Multi-Objective Multi-Gene Genetic Programming for the Prediction of Leakage in Water Distribution Networks

Matthew Hayslep*
University of Exeter
Exeter, United Kingdom
mh989@exeter.ac.uk

Edward Keedwell
University of Exeter
Exeter, United Kingdom

Raziyeh Farmani
University of Exeter
Exeter, United Kingdom

ABSTRACT

Understanding leakage is an important challenge within the water sector to minimise waste, energy use and carbon emissions. Every Water Distribution Network (WDN) has leakage, usually approximated as Minimum Night Flow (MNF) for each District Metered Area (DMA). However, not all DMAs have instruments to monitor leakage directly, or the main dynamic factors that contribute to it. Therefore, this article will estimate the leakage of a DMA by using the recorded features of its pipes, making use of readily available asset data collected routinely by water companies. This article interprets this problem as a feature construction task and uses a multi-objective multi-gene strongly typed genetic programming approach to create a set of features. These features are used by a linear regression model to estimate the average long-term leakage in DMAs and Shapley values are used to understand the impact and importance of each tree. The methodology is applied to a dataset for a real-world WDN with over 700 DMAs and the results are compared to a previous work which used human-constructed features. The results show comparable performance with significantly fewer, and less complex features. In addition, novel features are found that were not part of the human-constructed features.

CCS CONCEPTS

• **Computing methodologies → Genetic programming**; *Supervised learning by regression*; Cross-validation; • **Applied computing → Physical sciences and engineering**.

KEYWORDS

Feature construction, genetic programming, minimum night flow, leakage, linear regression

ACM Reference Format:
Matthew Hayslep, Edward Keedwell, and Raziyeh Farmani. 2023. Multi-Objective Multi-Gene Genetic Programming for the Prediction of Leakage in Water Distribution Networks. In *Genetic and Evolutionary Computation Conference (GECCO '23), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3583131.3590499

*Corresponding author.

GECCO '23, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0119-1/23/07.
https://doi.org/10.1145/3583131.3590499

1 INTRODUCTION

Leakage within Water Distribution Networks (WDNs) is a pervasive and important challenge for water companies which has an economic, environmental and sustainability impact [28]. In the UK alone around 3bn litres of water are lost to leaks every day [27]. These losses of clean water increase pumping and treatment costs for companies, and therefore the costs for consumers, as well as energy consumption and CO2 emissions. Minimum Night Flow (MNF), which is the lowest flow rate during the night (usually between 12am-4am), is a common method of estimating and analysing leakage in WDNs [5]. Flow is measured using meters at all entry and exit points of smaller sub-networks, forming what are called District Metered Areas (DMAs). The sum of all meter measurements, where negative if water is leaving the DMA and positive if water is entering, is the total demand or consumption of the DMA. Using the MNF as an approximation of leakage is based on the assumption that water usage is lowest at night, thus, most of the water use measured during this period is leakage.

Leakage is commonly separated into bursts and reported leakage – large events which have interrupted service or where water has come to the surface –, and unreported and background leakage – smaller numerous weeps, seeps and leaks in the infrastructure which are very difficult to detect [7]. Unreported and background leakage can lead to bursts due to weather changes and abnormal pressure events, and small leaks can grow over time due to wear [28]. Therefore, background and unreported leakage, when ignored, can lead to further issues.

The focus of this article is on long-term unreported and background leakage. This is measured in this article by taking the lowest MNF during a week (Weekly MNF). This filters out the effects of any bursts or reported leaks that last less than a week. This is averaged over a long period of time to create the Average of the Weekly MNF (AWM) [12]. This gives a measure of the long-term unreported and background leakage of a DMA, and is the target variable for this regression task. The aim of this work is to create a model that can predict the AWM of a DMA solely based on its physical infrastructure. The methodology is trained and tested on real-world DMAs from a WDN, where the AWM can be observed. However, the final model and features presented below can estimate the AWM of theoretically any DMA using **only** knowledge of the characteristics of the pipes within it. Therefore, the trained model can be used to predict the leakage of meterless networks.

In real-world DMAs there are numerous factors that effect the total amount of leakage. Pressure, pipe age, pipe material, weather conditions, time of year, and pipe condition are some commonly described factors amongst many others [28]. Some of these factors mainly concern bursts and the extent to which they affect background and unreported leakage is uncertain. In this article, only the physical properties of pipes are considered. This constitutes the simplest version of the problem and is reliant on data that is widely available for water companies. The accuracy of the final model will tell us how much of the variation in leakage between different DMAs can be solely attributed to the physical properties of the pipes.

In this problem, a DMA is defined by its pipes. Each pipe can only be part of one DMA at a time and is defined by five variables: length, diameter, age, volume, and material. Volume is calculated from the length and diameter, based on the assumption that the pipes are cylinders. The material of a pipe is grouped into one of three categories: metal, plastic, and other. Every DMA can contain any number of pipes; in practice this can range from the hundreds to several thousand. Therefore, a method is needed which can take a variably sized set of pipes as input. A DMA, $\mathbf{X}$, can be given as a matrix:

$$\begin{aligned}\mathbf{X} &= \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ \dots & \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & x_{m3} & x_{m4} & x_{m5} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{X}_{length} & \mathbf{X}_{diameter} & \mathbf{X}_{age} & \mathbf{X}_{volume} & \mathbf{X}_{material} \end{bmatrix} \end{aligned} \quad (1)$$

where the columns of $\mathbf{X}$ represent the different pipe variables and the value of m represents the number of pipes and varies for every DMA, i.e. the size of $\mathbf{X}$ varies between DMAs.

