University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Data Science

Applying Machine Learning Models for Predicting Stream Network Dynamics

Supervisor
Prof. Nicolo Navarin
University of Padova

Co-supervisor
Prof. Gianluca Botter
University of Padova

Master Candidate
Sara Kartalovic

Student ID
2009468

Academic Year
2022-2023

"Today's scientists have substituted mathematics for experiments, and they wander off through equation after equation, and eventually build a structure which has no relation to reality. "
— Nikola Tesla

Abstract

The streams provide numerous benefits, some of which are maintaining the quality and quantity of drinking water, filtering pollutants, supplying food and providing habitat for wildlife and plants, and flood protection. The main characteristic of intermittent streams is the water flow during certain times of the year when groundwater and runoff from precipitation or snowmelt provide water for streamflow. Between July 2018 and October 2021, the study catchment Valfredda was monitored and the spatio-temporal dynamics of the active river network were observed on 30 occasions. In this study, climatic and geomorphic datasets and machine learning are used to predict the dynamics of intermittent streams along the Valfredda river in northern Italy. The prediction is made by performing the binary classification of the node's state where various sets of features are explored in order to determine the measurable characterization of the fundamental causes and effects. Different time ranges were used to test the sensitivity of the nodes and the influence of time-series predictors. Machine learning classification algorithms logistic regression, decision tree, random forest, k-nearest neighbors, and support vector machine were evaluated using various metrics in order to select the best model. The classifiers were able to perform well across all versions of the dataset when historical information was included, but the weekly and biweekly approximations of the weather data were proven to be the best choice for accurate prediction because most of the best models were trained using this data. The weekly and biweekly approximation of all weather data features were calculated by taking the average of historical data 7 and 14 days before the date of observation respectively. On the other hand, when all data was included in the training of the model, the best models in both experiments were daily logistic regression models. In this case, the weather features included the values from one day before the observation day. The models suggest that spatio-temporal relations are crucial in prediction, and features such as weighted averages of the current and previous state of the node have a significant role in producing the correct output. Additionally, local persistency is essential for the majority of the models' good performance.

Contents

Listing of figures

Listing of tables

Listing of acronyms

ML ..... Machine Learning

ARPAV ..... Veneto Region Environmental Protection Agency

ID ..... Identification Number

LR ..... Logistic Regression

DT ..... Decision Tree

RF ..... Random Forest

KNN ..... k-Nearest Neighbors

SVM ..... Support Vector Machine

TP ..... True Positive

TN ..... True Negative

FP ..... False Positive

FN ..... False Negative

ROC ..... Receiver Operating Characteristic

AUC ..... Area Under Curve

TPR ..... True Positive Rate

ACC ..... Accuracy

PREC ..... Precision

REC ..... Recall

1 Introduction

Temporary streams serving as animal and plant habitat, zones of nutrient and carbon processing, and connectivity corridors intimately linked to both the watersheds they drain and the river networks to which they are episodically connected [1]. They are defined by periodic flow cessation, and may experience partial or complete loss of surface water. One of the key distinctions between intermittence regimes is predictability: some systems experience predictable flow cessation or drying during summer or dry seasons, whereas unpredictable wet-dry cycles characterize other streams. [2] Intermittent streams flow seasonally in response to snowmelt and/or elevated groundwater tables resulting from increased periods of precipitation and/or decreased evapotranspiration [1]. These features along with network characteristics are going to be exploited in order to predict the dynamics of a stream network. The probabilistic approaches were applied [3] [4], but even though Machine Learning(ML) application in hydrology is vast[5][6], there is no unique general model suited for this prediction. An important task in many scientific fields is the prediction of a response variable based on a set of predictor variables. In many situations though, the aim is not only to make the most accurate predictions of the response but also to identify which predictor variables are the most important to make these predictions, e.g. in order to understand the underlying process. [7] With this being said, different feature combinations are going to be explored and modeled using various machine learning algorithms. Each combination of node features was coupled with each approximation of the weather data considering each year separately. The different performance metrics measure different tradeoffs in the predictions made by a classifier, and it is possible for learning methods to perform well on one metric, but be suboptimal on other metrics. [8] Because of this, the machine learning models are going to be evaluated using several metrics measures to ensure the performance is optimal given the data. It will present how important temporal and spatial correlations are in predicting the dynamics of stream flow.

