Data-Driven Decision Support for Water Pump Maintenance in Tanzania: A Predictive Modelling and Analysis Study

Filip Kornata and Adam Williams

***Abstract*—Communities across Tanzania rely on water pumps for access to potable water, however, such pumps are prone to failure. A method to predict which pumps are likely to fail would be invaluable to pump maintenance operations, enabling faster repair of pumps upon failure. In this work, we propose several improved approaches to water pump failure prediction using data from Taarifa and the Tanzanian Ministry of Water. We utilise a range of improved data imputation and feature engineering approaches; encoding methods including target encoding and ordinal encoding; feature selection methods such as Fast Correlation-Based Filter and two custom approaches; dimensionality reduction with Truncated Singular Value Decomposition (TSVD), and feature normalisation with Z-score and Min-Max normalisation. We propose a range of models, including ensemble and non-ensemble models utilising XGBoost, Bagging Classifier, LightGBM, AdaBoost, Neural Networks, Random Forests, Boosted Forests, Extra Trees, as well as stacking and voting methods. We achieved a best performance of 0.8242 on the Kaggle challenge page with a weighted vote of Random Forest, XGBoost, CatBoost and Bagging Classifier. Outside of the competition, we achieved a result of 82.44% on validation set using a Stacking Classifier based on the same models. This approach can be used by the government to improve maintenance operations in the country.**

I. Introduction

COMMUNITIES across Tanzania are heavily reliant on water pumps for access to clean, potable water, so continual maintenance of pumps is essential. Data from Taarifa and the Tanzanian Ministry of Water has been made available to facilitate the creation of a system which can predict water pump failure across the country, helping to improve maintenance operations and resource allocation. Our goal is to create a system which provides feature analysis and predictive modelling on this dataset to provide decision support for the ministry and local governments.

A. Dataset Description

The dataset used in this work was collected using crowdsourcing platform Taarifa, where members of communities can report water pump functionality. The provided dataset contains entries for 74,250 pumps across the country, split into training and testing datasets of N=59,400 and N=14,850 respectively. The dataset contains 39 features describing a range of information about each pump, including factors such as geographic location, operator, type, surrounding population, construction year and water quality. For each pump, its current functionality is given in a label split into 3 classes: Functional, Non-Functional, and Functional (Needing Repair) – these classes are imbalanced, with 54.3% Functional (N=32,259), 38.4% Non-Functional (N=22,824), and 7.3% Needing Repair (N=4,317).

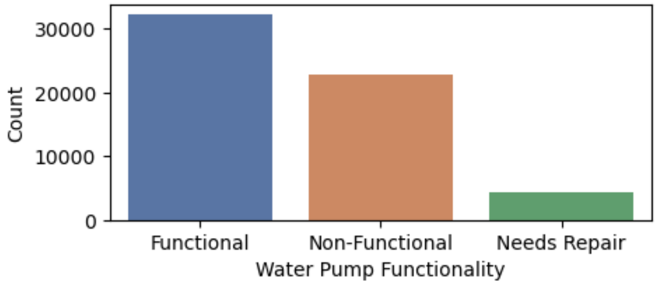

Fig. 1. Bar Plot showing the class distribution of Water Pump Functionality

B. Research Questions

The goal of this project is to provide insights which could allow a decision maker to better allocate maintenance resources across Tanzania, and possibly lead to improved pump installation considerations into the future. We have formulated 5 research questions to provide these insights:

1) What are the main factors associated with pump failure or malfunction, and how do these vary across different regions of Tanzania?
2) Which operators and/or management groups have the highest success rates in maintaining water pumps, and how do these rates vary based on factors which may make pump maintenance easier, such as water cost, pump type, or location remoteness?
3) What are the interactions between different features, such as water quantity and pump type, which could provide insights into the underlying causes of pump failure?
4) How does the age of a water pump relate to its functionality, and is there a point at which pumps become significantly more likely to break down or require replacement?
5) What are the most reliable machine learning methods for pump failure prediction?

II. Related Work

Water pump failure prediction, as with similar issues such as pipe failure prediction, have garnered significant attention in the machine learning field.

In the approach used by Darmatasia et al. [1] on the same dataset, Correlation Matrices and Recursive Feature Elimination (RFE) were used as feature selection methods. The latter constructs multiple random forest models from subsets of a dataset to identify features with high predictive potential. The study experimented with four machine learning classification methods including XGBoost (XGB), Random Forest (RF), Gradient Boosting Machine (GBM), and Support Vector Machine (SVM), achieving a best accuracy of 80.38% with XGB, using features selected by RFE.

Bejarano et al. [2] apply XGB and RF ensemble learning methods to the Tanzania Water Pump Dataset, achieving identical performance scores of 0.696 with both an XGB ensemble and an RF ensemble respectively, outperforming a Support Vector Machine (SVM) approach. In their approach, they also successfully ranked each of the features by their importances, finding that *latitude*, *longitude* and *gps_height* were consistently some of the most important features, calculated using the inherent feature importances from the XGBoost tree model. An overall *accuracy* metric was not utilised, making it difficult to compare results between the two methods, despite the same dataset being used.