This paper is organised as follows: In Section 2 a brief literature review is given, focusing on methodologies for similar problems. In Section 3 the methodology used in this article is described and explained. Section 4 presents the results of the evolutionary process and the best solution found. Finally, Section 5 discusses the results and Shapley values are used as part of an analysis of the individual trees.

2 BACKGROUND AND RELATED WORK

As described above, the problem presented here requires a method which can handle variably sized matrices or sets of objects. Few methods can do this directly, instead needing this data to be summarised into a number of features. First, methods which can be used to summarise data in this way are discussed. This is followed by a discussion of graph-based methods as an alternative approach. Finally, an overview of other leakage prediction work is given.

2.1 Genetic Programming

Genetic Programming (GP) is a flexible methodology that can evolve computer programs, mathematical formulae, rule-based systems and other types of expressions. Encoding solutions as trees allows GP to solve a wide variety of problems. Multi-Gene Genetic Programming (MGGP) [30], also called multi-tree genetic programming, differs from standard GP by changing the representation from a single tree to a set of trees. GP has been used for feature construction with classification [3, 10, 32, 33]. However, this tends to be the construction of new features from existing ones, rather than the construction of features from a matrix of data for each sample, as is required here.

The use of GP for feature extraction on images may be better aligned with the requirements of this approch, as images are generally processed as matrices of pixel values. In [31] multi-objective GP is used to perform feature construction on images for classification. In this work a single GP tree is evolved, where each subtree of the root node is a separate feature. The authors used an SVM to do the final classification, with the resulting error rate as one objective and the number of nodes as the other objective. In [4] an additional level of complexity is added, where the classifier is also evolved along with the set of features. Both works found good results, but with relatively complex trees.

Evolutionary Polynomial Regression (EPR) [9] is a method which combines some of the advantages of genetic programming and numerical regression. The method is capable of finding polynomial formulae of various levels of complexity, and has seen numerous applications in the field of hydroinformatics. EPR employs a genetic algorithm to find a matrix of exponents that define a set of expressions. These expressions form the variables for a linear model. However, EPR is not applicable to this problem directly for several reasons. Firstly, the matrix of exponents that it considers must be applied to numerical data, and would require an encoding of categorical variables such as material. Secondly, the matrix of exponents is of a fixed size. EPR could be modified such that the matrix of exponents and the operators are appropriate for vectors, since this problem has a fixed number of columns in the matrix. However, such a modification is outside of the scope of this article.

2.2 Graph Regression

DMAs can be represented as graphs since they are networks of connected physical pipes [13]. Therefore, graph regression techniques such as those common to chemical analysis [11] are applicable to this problem. In other fields, graph kernels, random walks, and graph convolutional neural networks have been applied to a variety of problems [21]. In water systems, graph-based approaches have primarily been used to assess resilience [8, 13], model DMAs [19], or to divide WDNs into DMAs to reduce and improve the monitoring of leakage [24, 25]. Graph-based representation of water networks have been used to detect or locate leakage [1, 29] or reconstruct pressure for leak detection [35]. However, these methods are often restricted to simulated systems. In addition, as far as the authors are aware, graph regression methods have not been directly applied to predicting leakage in DMAs.

2.3 Leakage Prediction

The prediction of leakage levels or similar metrics will often include data on some of the other important factors previously mentioned: pressure, weather, time of year, etc. The rate of leakage is estimated in [15–17] in a case study of over 150 DMAs from a city in South Korea. These studies found that pipe length was the most important factor by far. However, the overall accuracy of the models were low, potentially due to the limited number of samples. Leakage rates were also predicted in [20] with more success. These studies primarily used neural networks to predict these values. The main disadvantage with this method is its black-box nature which can make these methods less applicable in the real-world where decisions are required to be explained.

Several other studies have predicted leakage, bursts, or pipe risk. For example, demand characteristics have been used to assess background leakage levels in a quantitative analysis [34]. An aggregation of the physical properties of pipes was used in [18] to predict pipe bursts using EPR. The optimal time to replace pipes was estimated in [14] based on physical properties and pressure using a different neural network for each pipe material. Finally, in [12], the authors used a set of human-defined features to predict AWM for real-world DMAs. One of the aims of the current article is to produce a simpler set of features with similar, or better accuracy.

3 MULTI-OBJECTIVE MULTI-GENE GP

In this article, an approach using multi-objective multi-gene genetic programming is taken. In multi-gene genetic programming (MGGP) [30] each individual consists of several genes, where each gene is a separate tree. The output of this model is the weighted linear combination of the outputs of each tree. Ridge linear regression is used to determine the weights for each tree. The resulting model takes the form of Equation (2):

$$y = c_0 + \sum_{j=1}^{m} c_j \cdot f_j(\mathbf{X}) \tag{2}$$

where y is the estimate of the target variable (AWM in this article), c_0 is the bias, m is the number of trees in an individual, f_j is the function resulting from the jth tree of an individual, and $\mathbf{X}$ is a matrix representation of a DMA.

Multi-gene genetic programming was chosen over 'single gene' genetic programming because the aim of the work is to find a *set* of features, which together can be used by a linear model. A single complex tree is undesirable in this application because explainability is important. Therefore, a method is needed to create multiple simple trees. 'Single gene' genetic programming could still be used in this context. However, explicit mechanisms for forcing diversity would be needed to prevent the population from converging on the same feature. Multi-gene genetic programming circumvents this problem.

In MGGP, crossover has two modes of operation: 'high level' and 'low level'. In high level crossover, whole trees are exchanged between individuals similarly to uniform crossover. In low level crossover, random trees from each parent are selected for GP subtree crossover [30]. High and low level crossover are chosen at random with an equal probability. During mutation, there is a uniform probability for each gene to undergo point mutation, where a random subtree is replaced by a newly generated random subtree.