The study catchment Rio Valfredda is located in Northern Italy and is characterized by main Alpine weather features, high precipitation throughout the year major amount of snow falling during winter. The stream network was mapped by 509 nodes and 476 edges. This research focuses on the prediction and classification of the state of the node (wet or dry) while at the same time evaluating the influence spatio-temporal correlation of the

network and weather data have on the model and its performance. The topics of the thesis are delivered as follows: the study catchment along with the description, cleaning, preprocessing, feature engineering, and analysis of the datasets are described in chapter 2. The Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), k-Nearest Neighbors (KNN), and Support Vector Machine (SVM) algorithms explained in the chapter 3 as well as the experiments and methods of the model evaluation applied in the research. Lastly, chapter 4 include the performance evaluation of all the models and deeper analysis of the models with higher achievement.

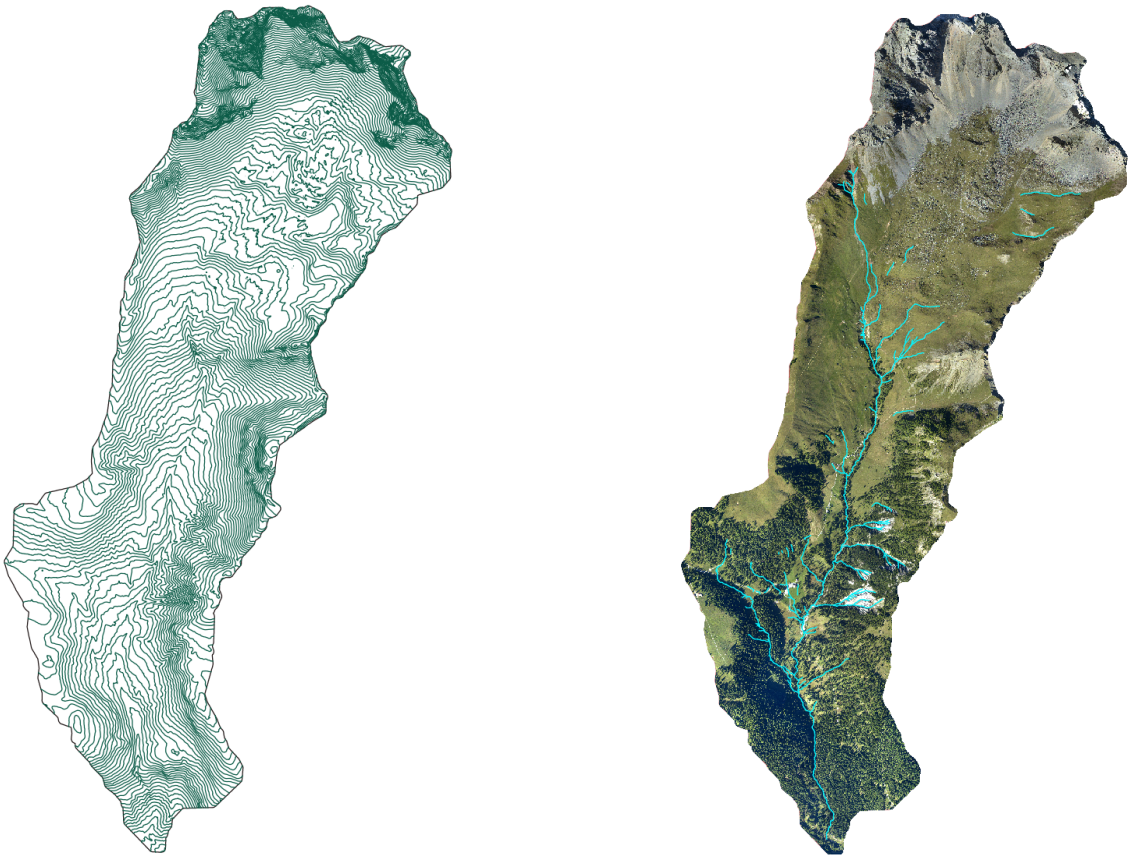
2

Dataset

The Valfredda creek is a headwater stream located in northern Italy, has an alpine climate, and flows from North to South. The maximum drainage area is 5.3 km^2, whereas the maximum contributing area at the confluence with the Biois river is about 7.5 km^2. The stream network is almost 17 km long, with channel widths ranging from 10 cm to 1.5 m. Stream networks are not steady, as they expand and retract in response to changing hydrologic conditions in the surrounding landscape. [3] The elevations range between 1,500 to 3,000 m a.s.l. where pasture and grass cover the biggest area of the catchment; moreover, rocks dominate in the uppermost part of the catchment while forests are observed in the lower part. The mean annual temperature in the catchment is around 4°C, and the mean annual rainfall is about 1.300 mm. During the significant winter snow accumulation the whole catchment is covered by the snow that melts in spring; furthermore, this is the reason why the hydrological regime exhibits a strong seasonality. [9]. Flowing streams expand and contract following ever-changing hydrological conditions of the surrounding environment.[10]