Neither of the above studies specifically addressed the water pump class imbalance problem discussed in [1], a major flaw with these approaches. Several papers have proposed solutions to class imbalance problems in similar problems: Liu et al. [3] utilised undersampling, oversampling, and Synthetic Minority Oversampling Technique (SMOTE), three methods used to alleviate the issue of class imbalance, in their water pipe failure study. According to this study, SMOTE was most effective at increasing the performance score, followed by oversampling and then undersampling. SMOTE functions as a class balancing method by generating synthetic data entries using a k-Nearest Neighbours (kNN) approach and linear interpolation [4]. SMOTE-NC (Synthetic Minority Oversampling Technique-Nominal Continuous) is a generalised version of SMOTE that extends oversampling to categorical features [4], which shows promise.

A wider range of other machine learning classification approaches have been utilised. Artificial Neural Networks are a popular approach to classification problems, being successfully used by a range of methods for pipe failure prediction [5], [6], [7], [8]. Despite this popularity, ANNs have been outperformed by a range of methods, including SVMs, LightGBM, and RFs for water pipe failure prediction [9], [10], while they were still superior for groundwater quality prediction [6]. Decision trees have also been proposed quite widely for similar problems [11], with them outperforming SVM approaches in sewer pipe failure prediction in one study [12]. XGB has also been proposed in further problems, with a Bagged XGB approach being utilised by Deng et al. [13]. Dimensionality Reduction approaches such as Principal Component Analysis (PCA) have also seen some success in different problems, such as in Financial Fraud Detection [14] or vibration-signal based identification of leaks in water pipes with [15].

III. Methodology

A. Exploratory Data Analysis

According to Steven's four-level scale, there are 31 nominal features, 5 ratio features, 2 interval features, no ordinal features, and one unclear feature (num_private). Many of the numerical values show central tendency towards 0, which is likely related to faulty data input, leading to inaccuracies in standard deviation. The only relatively well- distributed numerical feature is latitude with approximate mean -5.7 and standard deviation 2.95. Basin and source_type are the most equally distributed nominal features in the dataset, while the rest of the features are dominated by one or more prominent values. Below, we show the percentage of missing values in columns with missing data.

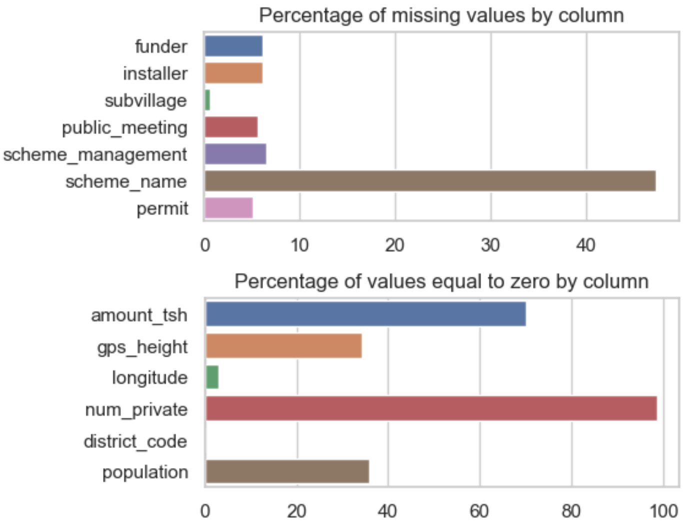

Fig. 2. Diagram showing missing values in the dataset by feature.

We identified several salient features in the dataset, which were highly indicative of pump functionality. This was calculated by extracting the intrinsic feature importances from a Random Forests model, which is based on each feature's position in the tree-based model, with more important features being used earlier in the decision tree calculation. The best predictor of pump functionality was consistently the *quantity* feature. This feature represents the volume of water available at each pump.

Older water pumps could also be expected to fail more often. *Figure 4* shows the distribution of pumps by construction year, and their functionality. The majority of pumps were constructed between 2008 and 2010, with most of these pumps remaining functional.

There are multiple features in the dataset that represent very similar information in a different way, like *region_code* and *region*, or *management* and *scheme_management*, while *quantity_group* and *quantity* convey the exact same information. Some missing values are implicit and not easy to identify. For example, *latitude* feature contains values close to zero, while none part of Tanzania is located in that coordinates. Some other categorical features like *installer* and *funder* have very

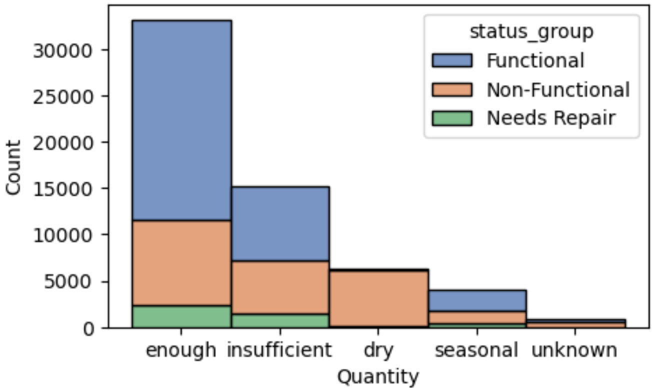

Fig. 3. Histogram showing the distribution of water pumps by Construction Year and functionality.