The desired output for this application is a single scalar value from each tree, but the input is a matrix (the collection of pipes and their features). Therefore, the strongly typed genetic programming [23] method is applied, which enforces data-type constraints on the functions (primitives) that make up the trees. This not only reduces the size of the search space [23], but ensures the trees are valid expressions. Figure 1 shows the different data types used, and how they relate to each other through the different primitives.

The multi-objective evolutionary algorithm used is NSGA-II [2] which uses binary tournament selection for crossover, and Pareto dominance and crowding distance to select individuals for the next generation. The particular implementation of the methodology in this article was done using DEAP [6].

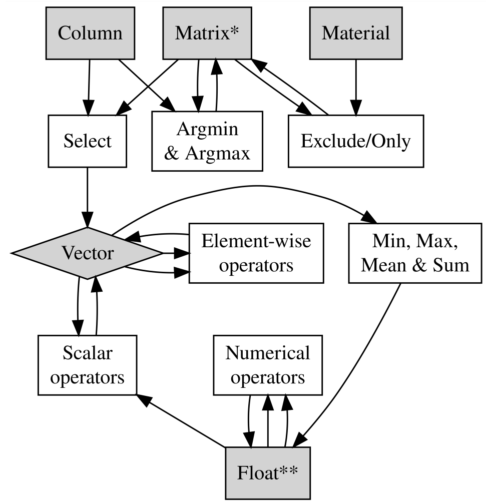


Figure 1: Graph showing data types and how the primitives transition between them. The arrows show the input and output data types for the different primitives. Types are shown in grey: a box indicates this type can be provided by a terminal while a diamond indicates that it cannot. Primitives are shown in white boxes. The tree argument/input type is shown by *. The tree output type is shown by **.

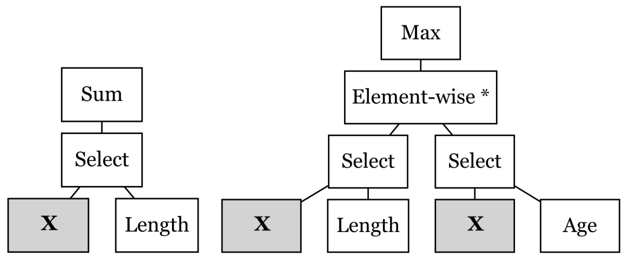
3.1 Objectives

During each fitness evaluation, k-fold cross validation is performed on the training set. The training set is made up of 70% of the DMAs in the dataset, which are randomly selected. During k-fold cross validation, the training data is split into k evenly sized segments or folds. Each fold is used as a validation set to test the linear model, which is trained on the remaining folds. The validation R^2 of the model is recorded for each fold. Five folds ($k = 5$) are used and the mean R^2 over all folds is used as an objective. Two objectives are defined:

$$\begin{aligned} \text{To maximise:} \quad & g_1(\mathbf{Z}) = \frac{\sum_{i=1}^{k} R_i^2}{k} \\ \text{To minimise:} \quad & g_2(\mathbf{Z}) = \sum_{j=1}^{m} |\mathbf{Z}_j| \end{aligned} \tag{3}$$

where m is the number of trees in an individual $\mathbf{Z}$, $|\mathbf{Z}_j|$ is the number of nodes in the jth tree, k is the number of k-folds in the cross-validation, and R_i^2 is the validation R^2 of the ith fold. Therefore, g_1 is the *validation* R^2 and g_2 is the total tree size.

The test set, made of the remaining 30% of DMAs, is held back to evaluate the final models. Once the evolutionary process is complete, each individual on the Pareto front is evaluated by training a new linear model on the evolved features using the entire training set. The R^2 of the models on the test set (test R^2) indicates the true fitness of the evolved features and is used to evaluate the solutions.



(a) Sum of pipe length (b) Max of pipe length multiplied by age

Figure 2: Two example trees. These trees can be expressed as Equation (4) and (5) respectively.

3.2 Primitives

As shown in Figure 1, the 'Select' primitive takes as input a matrix and a column value, indicating which column of the matrix to select (age, diameter, etc.), then outputs a vector. This primitive forms a bottleneck between the input type (matrix) and output type (float) of the trees. Two primitives are used to filter the matrix by material. The 'Exclude' primitive is used to filter out a particular material, while the 'Only' primitive is used to filter out every other material. These primitives take as input a matrix and a material value (metal, plastic, etc.) and output a new matrix filtered by that material. The relationships between each data type and all the primitives is shown in Figure 1. There are two numerical terminals; π and e, and two ephemeral constants; any real number in $[-1, 1]$, and any integer number in $[-10, 10]$. The use of strongly typed GP results in overlap in function between some of the primitives. For example, multiplication, division, subtraction and addition each appear as element-wise operators (for two vectors), scalar operators (for a vector and a real number), and numerical operators (for two real numbers).

Two example trees are shown in Figure 2. Example tree (a) can be expressed as:

$$f(\mathbf{X}) = \sum_{i=1}^{m} \mathbf{X}_{i,length} \tag{4}$$

where $\mathbf{X}$ is a DMA, i.e. a set of pipes. Similarly, example tree (b) can be expressed as:

$$f(\mathbf{X}) = \max_{1 \le i \le n} (\mathbf{X}_{i,length} \cdot \mathbf{X}_{i,age}) \tag{5}$$

Example tree (a) shows a minimally sized 'active' tree, with four total nodes. The smallest possible tree is a numerical terminal or ephemeral constant, which constitutes an 'inactive' tree, meaning it does not respond to changes in input, and can therefore be ignored. This allows an individual in the population to effectively have fewer features than the number of genes it has. By optimising along the total size of all trees, as per Equation (3), it should be expected that individuals with a range of solutions with a different number of active trees will be seen. The minimum depth of an active tree is the shortest path on the graph shown in Figure 1, from the input type to the output type. In this case the minimum depth is three.