The Figure 2.1a illustrates the shape and contour of the study catchment where each line has a constant elevation in the landscape; moreover, each subsequent line is 10m above the previous one. This figure proves that elevation ranges between 1300 and 1500 m a.s.l. and that the upper part of the catchment is the most elevated. Note that the shape of the catchment is determined by the topographic slope which allowed us to determine all the points that drain through the outlet. The site is characterized by significant spatial heterogeneity in lithology and soil cover [3]. As shown on the satellite view image on Figure 2.1b, in the lower area the forest prevails, while the middle and upper areas of the catchment are characterized by grass, large rocks, debris deposits, and limestone. This orthophoto also shows the stream network in the blue line. The stream bed to the greatest extent is a synthesis of large rocks, gravel, and silt.

The 30 field surveys were taken from July 12th, 2018 to October 10th, 2021 with the different numbers and different dates of observations for each year. In 2018, 9 field surveys have taken a place from 12th July to 3rd November, whereas in 2019 there were only 2 surveys on 18th January and 7th August. Next, 2020 is the year with the highest number of observations and these 12 field surveys were conducted between 12th May and 25th



(a) Shape of the study catchment with the contour lines.

(b) Orthophoto of the study catchment with the stream network outlined in light blue.

Figure 2.1: The contour and orthophoto of Valfredda catchment

November. Lastly, in 2021 there were 7 observations from the 14th of July to 28th of October. Each year was considered separately and using feature engineering, 1 feature was recalculated while 3 features were added to the basic dataset to increase prediction accuracy. Additionally, different subsets of predictors were used (the smallest subset contains 7 and the largest one contains 26 predictors).

2.1 Shape Data

The shape data is containing information about the graph, so the nodes and edges. It includes the points that are collected from the field where each data point in the dataset corresponds to a unique node. The description of the node features is given below:

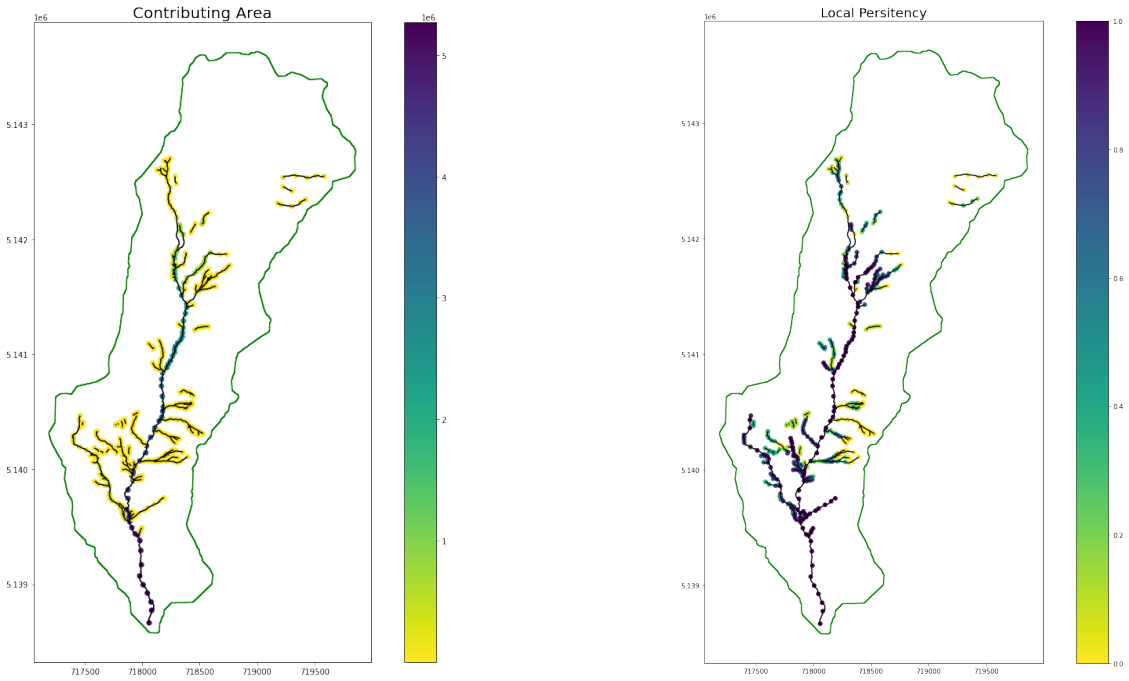
1. Id: An identification number (ID) for each specific node
2. drainsTo: The id of the next node in the network, when moving downstream