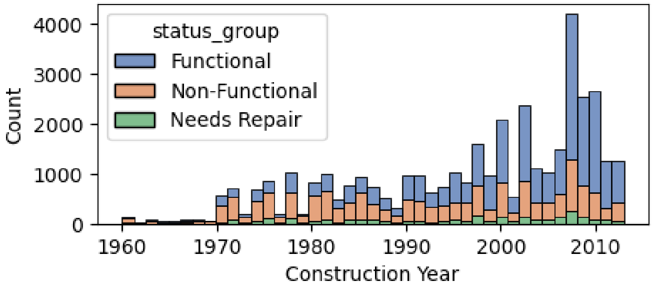

Fig. 4. Histogram showing the distribution of water pumps by Construction Year and functionality.

high cardinality with most of the categories containing very vew data samples.

B. Pre-Processing

We base some of our pre-processing approach on an existing solution by Brenda Loznik [16]. This is due to her highly extensive implementation of many data imputation, feature engineering, and feature selection approaches. We decided to use her implementation in order to avoid simply re-inventing the wheel, and to instead build on her solution in order to achieve an improved performance, which we were ultimately successful in doing. We have clearly provided full credit to her solutions in our code, where used. In this section, we will briefly describe the implemented approach we used, making clear our own contributions.

1) Data Imputation: Many columns in the dataset contain missing values, requiring data imputation. It is important that suitable imputation methods are used in order to avoid wrongly adjusting the feature's original distribution, and thus negatively impacting our model's performance. We experiment with a range of data imputation methods for various columns.

For *latitude*, *longitude* and *gps_height* imputation, missing values are imputed based on the mean value of other pumps in the closest possible region match. To achieve this, several categorical variables, including *subvillage*, *lga*, *ward*, *region*, and *basin*, and based on the highest resolution feature available, the mean latitude and longitude of other pumps from this match are used. We identified the need to impute the *latitude* feature, as we identified many near-zero values, which was impossible since no part of Tanzania lies on the equator.

For *gps_height*, we experiment with using two further imputation approaches. We experimented with using an external dataset, the Shuttle Radar Topography Mission (SRTM), to impute the height value. This yielded improved practical results for our model, however, we did not use it for our final challenge submission, as it is against the challenge rules. Secondly, we propose a custom K-Nearest Neighbours (KNN) approach to impute the height feature, by averaging the GPS height of the 5 nearest pumps.

For *installer* and *funder*, we experimented with two approaches for feature imputation. In both approaches, the first step was to convert each value to lowercase, to ensure that identical values were grouped properly by capitalisation. In the first approach, initially implemented by Loznik, we simply selected the top 150 values from the columns by count, and binned the remaining values into an *other* category. This left a very large portion of the dataset in this ”other” category, so we experimented with a second approach. In this second approach, we still performed binning into an *other* category, but only categories with less than 10 total instances were binned into this category. This left a large number of categories (N=437), so we performed target encoding to reduce the dimensionality of the feature. In target encoding, each category is set to a value between 0 and 1 depending on the ratio of functional to non-functional pumps. Categories with 100% pump functionality are assigned a value of 1, while categories with 0% functionality are assigned a value of 0.

For *public_meeting*, mode encoding is used, as 90% of pumps have a TRUE value. Meanwhile, to impute *permit*, the mode of similar records is used, based on the *management_group* and *public_meeting* features. To impute *construction_year*, the mean of each row's matching *extraction_type_group* is taken. Lastly, a similar mode imputation approach for *scheme_management*, using the *management_group*, *scheme_management*, and *management* features.

2) Feature Engineering: Many features contain information which can be ”unlocked” through engineering, but which would otherwise be near-useless. We have provided a list of our feature engineering approaches below, which each generate a new feature column.

- *age*: This represents the amount of time between pump construction and the year in which the pump's functionality was recorded. It is calculated by taking the year component of *date_recorded*, and subtracting *construction_year* from this. This feature may provide a more reliable feature for pump age than *construction_year*.
- *Recorded Season*: We utilised Loznik's approach to generate a feature for the season in which the water pump's functionality was taken. Tanzania has wet and dry seasons[17], which are highly likely to have an impact on water pump functionality, due to wells drying out or becoming flooded.
- *amount_tsh missing*: While amount_tsh itself is a poor feature due to the large amount of missing values, cre-

ating a binary feature for whether the value is missing proved to be a reasonable indicator of pump functionality.

- *pump_density*: This feature represents the distribution of nearby pumps, calculated from the *latitude* and *longitude* features. We implemented this by taking the 6 nearest neighbours from a given pump, and calculating the mean of the distances to these pumps. This feature is salient as each pump's remoteness may be likely to influence how often it can be serviced and how accessible it is to maintenance crews. As discussed later, this feature proved to be a good indicator of pump functionality.

We also perform encoding on several categorical columns. Encoding is the process of transferring a feature's data to another format to facilitate easier analysis. A brief overview of the encoding methods used is provided below:

Target Encoding: Target Encoding is the process of converting categorical data into a continuous value between 0 and 1, We utilised Target Encoding on *installer*,*funder*, *region*, and *lga*.