When generating a random tree, there are many more potential primitives which transform from a vector into another vector than there are to transform from a vector to a float, as can be seen in Figure 1. This can result in significant bloat in the best case or an infinite loop of branching primitives that never finishes in the worst case. To solve this problem, Dijkstra's algorithm is used to find the shortest path to the nearest terminal. Each primitive is represented as an edge in a directed multigraph, similar to Figure 1. The cost of edges related to the same primitive are equal, and primitives which lead to duplicate inputs of the same type are heavily penalised. Once a random tree has depth greater than or equal to a max depth, the graph-based algorithm is used to add primitives or terminals to quickly complete the tree. This addition helps to control bloat within the algorithm.

4 RESULTS

The real-world dataset consists of 750 UK DMAs from a single WDN. Together, these DMAs represent over 11 million metres of pipe, ranging from dense urban networks to huge sparse rural networks. Generally, the DMAs in this dataset do not contain trunk mains, which distribute large amounts of water to numerous DMAs throughout a region. However, some DMAs in this dataset receive water from trunk mains but distribute a large proportion to other DMAs further downstream. Due to the commercially sensitive nature of the data used in this article, it cannot be made publicly available.

Historic Weekly MNF records cover 2018-2022 for each DMA. However, many DMAs have missing data. On average, each DMA is missing roughly 1.5% of its Weekly MNF records. All DMAs have at least half of their total records, but some only have a handful of records for certain years. However, every DMA has at least one Weekly MNF record for each year. The Average of the Weekly MNF (AWM) is simply the average of all Weekly MNFs for each DMA across the entire time period. As discussed in Section 1, this filters out the effects of any bursts or reported leaks that last less than a week to reflect background leakage rather than these unpredictable events. This focuses the problem onto background and unreported leakage. The AWM is the target variable that this method is going to predict.

The parameters of this experiment are shown in Table 1. These values were chosen after considering the literature referenced in this article and through limited trial and error. The number of genes was chosen based on the maximum number of features desired. The maximum depth of a random subtree was chosen based on the maximum shortest path from terminals to output type, as shown in Figure 1. The actual depth of a random subtree can be greater than this, see Section 3.2. However, no further optimisation of

Table 1: Parameters of the experiments.

Parameter	Value
Population	40
Number of genes	9
Max generations	400
Crossover probability	0.9
High-level crossover: Uniform probability	0.1
Mutation: Uniform probability	0.1
Random subtree: Max depth	4

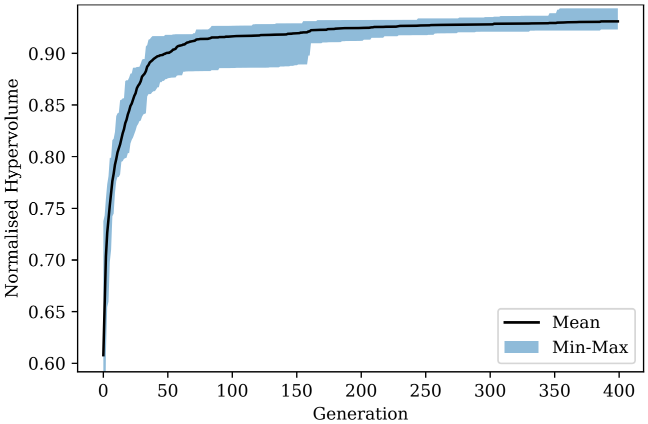


Figure 3: Normalised hypervolume of the populations of 11 runs over time. The nadir point for the normalised hypervolume is defined as $[0, 324]$, with $[0.7, 9]$ as the utopia.

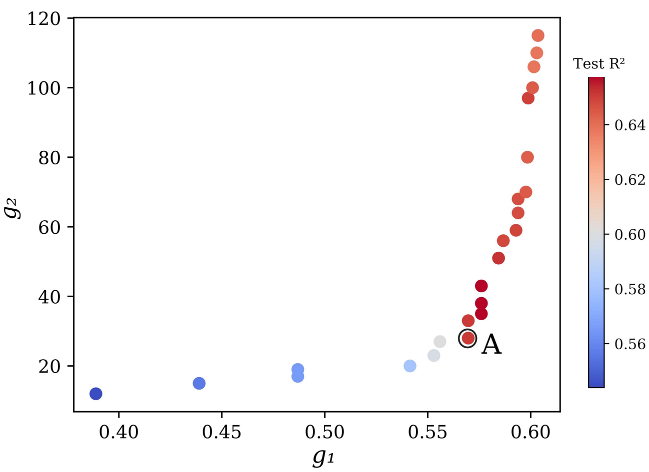


Figure 4: Final population positioned in objective space i.e. g_1 and g_2 from Equation (3). The colour (test R^2) is the performance of an evolved solution on the test set after evolution has finished. Not shown on the front is an individual with $g_1 \approx 0$ and $g_2 = 9$. At around $g_2 = 40$, test R^2 begins to decrease even as g_1 increases, likely due to overfitting. The 'best' solution that was selected for further analysis is labelled as A.

these parameters was undertaken due to computational resource limitations.