3. ad8: The size of the contributing area
4. ad8nopits: The size of contributing area where the area draining into pits has not been counted
5. type: Refers to the type of the node
    - N: Node
    - C: Confluence (the point where two streams become one)
    - H: Head (the up-most node of the stream)
    - S: Sink (the down-most node of the parts that are disconnected from the mainstream network)
6. Pi: The local persistency (the fraction of all observations in which the node was active (had visible flowing water)
7. dates: Correspond to the date of the observations when a survey of the network was taken. The information that was observed is the state of the node which contains one of the following values:
    - 1: The node was active (wet)
    - 2: The node was not active(dry)
    - -1: The node was not observed

The Figure 2.2 represents the mapping of 509 nodes where the color of the nodes was based on the two essential features. The first feature is contributing area and on the Figure 2.2a the nodes are colored from yellow (smallest size of the contributing area) to black (largest size of the contributing area). It is evident that nodes with the largest contributing area are positioned along the mainstream in the network, especially in the lower part of the catchment near the outlet. It is also evident that contributing area increases along the mainstream from the North to the South. Instead, the nodes with the small contributing areas are primarily side stream nodes or nodes in the upper part of the catchment. Another important feature that was also used as the basis for building other features is local persistency. In the Figure 2.2b local persistency is calculated using all available data and nodes are colored from yellow (lowest local persistency) to black (highest local persistency). It is noticeable that most nodes with low persistency are located alongside streams particularly on the middle right side of the main network, while nodes with higher persistency follow the mainstream and lower left side streams of the network.

2.1.1 Data Cleaning and Preprocessing

The first step was to eliminate column ”ad8nopits” because it had missing values. Also, since in the experiments observations were considered each year separately, two observations from field surveys in 2019 were disregarded. The data was sorted to follow the timeline and divided into 3 separate time series by year. Next, the column ”type” was disregarded because it did not hold relevant information. Lastly, column ”Pi” was removed because it was based on the whole dataset and it proposed a strong assumption that we know the distribution not only for data from the past but also data from the future. Later on, this column will be replaced by calculating local persistency at each time step which will eliminate the bias about the future.



(a) Mapping of the nodes where nodes are colored base on the size of contributing area

(b) Mapping of the nodes where nodes are colored based on the value of local persistency

Figure 2.2: Mapping of the fundamental node features

The Figure 2.3 represents the distribution of not observed nodes at each date when a survey was conducted. The number of not observed nodes varies from 0 to 341, and as it is shown the number increases after 2019 whereas in 2020 and 2021 the number of not observed notes varies from 211 to 341.

To fill in the missing data for not observed nodes, a hierarchical model was used with the assumption that nodes activate in the order hierarchically. As the active network expands (retracts) stream portions activate (dry out) following a fixed sequence determined by the spatial patterns of transport capacity, originating stream network dynamics that are strictly hierarchical. [11] Temporary stream activation follows a fixed and repeatable sequence, in which the least persistent sections activate only when the most persistent ones are already flowing. This imposes that stream sections potentially extremely distant in the physical space might share the same average degree of persistency and exhibit systematically synchronous dynamics. [10]

The procedure to fill in the missing data follows these steps:

1. Calculate the local persistency P_i of each node using all available surveys
2. A threshold P* that separates wet from dry nodes is calibrated for each observation date (node i is wet if $P_i >= P*$, or dry if $P_i < P*$)
3. Use P_i and calibrated P* to fill in missing data