Ordinal Encoding: For ordinal encoding, each feature is simply given a unique integer value. We found that this worked best on all regional features.

3) Feature Normalisation: Feature Normalisation is the process of scaling and transforming the numerical features of a dataset to a common scale to prevent the dominance of certain features based on their magnitudes. We experiment with using a range of normalisation methods, for various categories:

Min-Max Scaling: Min-Max Scaling is the process of scaling a feature to values between 0 and 1, maintaining its original distribution, based on a feature's minimum and maximum values. We use this with *latitude*, *longitude*, and *gps_height*, as we would like to preserve the original distribution. This method is particularly sensitive to outliers, so has not been used more extensively.

Z-Score Normalisation: Z-Score Normalisation, or standardisation, is the process of transforming a feature's values to have a mean of 0 and a standard deviation of 1, by subtracting each value by the mean and dividing by the standard deviation. We apply this to *age* and our engineered *pump_density* feature.

Robust Scaling: Robust Normalisation is a feature scaling method which scales values to have median 0 and valuenbetween -1 and 1 for values within the interquartile range. This scaling method is less sensitive to outliers compared to Z-score normalisation and Min-Max scaling.

C. Feature Selection

Before using typical feature selection methods, we iteratively evaluate the performance of the engineered features compared to non-engineered ones using base-line Random Forest (RF) classifier. We measure the accuracy on the dataset without engineered features and in each iteration one of the features is swapped with the engineered version of it. Once the accuracy is compared, the dataset is reverted to its original shape and the next iteration is run. This method is later referred to as Iterative Cross-Validated Engineered Features Evaluation (ICVEFE).

While exploring the dataset, we identified several features that are highly related to others. These include: *waterpoint_type_group*, *source_class*, *quality_group*, *management*, *management_group*, *extraction_type_group*. To examine their usability, we use a similar method to the one above, but instead of swapping these with some feature, we simply add it to the dataset in each iteration. This method is later referred to as Iterative Cross-Validated Optional Features Evaluation (ICVOFE).

We experiment with various feature selection approaches to discover the optimal set of features to use in our dataset.

1) **Recursive Feature Elimination Cross-Validated (RFECV)**: RFECV is a wrapper-based method which recursively eliminates features based on their impact on a model's performance. We use a Random Forest (RF) classifier to provide a performance for each feature combination.
2) **Fast Correlation-Based Filter (FCBF)**: FCBF [18] evaluates each feature's correlation with the target variable, while minimizing redundancy among the features. It computes a merit score for each feature based on its correlation with the target and its correlation with other features.

D. Oversampling

We utilise SMOTE-NC (Synthetic Minority Oversampling Technique-Nominal Continuous) to perform oversampling on our dataset, due to the class imbalance between the pump functionalities, using the method as originally described in [4]. *Functional Needs Repair* is the most underrepresented class, with only 4,317 pumps of the overall 59,400 pumps in the train set belonging to this class. Using SMOTE-NC, we create synthetic data points to bring this class to a count of 7,000 pumps.

E. Dimensionality Reduction

Lastly, we also experiment with applying dimensionality reduction using Truncated Singular Value Decomposition (TSVD).

F. Classification

We test 12 different models in our approach, across a variety of pre-processing configurations. Several of these models have been utilised for this problem previously, however, we seek to provide improved metrics for all models, given our extensive improvements across the pre-processing pipeline. We utilise the following models:

RandomForestClassifier (RF): Ensemble method that combines predictions of multiple decision trees to produce final predictions. [19] **XGBClassifier (XGB)**: Providing a paralllel tree bosting mechanism using multiple weak learners (low-depth decision trees). [20] **CatBoostClassifier (Cat)**: A similar approach to XCB extended with better handling of categorical features. [21] **BaggingClassifier (Bag)**: A meta-estimator using multiple base classifiers like XGBoosts or

Decision Trees on random subsets of the dataset. [22] **StackingClassifier (Stack)**: A meta-classifier fitted on the outputs of multiple individual classifiers. [23] **A Weighted Vote classifier (WV)**: Various weighted combinations of the above models. **XGBClassifier Ensemble (ensXGB)**: Ensemble of multiple XGB models. **LightGBMClassifier (LGBM)**: A gradient boosting classifier for distributed training that uses histogram-based algorithm for finding the best feature split points. [24] **GradientBoostingClassifier (GB)**: A less memory-efficient version of the above that uses pre-sorted algorithm to find the best feature split. [25] **AdaBoostClassifier (Ada)**: Another modified gradient boosting method. [26] **ExtraTreesClassifier (Ext)**: A ensemble method using different random subsets of features for each classifier. [27] **MLPClassifier (NN)**: A solution using Neural Networks for classification.

IV. Experimental Setup

A. Experiment Configuration

We used the same device with Apple M1 Pro processor and 16GB of RAM (Random Access Memory) to perform all the experiments. Due to issues with using LightGBM model on ARM-based processor, it was trained on a different machine.

