. A. & Cao, J. 2019 Extreme learning machine model for water network management. *Neural Computing and Applications* **31**, 157–169. doi:10.1007/s00521-017-2987-7.
- Sevilla, J., Heim, L., Ho, A., Besiroglu, T., Hobbhahn, M. & Villalobos, P. 2022 Compute trends across three eras of machine learning. Arxiv. doi:10.48550/arXiv.2202.05924.
- Song, Y. H., Yun, R., Lee, E. H. & Lee, J. H. 2018 Predicting sedimentation in urban sewer conduits. *Water* **10** (4), 462. doi:10.3390/w10040462.
- Sorzano, C. O. S., Vargas, J. & Pascual-Montano, A. 2014 A survey of dimensionality reduction techniques. Arxiv. doi:10.48550/arXiv.1403.2877.
- Tavakoli, R., Sharifara, A. & Najafi, M. 2020 Prediction of pipe failures in wastewater networks using random forest classification. In: *Pipelines 2020*. doi:10.1061/9780784483206.011.
- Thabtah, F., Hammoud, S., Kamalov, F. & Gonsalves, A. 2020 Data imbalance in classification: Experimental evaluation. *Information Sciences*. **513**, 429–441. doi:10.1016/j.ins.2019.11.004.
- Tscheikner-Gratl, F., Caradot, N., Cherqui, F., Leitão, J. P., Ahmadi, M., Langeveld, J. G. & Le Gat, Y. 2019 Sewer asset management – State of the art and research needs. *Urban Water Journal* **16** (9), 662–675. doi:10.1080/1573062X.2020.1713382.
- Vanegas, S., Montes, C. & Saldarriaga, J. 2022 Prioritizing inspection of sewer pipes based on self-cleansing criteria. *Urban Water Journal* 1–10. doi:10.1080/1573062X.2022.2035408.
- Wang, Y., Yao, H. & Zhao, S. 2016 Auto-encoder based dimensionality reduction. *Neurocomputing* **184**, 232–242. doi:10.1016/j.neucom.2015.08.104.