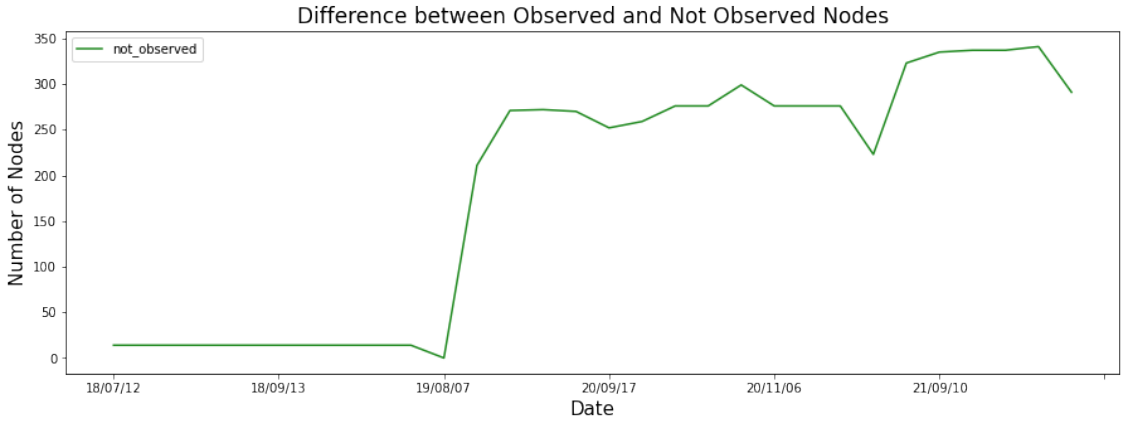


Figure 2.3: The distribution of not observed nodes through time

2.2 Weather Data

The weather data comes from a weather station of the Veneto Region Environmental Protection Agency (ARPAV) located in Falcade and it covers all the dates of the field surveys of the network. This time-series data are characterized by the following:

1. prcp: daily precipitation in mm
2. radsol: solar radiation
3. snow: snow height on the ground measured at Cima Pradazzo in cm
4. Tmin: minimum temperature in °C
5. Tmean: mean temperature in °C
6. Tmax: maximum temperature in °C
7. RHmin: minimum relative humidity in %
8. RHmax: maximum relative humidity in %
9. windspeed: speed of the wind in m/s
10. date: dates are available since 2010

2.2.1 Data Cleaning and Preprocessing

Firstly, the date column was converted to consider only the date. Also, only the years and dates that are correspond to the dates of the field surveys are considered. The dates are considered not relevant if they are not in the time frame of 1, 7, 14, or 30 days before the observation date.

In order to calculate the mean relative humidity the minimum and maximum relative humidity values and the following formula were used: $RHmean = \frac{RHmin+RHmax}{2}$. Because "Tmean" column had missing values, I used the same formula to populate empty fields. Then, rows with missing values in the columns "radsol", "neve" and "RHmean" are removed.

Time series data is exploited into 5 different datasets based on daily, weekly, biweekly, and monthly approximations of feature values. The daily dataset contains the measurements of the 1 day before the observation, while the weekly, biweekly, and monthly contain approximations of the prior 7, 14, and 30 days' data respectively. The fifth, full dataset, contains all weather data combining together four previously mentioned datasets.

2.2.2 Data Analysis

Before anything else, precipitation frequency was examined and the results are shown in the Figure 2.4. The numbers in each cell show how many rainy days occurred during a specific month and specific year. The lowest count of rainy days is 5 and lower numbers are colored in green and dark blue color shades. The highest number of rainy days happened in July 2019 and months with higher numbers of rainy days are colored in light blue shades. Generally, the data supports the main characteristics of the Alpine climate with high precipitation during summer. An evident outlier is 2019 when there were many precipitation days also during October, November, and December.