For the feature selection methods, that involve modelling, we use base-line Random Forest (RF). The results are obtained using cross-validation with 10 folds and the same random state variable for is used for each experiment.

We utilise a 80/20 train-test split to perform our evaluation. The split is always performed using the same random state variable to ensure fair comparisons between models. The entire experimental process is separated into two stages. The first stage includes manual parameters selection to establish the best pre-processing configuration, which is later used for the second stage, where we experiment with different models.

We decided on using RF for our experiments due to them having a demonstrably strong performance on this dataset and for similar problems [9], and due to its faster execution time in comparison to some of the larger models implemented, such as our 11 XGBoost ensemble.

B. Dataset Pre-Processing Configuration

Due to the many possible configurations of our pre-processing pipeline, we iteratively evaluate each configuration of one stage and select the best performing combination that is used for each following stage. To estimate each approach's performance, we utilise a tuned version of RF model with the following parameters:

- maximum tree depth: 30
- maximum number of features for splits: log2 of the total
- minimum number of samples for internal node split: 8
- number of trees: 150

In this experiment, we focus on the new methods suggested by us by comparing them with the best publicly-available solution to that problem. We initially compare all the imputation extensions we proposed. These include improved *latitude* feature imputation, using external data for *gps_height* imputation, and extended methods for dealing with high cardinality of *funder* and *installer* features. In Table I with their initial letters.

Then, using the best set of imputation techniques, we move on to feature engineering experiments. As we have a separate feature selection algorithm dedicated to evaluating all the engineered features, we only compared three configurations here. The first one uses a selected set of engineered and non-engineered features assembled in the previous solution. The second configuration extends that set with our engineered features, and in the last approach we use all the original features, previously engineered ones, as well as our new solution.

After finding the best set of features to proceed with, we compare feature selection methods. We tested ICVEFE, ICVOFE, RFECV and FCBF. Because FCBF drops a significant number of features, it hasn't been used alongside other methods. ICVEFE, which does a comparison between engineered and non-engineered features is highly likely to be useful, since leaving both versions of a feature introduces redundancy making the model unnecessarily complex. For this reason, combination of this method with ICVOFE and RFECV is also evaluated.

Three feature normalisation techniques were evaluated against a solution without any normalisation. The first two methods are Min-Max and Z-Score, and the third, custom solution, involves the use of both of them and Robust Scaling applied to relevant features based on Exploratory Data Analysis.

We evaluate TSVD for Dimensionality Reduction.

C. Model Configuration

For each model, we performed hyperparameter tuning to gain the best possible performance on the test set. Using the established pre-processing configuration, we utilised a parameter grid search to iteratively test and evaluate each combination of parameters. The table of these is provided in the Appendix.

Several common evaluation metrics are used to quantitatively evaluate our models. Relative True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) counts are used.

- **Precision** is the ratio of true positives to the total number of predicted positives.
- **Recall** is the ratio of true positives to the total number of actual positives.
- **Accuracy** is the total proportion of correct predictions.
- **F1 Score** is the harmonic mean of precision and recall.

V. Results

A. Pre-processing

As seen in I, although all of the methods used separately improve the previous approach, a combination of data imputation for *latitude*, *gps_height* and *funder* features, excluding *installer*, gives the best performance overall bringing up the accuracy from 81.42% to 81.68%. Although some other configurations have more impact on improving F1 Score, we proceeded with the best accuracy configuration.

The proposed feature engineering methods also achieve higher accuracy and F1 Score. Following that, ICVEFE and

TABLE I
PRE-PROCESSING CONFIGURATION

Data Imputation		
Method	**Accuracy**	**F1 Score**
Previous approach	81.42	80.44
latitude	81.46	80.46
gps_height	81.52	80.54
installer	81.56	80.63
funder	81.41	80.45
l + g	81.66	80.68
l + i	81.58	**81.62**
l + f	81.50	81.54
g + i	81.58	80.63
g + f	81.33	80.34
i + f	81.46	80.48
l + g + i	81.49	80.52
g + i + f	81.25	80.26
l + i + f	81.36	80.40
l + g + f	**81.68**	80.71
l + g + i + f	81.41	80.40
Feature Engineering		
Method	**Accuracy**	**F1 Score**
Previous approach (subset)	81.68	80.71
Previous + our engineered	81.80	81.36
All features + all our engineered	**81.88**	**81.42**
Feature Selection		
Method	**Accuracy**	**F1 Score**
Previous approach	81.88	81.42
ICVEFE	81.85	81.39
ICVOFE	81.76	81.31
RFECV	81.88	81.42
FCBF	77.42	76.78
ICVEFE + ICVOFE	**81.89**	**81.43**
ICVEFE + RFECV	81.87	81.40
Feature Normalisation		
Method	**Accuracy**	**F1 Score**
None	81.89	81.43
Min-Max	81.96	81.5
Z-Score	81.97	81.51
Custom	**81.98**	**81.52**
Dimensionality Reduction		
Method	**Accuracy**	**F1 Score**
None	**81.98**	**81.52**
TSVD	76.78	76.30

ICVOFE combined are able to select the most promising features. The only engineered feature selected by ICVEFE that showed better performance than the original one is *month* (over *season*). However, that is only the case when oversampling was used. When running the experiment without it, the following engineered features were selected: *region*, *latitude*, *gps_height*, *extraction_type*, *extraction_type_class*, *source*, *waterpoint_type*. All of the optional features are dropped by ICVOFE. When using RFECV for identifying important features, *pump_density*, which was engineered from other features was one of the selected ones.