In total, 11 runs (varying the random seed across runs) of the methodology were undertaken to assess its validity. Figure 3 shows the mean normalised hypervolume of all populations over time, as well as the minimum and maximum at each generation. This number of runs was selected due to computational resource limitations. This figure indicates that the population has converged over the Pareto front. However, some improvements were found late in the evolutionary process by a single run. Extending the maximum number of generations was outside of the limits of the computational and time budget for this article. The utopia point was chosen to be $[0.7, 9]$ where; 9 is the smallest multi-gene individual possible, and, while 1.0 is the perfect R^2, this real-world data problem has a practical R^2 ceiling of 0.7. A single run near the mean normalised hypervolume was selected for further analysis.

Table 2: Table of results comparing the solutions from the final population and the approach presented in [12]. Complexity is measured in nodes, or an estimation of the number of nodes that would be needed for the previous work.

Model	Features	Complexity	Test R^2
A	5	28	0.651
Elastic Net [12]	24	>324	0.680

Figure 4 shows the Pareto front of the final population from the selected run. This figure demonstrates how increasing the size of the tree after a certain point leads to overfitting, as the test R^2 starts to decrease. The test R^2 is generally higher than the validation R^2 (g_1) because the final model is trained on more data (see Section 3.1). The 'best' solution, labelled A and taken from the 'knee-point' on the front, has a size (g_2) of 28 nodes, a validation R^2 (g_1) of 0.57, and a test R^2 of 0.651. This solution contains four inactive genes and, therefore, provides a set of five features. The solution contains three minimally sized active genes with four nodes each, and two genes with six nodes. Figure 5 shows the trees of this solution. These genes can be expressed as:

$$
\begin{aligned}
f_1(\mathbf{X}) &= \sum_{i=1}^{m} \mathbf{X}_{i,length} \\
f_2(\mathbf{X}) &= \sum_{i=1}^{m} \mathbf{S}_{i,age} \\
&\quad \text{where} \quad \mathbf{S} = \{x \in \mathbf{X}_{material} | x = Plastic\} \\
f_3(\mathbf{X}) &= \sum_{i=1}^{m} \mathbf{S}_{i,diameter} \\
&\quad \text{where} \quad \mathbf{S} = \{x \in \mathbf{X}_{material} | x \neq Metal\} \\
f_4(\mathbf{X}) &= \max(\mathbf{X}_{age}) \\
f_5(\mathbf{X}) &= \min(\mathbf{X}_{diameter})
\end{aligned}
\tag{6}
$$

where $\mathbf{X}$ is a DMA, and m is the number of pipes in $\mathbf{X}$.

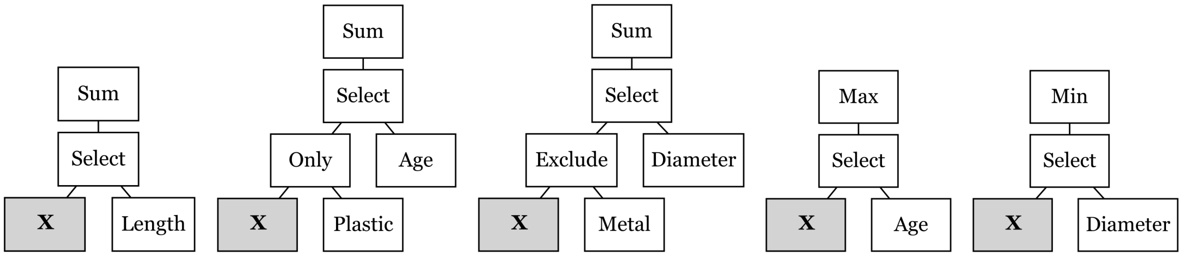
Table 2 compares the results for the best solution and the approach presented in [12]. The human-defined features described in [12] include equivalents of f_1, f_4, and f_5. The best solution presented here contains new novel features, i.e. f_2 and f_3. The linear model of the best solution, as per Equation 2 in expanded form, is as follows:

$$y = 1.41 + 2.12f_1 + 1.81f_2 + 1.34f_3 + 0.93f_4 - 0.70f_5 \tag{7}$$

where all features except f_5 have positive coefficients.

5 DISCUSSION

Shapley values can be used to measure the effect of including a feature on a model [22]. Machine learning models aim to minimise the error between their predictions and the observed values. As a result, the expected output of a model over an entire problem is usually very close to the mean of the observed values. Shapley



(a) f_1 **(b)** f_2 **(c)** f_3 **(d)** f_4 **(e)** f_5

Figure 5: The active trees of the best solution, labelled A in Figure 4. The remaining four genes were inactive, i.e. composed of numerical constants. These trees can be expressed as in Equations (6).

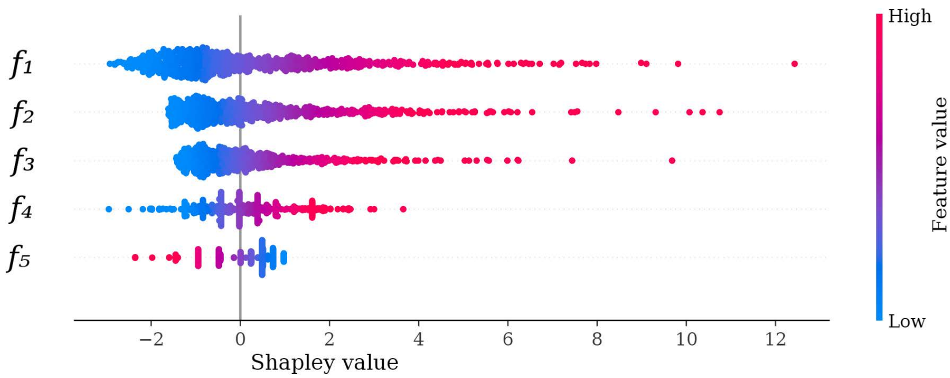


Figure 6: Summary plot of the features of the best solution. Each point is a DMA, the width indicates the density of points. The colour indicates whether the value of the feature is higher or lower. The Shapley value indicates how each feature changes the prediction with relation to the expected output of the model.