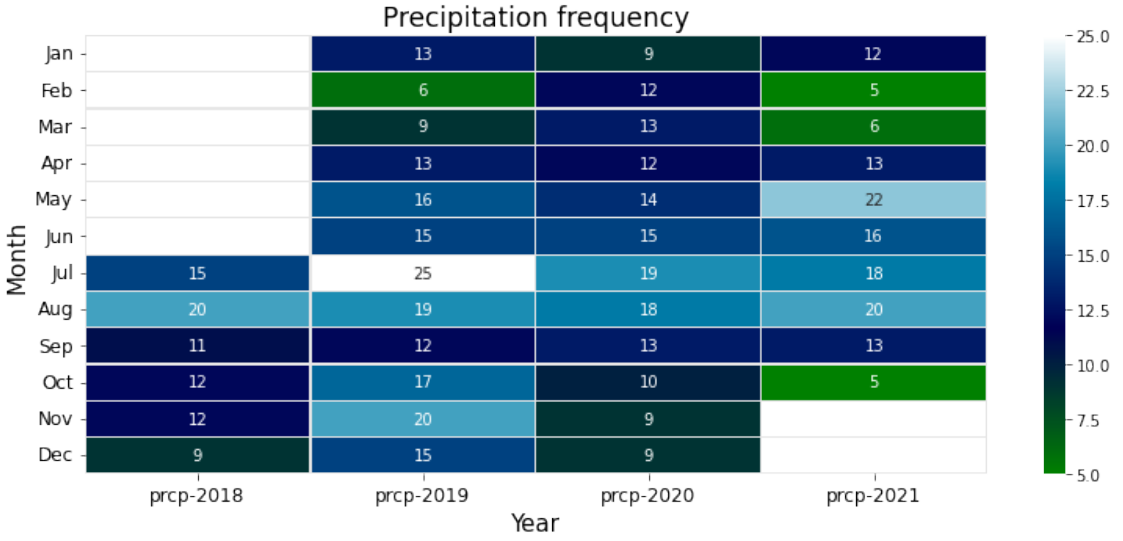


Figure 2.4: Precipitation frequency

Another proof that the weather in the catchment has characteristics of the Alpine climate is the snow frequency shown in the Figure 2.5. Same as before, the numbers represent the count of snowy days for each month between July 2018 and October 2021. It is noticeable that a high number of snowy days are during winter, early spring, and late fall. During summer there are very few days when it was snowing with the exception of 2021 when it was snowing more during summer compared to the previous 3 years.

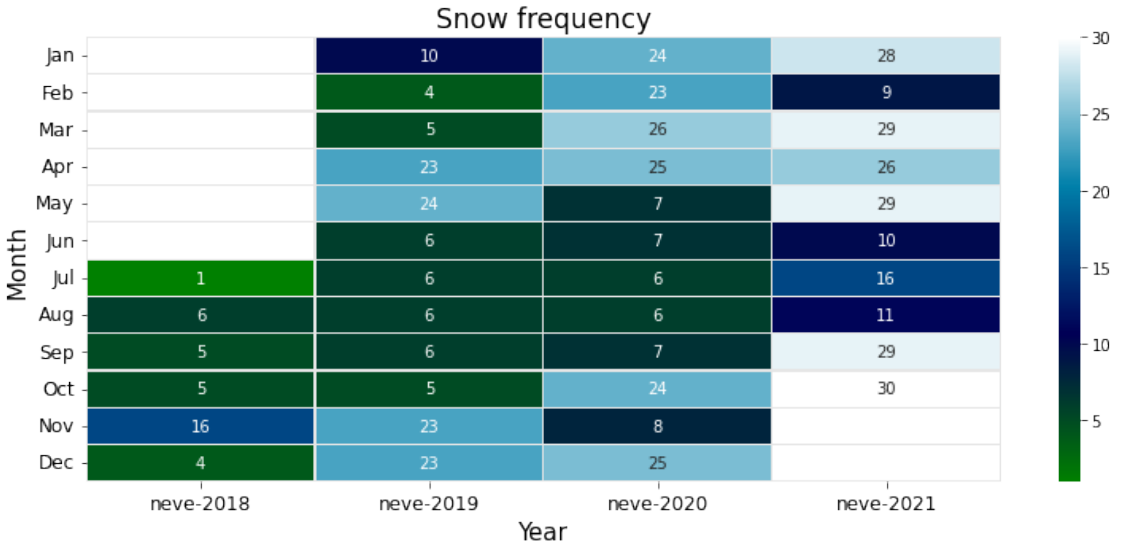


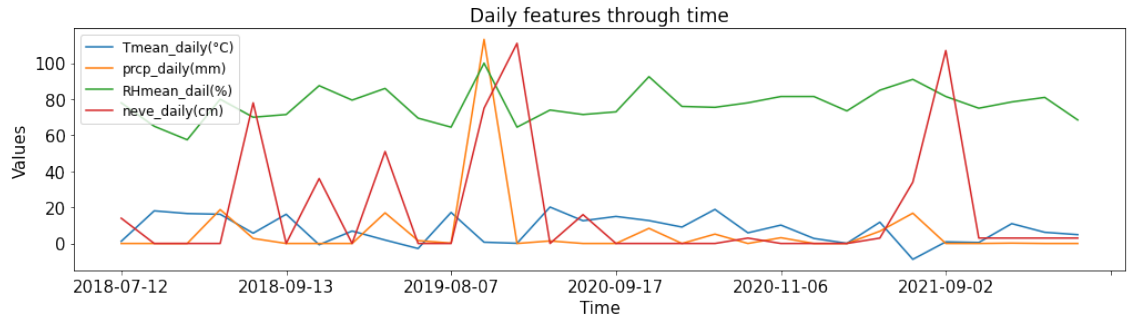
Figure 2.5: Snow frequency

The weather data feature correlation matrix on the Figure 2.7 represents the correlation coefficients among all features where darker colors have features with negative or lower correlation while brighter colors depict features with higher correlation. The highest correlation coefficient of 0.68 has mean temperature and solar radiation. As expected, also the high coefficient of 0.41 has precipitation and relative humidity. The highly negative correlation of -0.56 has mean temperature and snow as well as solar radiation and relative humidity whose coefficient is -0.44.