All normalisation techniques showed improvement, but the custom method showed the highest accuracy and F1 Score. TSVD method for Dimensionality Reduction didn't improve the result.

After executing the full pre-processing pipeline, we reduce the number of features from 39 to 26, which significantly reduces the time cost of training and minimises redundancy.

TABLE II
OVERSAMPLING RESULTS

	Train		**Test**	
Oversampling	Accuracy	F1 Score	Accuracy	F1 Score
Yes	93.37	93.26	**81.98**	**81.52**
No	92.13	91.80	81.66	80.59

As depicted in Table II, oversampling the underrepresented class improved the model's accuracy, but more significantly it helped to better classify data samples in the *functional needs repair* class by improving its individual prediction accuracy from 64.35% to 81.86%.

B. Modelling

As Table III suggests, Stacking Classifier showed the highest performance among all the used models including accuracy, F1 Score and Recall metrics. Weighted Vote model performed slightly worse overall, but achieved the best Precision score. Ensemble of 11 XGBoost classifiers achieved the third best accuracy result. AdaBoost, Extra Trees and Neural Networks showed the lowest scores.

Making use of Random Forest's ability to determine feature importance, we identified nine features with strong significance. These are: *quantity*, *longitude*, *latitude*, *gps_height*, *pump_density*, *funder*, *age*, *waterpoint_type* and *population*. The rest of the features achieved a Relative Importance score below 0.04.

VI. DISCUSSION

A. Research Questions

We were able to successfully answer each of our research questions:

1) What are the main factors associated with pump failure or malfunction, and how do these factors vary across different regions of Tanzania?: Across Tanzania, we found that *quantity* was the main factor predicting pump failure, followed by *longitude*, *pump_density*, *latitude*, *funder*, *gps_height* and *age*. Quantity was the most influential factor as the feature is intrinsically indicative of pump functionality, with 96.9% of "dry" pumps being non-functional, and 65.2% of "enough" pumps being functional. Longitude and latitude are also strong indicators of pump functionality, suggesting that there is a spatial element to pump functionality, with certain regions having different quantities of functional pumps. Our engineered *pump_density* feature is also a strong predictor, with pumps located in a high pump density region being more likely to be functional. Lastly, *funder* and *age* are good predictors of status, as pump funder likely indicates that some funders invest into poor quality pumps, and the pump's age suggesting that breakdown is more likely as the pump ages.

Table V contains the most important feature for each of the 21 regions of Tanzania. In 10 regions, *quantity* was the most

TABLE III
MODEL PERFORMANCE

Model	Accuracy	F1 Score	Precision	Recall
RF	81.98	81.52	82.22	81.98
XGB	82.29	81.88	82.57	82.29
Cat	81.24	80.82	81.35	81.24
Bag	81.61	81.18	81.74	81.61
ensXGB	82.29	81.88	82.62	82.29
LGBM	70.40	68.80	71.12	70.40
GBC	81.44	80.95	81.83	81.44
Ada	69.97	66.42	71.71	69.97
Ext	71.73	68.15	75.28	71.73
NN	76.31	75.54	76.47	76.31
Stack	**82.44**	**82.10**	82.56	**82.44**
WV	82.37	81.96	**82.66**	82.37

TABLE IV
MOST IMPORTANT FEATURES FOR EACH REGION IN TANZANIA

Region	Feature	Region	Feature
Iringa	Quantity	Rukwa	Population
Mara	Age	Mwanza	Longitude
Manyara	Quantity	Kigoma	Longitude
Mtwara	Quantity	Lindi	Extraction Type
Kagera	Quantity	Dodoma	Quantity
Tanga	Longitude	Arusha	Longitude
Shinyanga	Quantity	Mbeya	Longitude
Tabora	Quantity	Singida	Age
Pwani	Funder 2	Morogoro	Longitude
Ruvuma	Quantity	Salaam	Quantity
Kilimanjaro	Longitude		

influential factor impacting water pump functionality, while longitude was the 2nd most important at 6 regions.

In the *Pwani* region, *funder* was the main factor as most pumps are funded by a private individual, and these pumps having a 90% functionality rate. Meanwhile, in the *Iringa* region, >80% of pumps had enough water with a very high functionality, while the *Ruvuma* and *Dodoma* regions had two times dry pumps than usual.

2) Which operators and/or management groups have the highest success rates in maintaining water pumps, and how do these rates vary based on factors which may make pump maintenance easier, such as water cost, pump type or location remoteness?: The *Water Board* management group has the best record of pump maintenance for their pumps, with 74.7% of their 2700 water pumps being functional, while the *WUA*, *Private Operators*, and *Trusts* followed with 69.2%, 68.6% and 63.9% functional pumps, respectively. *SWC* had the worst track record, with only 20.6% of their 97 pumps being functional.