values describe, on a prediction by prediction level, the impact of including a feature. For a particular prediction, the Shapley value represents the deviation from the model's expected output that a feature causes. SHAP [22] is used to analytically calculate the Shapley values for the linear models used in this article. By using Shapley values, the individual contribution and impact of each tree, i.e. feature, can be assessed. This is shown in Figure 6 where the Shapley value indicates the impact of a feature with respect to the expected output of the model. This figure shows the features from Equation 6 sorted by total absolute Shapley value, with the highest-impact feature at the top.

Features f_4 and f_5 show distinct bands in their distribution. This is strongest in f_5 and is a result of how pipes come in discrete diameters with older pipes measured in inches and newer pipes measured in millimetres (all converted into mm in this dataset). For f_4 this suggests that many of the DMAs were constructed in waves of expansion, and that this is evident from the oldest pipes in each DMA.

Figure 6 shows that f_1, f_2, f_3, and possibly f_5, have distributions similar to a lognormal distribution. This is logical given that AWM is lognormally distributed as well. The coefficients for f_1, f_2, and f_3, as shown in Equation 7, are positive. However, for many DMAs the value of these features is small enough that they effectively decrease the predicted AWM from the average observed value. This is inverted for f_5 where most values are small, and tend to increase the predicted AWM from the average observed value. Although f_4 is clustered into distinct bands, this feature appears normally distributed with the mean value of f_4 causing no deviation in the expected output.

From Figure 6 and Equations (6), f_1 indicates that a DMA with more pipe has more leakage. This suggests that some part of background and unreported leakage is uniformly distributed across all pipes. The quantification of the amount of variation in leakage between DMAs which is attributable to this feature is discussed later in this section.

From Figure 6 and Equations (6), f_2 indicates that a higher gross age of all plastic pipes increases long-term background and unreported leakage. This suggests that plastic pipes may deteriorate more compared to other materials, and leads to an increase in the long-term background and unreported leakage over time. The impact of soil may be important to consider; the soil in the area from which the DMAs in this study originate is generally acidic. However, the installation of piping often includes a backfill material, such as gravel or sand, though the specifics of the backfill material could also be a cause of corrosion [26]. Alternatively, plastic pipes might degrade quicker due to other factors, such as ground movement, pressure, hydraulic transients or water quality, for example.

From Figure 6 and Equations (6), f_3 indicates that a higher gross diameter of all non-metal pipes increases long-term background and unreported leakage. This suggests that in general larger diameter pipes contribute more to long-term background and unreported leakage. This is an expected relationship. However, this feature filters out metal pipes, potentially suggesting that metal pipes contribute less in this fashion.

From Figure 6 and Equations (6), f_4 indicates that the age of the oldest pipe is indicative of long-term background and unreported leakage. This feature represents an expected relationship between overall age of infrastructure and its condition. Pipes will deteriorate over time and their propensity to leak will therefore increase. Features f_4 and f_2 are different in two important ways. First, f_2 only considers plastic pipes while f_4 considers all materials. Second, f_2 sums the ages while f_4 finds the maximum age. The presence of these features and their differences suggests that leakage does not increase linearly with pipe age. It also suggests that a set of old pipes may have the same impact on long-term background and unreported leakage regardless of the age of the rest of the network. This further suggests that metrics such as average age could be misleading when trying to understand leakage in DMAs.

From Figure 6 and Equations (6), f_5 indicates the minimum diameter of all pipes in a DMA increases long-term background and unreported leakage when this value is *smaller*. The minimum diameter of all pipes in a DMA may be indicative of other factors. In particular, pressure may be important. As high pressure water travels from a larger diameter pipe to a smaller diameter pipe, the pressure decreases and the velocity of the water increases. In order to service customers, water companies in the UK are required to keep water pressure above a threshold. Therefore, in order to maintain service level pressure across a DMA, there may be a higher overall pressure in DMAs where f_5 is small compared to where f_5 is larger. Alternatively, this may suggest that the smallest diameter pipes in DMAs may be more prone to leakage themselves, for example due to hydraulic transients causing more wear and tear over time, in part also because of the increased velocity.

In summary, the factors described by Equations (6) and their impacts as shown by Figure 6 are *explainable* and grounded in this real-world application. However, the Shapley values alone may not be sufficient to truly understand the importance of each feature in relation to each other. The Pareto front provides an alternative means of exploring the importance of each feature. Table 3 shows the consistent presence of the different features throughout the set of solutions that are smaller, in terms of g_2, than the best solution. Every solution contains f_1. Features f_2 and f_4 are also present in many of the solutions, indicating that these feature are important genes across the Pareto front. The smallest solution shown in Table 3 consists of only f_1. The R^2 of this smaller solution suggests that the uniform distribution of background and unreported leakage accounts for 54.4% of the variation of long-term leakage (measured by AWM) between DMAs. The combination of all five features accounts for 65.1% of the variation of long-term leakage between DMAs.