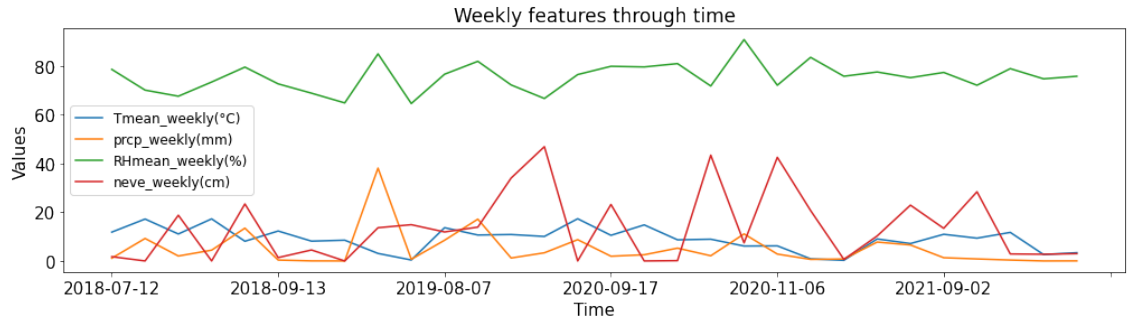
Since the weather data was divided into four datasets that included the daily, weekly, biweekly, and monthly values approximations, the analysis of the main weather features was divided by datasets and further examined using Figure 2.6. The first noticeable fact is that the distribution of the weather data becomes smoother when approximated over a longer period of time. For instance, there are many peaks in temperature on the day before the observations while when an approximation of 30 days before the observation is taken into account there is hardly one. While the peaks are the most conspicuous in the daily data, the weekly and by-weekly data have more distinct jumps almost in all four variables. The distribution of the snow seems to be almost the same across all 4 datasets which can be contributed to the fact that most observations were done during summer or fall. It is hard to notice on the monthly distribution graph, but on the other three graphs, it is easy to see that precipitation and relative humidity have spikes almost always at the same. Next, with the increase (decrease) in snow, temperature always decreases (increases).

2.3 Feature Engineering

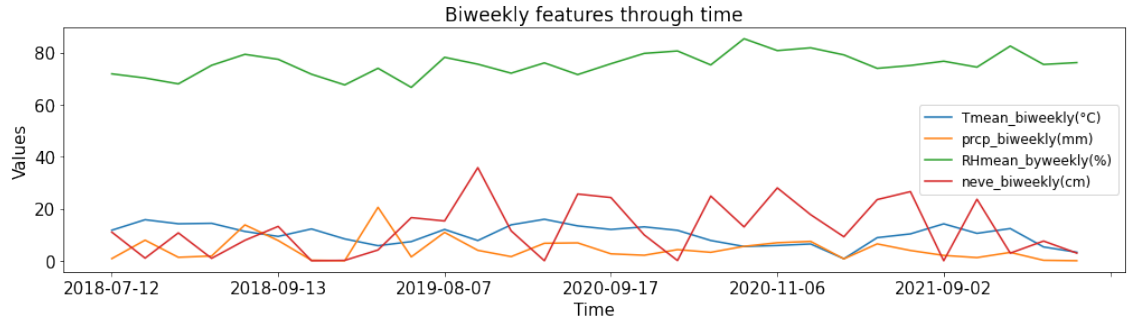
The nodes will be classified by the models as wet or dry in order to determine the dynamics of the Valfredda river network. Different combinations of node features were used to classify the state of the node along with the different approximations of the weather data. The basic dataset contained only the contributing area of the node



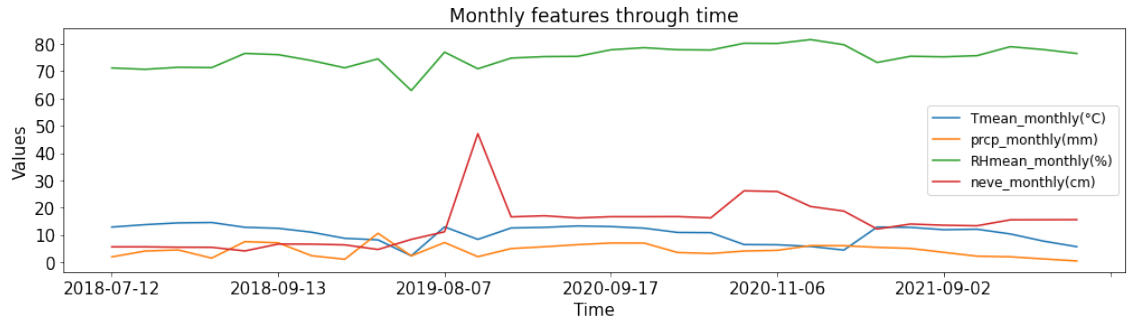
(a)