3) What are the interactions between different features, such as water quantity and pump type, which could provide insights into the underlying causes of pump failure?: Most pump types have a good functionality rate except for ”motorpumps”. ”wind-powered” pumps don’t have a great performance, but there are 117 of them. ”other” pumps are likely to have bad functionality, and make up 10% of all pumps.

Another highly impactful feature is water *quantity*. The most influential feature on badly performing pump extraction types are dry pumps. Extraction types with good functionality have 58% ’enough’ *quantity* and 8% ’dry’ *quantity*, while the ones with bad functionality have 44% ’enough’ *quantity* and 23.4% ’dry’ *quantity*.

4) How does the age of a water pump relate to its functionality, and is there a point at which pumps become significantly more likely to break down or require replacement?: We found that as pumps aged, they became increasingly likely to become non-functional. There was no specific point in time that pumps began to fail more often, however, on average, pumps operating for longer than 15 years had >50% functionality.

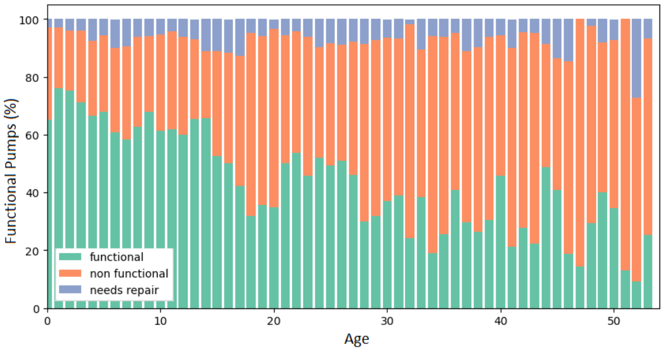

Fig. 5. Proportion of functional water pumps by Age.

B. Results Discussion

We were able to generate strong results based on our wide array of data imputation, feature engineering, feature selection and modelling approaches. By building on the results of Brenda Loznik [16], we were able to generate very strong model performances, exceeding that of her original solutions.

Through extensive feature selection testing, we found that introducing 3 of our 4 new data imputation approaches resulted in the best performance, with new imputation methods being used for *latitude*, *gps_height* and *funder*, outperforming the previous approach. We found that

We experiment with the utilisation various new feature selection approaches. Of our new approaches, FCBF had the worst performance with an F1 score of 76.78%. This was due to its overly aggressive feature removal, which resulted in the loss of a lot of information, making it an unsuitable method for this dataset. Our combined ICVEFE and ICVOFE approach, which is based around the comparison of engineered features to their originals, had the best performance.

Our best model performance was achieved using a Stacking classifier, with a F1 score of 82.10%, and an overall accuracy of 82.44%. This model approach was originally implemented by Loznik, however, we found that by introducing oversampling using SMOTE; applying various feature normalisation approaches; ICVEFE and ICVOFE feature selection; our added engineered features, and the improved imputation approaches, helped to raise the score of this classifier.

We propose a total of 7 new models in addition to the models initially implemented by Loznik, and evaluate each of them on the entire pipeline. We found that of our newly proposed approaches, the XGBClassifier Ensemble had the best performance, with an F1 score of 81.88, followed by GradientBoostingClassifier, which achieved an F1 score of 80.95%. The AdaBoost implementation had the worst performance, with an F1 score of 66.42%.

VII. Conclusion

In this work we were able to produce several high-performing solutions to water pump functionality prediction, using a variety of configurations. We provided insightful analysis to help the Government of Tanzania tackle the problem with water pumps functionality leading to faster repairs and improved water availability.

Our contributions to data imputation, feature extraction, feature selection and normalisation enhanced the quality of predictions. Oversampling helped to overcome the issue of low accuracy rate for underrepresented *functional needs repair* class. We provided results for a diverse model selection with many of them mentioned in the context of this issue for the first time. During our literature review, a research gap was identified into the use of dimensionality reduction methods. We were able to address this by providing implementation of TSVD.

We have identified several possible areas for future research work resulting from the limitations of this study. Our final result showed high difference between train and test sets results. Further research of this problem can focus on reducing overfitting of our approach. Limited by our resources, we didn't test different model configurations for the Stacking Classifier and Weighted Vote. Since these are the best performing models, this could be further investigated using Gradient Boosting Classifier, which showed high accuracy. The use of several other methods such as autoencoders, PCA and LDA were not implemented and could be further explored.

Contributions

Our individual contributions are highlighted at the top of our notebook solution. We feel that we both worked well, each implementing new approaches to data imputation, feature engineering, and feature selection, as well as several new models each. We also performed pair programming in the later stages of the project.

References

[1] "Predicting the status of water pumps using data mining approach," in *Darmatasia, Arymurthy)*, 2016.

[2] G. Bejarano, M. Jain, A. Ramesh, A. Seetharam, and A. Mishra, "Predictive analytics for smart water management in developing regions," in *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2018.