The solutions shown in Table 3 contain very little bloat. Largely, this will be due to the multi-objective optimisation. However, the cross-validation will reduce the overfitting on the training set, which may also reduce bloat to some degree. Additionally, the graph-based tree completion may also contribute to this by reducing the complexity of the randomly added trees. Solutions that are larger than those shown in Table 3 often contain some degree of bloat, for example a tree which multiplies two numerical constants together. This indicates that there would be some benefit to continuing the evolution beyond the number of generations shown here. However, as stated before, this was outside of limits of the computational resources and time available for this article.

Table 3: Shows the presence of Equations (6) in solutions of decreasing size along the Pareto front shown in Figure 4. The best solution (A in Figure 4) is shown in bold. The presence of a tree not part of Equations (6) is indicated by *

g_1	g_2	Trees	f_1	f_2	f_3	f_4	f_5	Test R^2
0.57	**28**	**5**	✓	✓	✓	✓	✓	**0.651**
0.56	27	4*	✓	✓		✓		0.601
0.55	23	4	✓	✓		✓	✓	0.598
0.54	20	3	✓	✓		✓		0.581
0.49	17	2	✓	✓				0.565
0.44	15	2	✓			✓		0.556
0.39	12	1	✓					0.544

6 CONCLUSION

This article has applied a novel multi-objective multi-gene genetic programming approach to the problem of feature construction for the prediction of long-term leakage quantities in water distribution networks. The resulting machine learning model performed well and was significantly less complex than the previous human-made model, both in terms of the number of features, and the complexity of those features. In addition, new features which were not previously considered were found. This demonstrates the value of using evolutionary approaches on real-world problems.

The results found five features which together accounted for 65% of the variation in long-term background and unreported leakage between different real-world DMAs. The individual impact of each feature was analysed using Shapley values. The derived model can accurately predict the average long term minimum night flow with low computational complexity for DMAs. Finally, this may allow companies to direct their maintenance efforts towards areas of the network that are likely to be the largest contributors towards consistent leakage.

ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support from South West Water in collaboration with the University of Exeter Centre for Resilience, Environment, Water and Waste (CREWW).

REFERENCES

[1] Antonio Candelieri, Dante Conti, and Francesco Archetti. 2014. A Graph based Analysis of Leak Localization in Urban Water Networks. *Procedia Engineering* 70 (Jan. 2014), 228–237. https://doi.org/10.1016/j.proeng.2014.02.026

[2] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (April 2002), 182–197. https://doi.org/10.1109/4235.996017

[3] Pedro G. Espejo, Sebastián Ventura, and Francisco Herrera. 2010. A Survey on the Application of Genetic Programming to Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40, 2 (March 2010), 121–144. https://doi.org/10.1109/TSMCC.2009.2033566

[4] Qinglan Fan, Ying Bi, Bing Xue, and Mengjie Zhang. 2022. Genetic programming for feature extraction and construction in image classification. *Applied Soft Computing* 118 (March 2022), 108509. https://doi.org/10.1016/j.asoc.2022.108509

[5] Malcolm Farley and Stuart Trow. 2005. *Losses in Water Distribution Networks: A Practitioners' Guide to Assessment, Monitoring and Control.* Vol. 4. IWA Publishing, London, UK. https://doi.org/10.2166/9781780402642

[6] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: evolutionary algorithms made easy. *The Journal of Machine Learning Research* 13, 1 (July 2012), 2171–2175.

[7] Vicente J. García, Enrique Cabrera, and Enrique Cabrera, Jr. 2006. The Minimum Night Flow Method Revisited. In *Water Distribution Systems Analysis Symposium 2006.* ASCE, Cincinnati, Ohio, United States, 1–18. https://doi.org/10.1061/40941(247)35

[8] Alireza Gheisi, Mark Forsyth, and Gholamreza Naser. 2016. Water Distribution Systems Reliability: A Review of Research Literature. *Journal of Water Resources Planning and Management* 142, 11 (Nov. 2016), 04016047. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000690

[9] Orazio Giustolisi and Dragan A. Savic. 2006. A symbolic data-driven technique based on evolutionary polynomial regression. *Journal of Hydroinformatics* 8, 3 (July 2006), 207–222. https://doi.org/10.2166/hydro.2006.020b

[10] Ling Guo, Daniel Rivero, Julián Dorado, Cristian R. Munteanu, and Alejandro Pazos. 2011. Automatic feature extraction using genetic programming: An application to epileptic EEG classification. *Expert Systems with Applications* 38, 8 (Aug. 2011), 10425–10436. https://doi.org/10.1016/j.eswa.2011.02.118

[11] William L Hamilton. 2020. Graph Representation Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14, 3 (2020), 1–159.

[12] Matthew Hayslep, Edward Keedwell, Farmani Raziyeh, and Joshua Pocock. 2023. Understanding district metered area level leakage using explainable machine learning. In *IOP Conference Series: Earth and Environmental Science*, Vol. 1136. IOP, Bucharest, Romania, 012040. https://doi.org/10.1088/1755-1315/1136/1/012040

[13] Manuel Herrera, Edo Abraham, and Ivan Stoianov. 2016. A Graph-Theoretic Framework for Assessing the Resilience of Sectorised Water Distribution Networks. *Water Resources Management* 30, 5 (March 2016), 1685–1699. https://doi.org/10.1007/s11269-016-1245-6

[14] Raed Jafar, Isam Shahrour, and Ilan Juran. 2010. Application of Artificial Neural Networks (ANN) to model the failure of urban water mains. *Mathematical and Computer Modelling* 51, 9 (May 2010), 1170–1180. https://doi.org/10.1016/j.mcm.2009.12.033

[15] Dongwoo Jang and Gyewoon Choi. 2017. Estimation of Non-Revenue Water Ratio for Sustainable Management Using Artificial Neural Network and Z-Score in Incheon, Republic of Korea. *Sustainability* 9, 11 (Nov. 2017), 1933. https://doi.org/10.3390/su9111933