[3] W. Liu, B. Wang, and Z. Song, "Failure prediction of municipal water pipes using machine learning algorithms," *Water Resources Management*, vol. 36, no. 4, pp. 1271–1285, 2022.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[5] H. D. Tran and A. W. M. Ng, *Classifying Structural Condition of Deteriorating Stormwater Pipes Using Support Vector Machine*, pp. 857–866.

[6] S. Bedi, A. Samal, C. Ray, and D. Snow, "Comparative evaluation of machine learning models for groundwater quality assessment," *Environmental Monitoring and Assessment*, vol. 192, no. 12, p. 776, 2020.

[7] R. Jafar, I. Shahrour, and I. Juran, "Application of artificial neural networks (ann) to model the failure of urban water mains," *Mathematical and Computer Modelling*, vol. 51, no. 9, pp. 1170–1180, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895717710000051

[8] S. K. Hanoon, A. F. Abdullah, H. Z. M. Shafri, and A. Wayayok, "A novel approach based on machine learning and public engagement to predict water-scarcity risk in urban areas," *ISPRS International Journal of Geo-Information*, vol. 11, no. 12, 2022. [Online]. Available: https://www.mdpi.com/2220-9964/11/12/606

[9] G. Herrera and P. Morillo, "Benchmarking of supervised machine learning algorithms in the early failure prediction of a water pumping system," in *International Conference on Computational Science and Its Applications*. Springer, 2021, pp. 628–641.

[10] X. Fan, X. Wang, X. Zhang, and P. ASCE Xiong (Bill) Yu, "Machine learning based water pipe failure prediction: The effects of engineering, geology, climate and socio-economic factors," *Reliability Engineering System Safety*, vol. 219, p. 108185, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832021006694

[11] R. Harvey and E. McBean, "Understanding stormwater pipe deterioration through data mining," *Journal of Water Management Modeling*, 2014. [Online]. Available: https://www.chijournal.org/C374

[12] R. R. Harvey and E. A. McBean, "Comparing the utility of decision trees and support vector machines when planning inspections of linear sewer infrastructure," *Journal of Hydroinformatics*, vol. 16, no. 6, pp. 1265–1279, 05 2014. [Online]. Available: https://doi.org/10.2166/hydro.2014.007

[13] X. Deng, A. Ye, J. Zhong, D. Xu, W. Yang, Z. Song, Z. Zhang, J. Guo, T. Wang, Y. Tian, H. Pan, Z. Zhang, H. Wang, C. Wu, J. Shao, and X. Chen, "Bagging–xgboost algorithm based extreme weather identification and short-term load forecasting model," *Energy Reports*, vol. 8, pp. 8661–8674, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484722012124

[14] S. O. Moepya, S. S. Akhoury, and F. V. Nelwamondo, "Applying cost-sensitive classification for financial fraud detection under high class-imbalance," vol. 2015-January. IEEE Computer Society, 1 2015, pp. 183–192.

[15] W. Xu, S. Fan, C. Wang, J. Wu, Y. Yao, and J. Wu, "Leakage identification in water pipes using explainable ensemble tree model of vibration signals," *Measurement : journal of the International Measurement Confederation*, vol. 194, p. 110996, 2022.

[16] B. Loznik, "Brendaloznik - pump it up solution." [Online]. Available: https://github.com/BrendaLoznik/waterpumps

[17] D. Korosuo, E. Anyamba, J. Eastman, and J. Small, "Rainfall variability in northern tanzania in the march-may season long rains and its links to large-scale climate forcing," *International Journal of Climatology*, vol. 33, no. 2, pp. 306–318, 2013.

[18] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," *Proceedings of the 20th international conference on machine learning (ICML-03)*, vol. 3, pp. 856–863, 2003.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *arXiv preprint arXiv:1603.02754*, 2016.

[21] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," *arXiv preprint arXiv:1810.11363*, 2018.

[22] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[23] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[24] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[25] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[26] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[27] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

TABLE V
HYPERPARAMETERS SELECTED FOR EACH MODEL

Model	**Param 1**	**Param 2**	**Param 3**	**Param 4**
Random Forest	max_depth	n_estimators	min_samples_split	max_features
	30	150	8	log2
XGBoost	max_depth	n_estimators	colsample_bytree	eta
	12	100	0.3	0.15
CatBoost	max_depth	rsm	iterations	
	9	0.2	1000	
Bagging	max_features	n_estimators		
	0.4	1000		
XGB Ensemble	max_depth	n_estimators	colsample_bytree	eta
	12	100	0.3	0.15
LightGBM	max_depth	bagging_fraction	num_iterations	
	9	0.3	1000	
Gradient Boosting	max_depth	n_estimators	learning_rate	
	9	1000	0.01	
AdaBoost	learning_rate	n_estimators		
	0.1	0.1		
Extra Trees	max_depth	n_estimators	min_samples_split	
	9	200	4	
Neural Network	hidden_layer_sizes	activation	solver	learning_rate
	(100, 50)	relu	adam	constant
Stacking	model1	model2	model3	model4
	RF	XGB	Cat	Bag
Weighted Vote	model1:weight	model1:weight	model1:weight	model1:weight
	RF:0.861	XGB:0.78	Cat:0.706	Bag:0.751