[16] Dongwoo Jang and Gyewoon Choi. 2018. Estimation of Non-Revenue Water Ratio Using MRA and ANN in Water Distribution Networks. *Water* 10, 1 (Jan. 2018), 2. https://doi.org/10.3390/w10010002

[17] Dongwoo Jang, Hyoseon Park, and Gyewoon Choi. 2018. Estimation of Leakage Ratio Using Principal Component Analysis and Artificial Neural Network in Water Distribution Systems. *Sustainability* 10, 3 (March 2018), 750. https://doi.org/10.3390/su10030750

[18] Konstantinos Kakoudakis, Kourosh Behzadian, Raziyeh Farmani, and David Butler. 2017. Pipeline failure prediction in water distribution networks using evolutionary polynomial regression combined with K-means clustering. *Urban Water Journal* 14, 7 (Aug. 2017), 737–742. https://doi.org/10.1080/1573062X.2016.1253755

[19] Hiremagalur K. Kesavan and Muthu Chandrashekar. 1972. Graph-Theoretic Models for Pipe Network Analysis. *Journal of the Hydraulics Division* 98, 2 (Feb. 1972), 345–364. https://doi.org/10.1061/JYCEAJ.0003225

[20] Burak Kizilöz. 2021. Prediction model for the leakage rate in a water distribution system. *Water Supply* 21, 8 (June 2021), 4481–4492. https://doi.org/10.2166/ws.2021.194

[21] Nils M. Kriege, Fredrik D. Johansson, and Christopher Morris. 2020. A survey on graph kernels. *Applied Network Science* 5, 1 (Dec. 2020), 1–42. https://doi.org/10.1007/s41109-019-0195-3

[22] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc., Long Beach, CA, USA, 1–10. https://doi.org/10.5555/3295222.3295230

[23] David J. Montana. 1995. Strongly Typed Genetic Programming. *Evolutionary Computation* 3, 2 (June 1995), 199–230. https://doi.org/10.1162/evco.1995.3.2.199

[24] Tianwei Mu, Manhong Huang, Shi Tang, Rui Zhang, Gang Chen, and Baiyi Jiang. 2022. Sensor Partitioning Placements via Random Walk and Water Quality and Leakage Detection Models within Water Distribution Systems. *Water Resources Management* 36, 13 (Oct. 2022), 5297–5311. https://doi.org/10.1007/s11269-022-03312-z

[25] Tianwei Mu, Yan Lu, Haoqiang Tan, Haowen Zhang, and Chengzhi Zheng. 2021. Random Walks Partitioning and Network Reliability Assessing in Water Distribution System. *Water Resources Management* 35, 8 (June 2021), 2325–2341. https://doi.org/10.1007/s11269-021-02793-8

[26] D. Kelly O'Day. 1982. Organizing and analyzing leak and break data for making main replacement decisions. *American Water Works Association* 74, 11 (1982), 588–594. http://www.jstor.org/stable/41271411

[27] Ofwat. 2021. *Service and delivery report 2020-21.* Comparative reports and data. Ofwat. 29 pages. https://www.ofwat.gov.uk/publication/service-and-delivery-2020-21/

[28] Raido Puust, Zoran Kapelan, Dragan Savic, and Tiit Koppel. 2010. A review of methods for leakage management in pipe networks. *Urban Water Journal* 7, 1 (Feb. 2010), 25–45. https://doi.org/10.1080/15730621003610878

[29] Aravind Rajeswaran, Sridharakumar Narasimhan, and Shankar Narasimhan. 2018. A graph partitioning algorithm for leak detection in water distribution networks. *Computers & Chemical Engineering* 108 (Jan. 2018), 11–23. https://doi.org/10.1016/j.compchemeng.2017.08.007

[30] Domonic Searson, David Leahy, and Mark Willis. 2010. GPTIPS: An Open Source Genetic Programming Toolbox For Multigene Symbolic Regression. In *International MultiConference of Engineers and Computer Scientists 2010 (Lecture Notes in Engineering and Computer Science, Vol. 1).* Newswood Ltd, Hong Kong, 77–80.

[31] Ling Shao, Li Liu, and Xuelong Li. 2014. Feature Learning for Image Classification Via Multiobjective Genetic Programming. *IEEE Transactions on Neural Networks and Learning Systems* 25, 7 (July 2014), 1359–1371. https://doi.org/10.1109/TNNLS.2013.2293418

[32] Matthew Smith and Larry Bull. 2007. Improving the human readability of features constructed by genetic programming. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation (GECCO '07).* Association for Computing Machinery, New York, NY, USA, 1694–1701. https://doi.org/10.1145/1276958.1277291

[33] Matthew G. Smith and Larry Bull. 2005. Genetic Programming with a Genetic Algorithm for Feature Construction and Selection. *Genetic Programming and Evolvable Machines* 6, 3 (Sept. 2005), 265–281. https://doi.org/10.1007/s10710-005-2988-7

[34] Peter van Thienen. 2022. Direct assessment of background leakage levels for individual district metered areas (DMAs) using correspondence of demand characteristics between DMAs. *Water Supply* 22, 7 (July 2022), 6370–6388. https://doi.org/10.2166/ws.2022.251

[35] Garðar Örn Garðarsson, Francesca Boem, and Laura Toni. 2022. Graph-Based Learning for Leak Detection and Localisation in Water Distribution Networks*. *IFAC-PapersOnLine* 55, 6 (Jan. 2022), 661–666. https://doi.org/10.1016/j.ifacol.2022.07.203