(b)



(c)



(d)

Figure 2.6: Weather Analysis

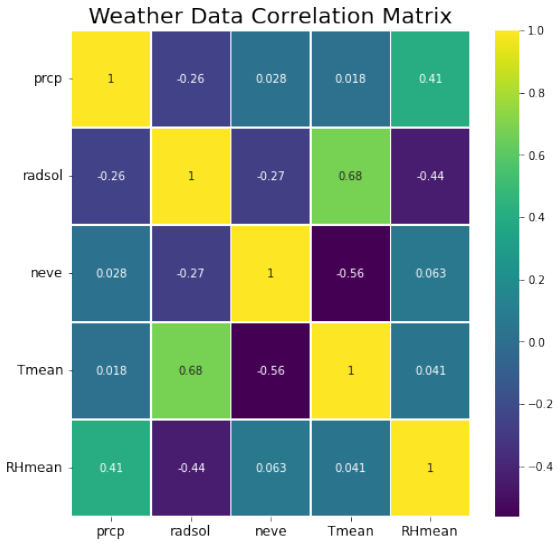


Figure 2.7: Weather Data Correlation Matrix

associated with one or all weather datasets. This served as the basis and did not include any temporal or spatial information. The first extension to the basic dataset was introduced by considering the time and the history of the previous states of the node. At this step, only time-series information was taken into account; moreover, the second extension comes from the graph data where two different connections between neighboring nodes were used. The two connections included the weighted average of the current states of the neighboring nodes and the weighted average of the previous states of the neighboring nodes with the assumption that nodes with the same or similar local persistency are synchronous and behave similarly. The spatio-temporal correlation was calculated for the specific node at each time step separately for every year. The weights were calculated based on the similarity in local persistency between the two nodes. This approach was used because of the assumption that nodes with higher local persistency will change from dry to wet before the nodes with lower local persistency values; moreover, nodes that have lower local persistency transition from wet to dry before the nodes with higher local persistency. In other words, the nodes with lower local persistency will be wet after the nodes with high local persistency are already wet.

To include information about the history, a feature ”previous_state” was created. This feature represents the state of the node at the previous observation. With this being said, if we consider a node at time step t, then the previous state of that specific node would be its state at time step t-1. Since the nodes at the first observation do not have the previous state, the first observation was excluded from the dataset. Considering this, the second observation becomes the first one in our dataset.

The local persistency quantifies the probability of observing surface flow in a given location within the network. Accordingly, the spatial distribution of flow persistency illustrates which portions of the stream network are more likely to experience surface runoff when the catchment wets up. [10] To provide additional information about the state of the node in the past, the local persistency was introduced as a feature and calculated so that it considers the past of each node separately. If we consider the node at time step t, the local persistency of that node is equal to the average of all previous states of that node including the state at the time step t. This feature was calculated

at each time step.

To create a correlation between pairs of nodes with the same or similar local persistency the last two features "Weighted_average_class" and "weighted_average_previous_state" were added. They contain spatio-temporal information and provide a correlation of local persistency between two nodes where the coefficient between two nodes should be high if they are correlated. By doing this at each time step, we have the correlation history between pairs of nodes and if two nodes have the same or similar persistency, they are synchronous and behave in the same way. The main specification of this feature is that it exploited the information in the training data and calculated "Pi" at each step only for the training data, while for the test data, it remained unchanged. So "Pi" calculated at the last step of the training dataset was used as the approximation and remained constant for all the time steps in the test data. These two features were calculated in two steps and exploited the similarity in local persistency among two nodes:

1. Calculate similarity adjacency matrix Aij using:

$$A_{ij} = 1 - |P_i - P_j|$$

where
P_i represents the local persistency of node i calculated over all training data
P_j represents the local persistency of node j calculated over all training data

2. Given a node, computed weighted average of the state of the neighboring nodes at each time step using:

$$\overline{X}_w = \frac{\sum_{i=1}^{\#ofnodes} x_i * w_i}{\sum_{i=1}^{509} w_i}$$

where
$\overline{X}_w$ is the weighted average variable
w_i is allocated weight value from the similarity adjacency matrix defined by the similarity in the local persistency of two nodes
x_i is the state of the observed node (wet/dry)

2.4 United Data Preprocessing

After both, the shape and climatic data, were cleaned, preprocessed, and analyzed, they were merged together to create one dataset which will be used for training and prediction. Because the node features were static while the weather data and states of a node were dynamically changing, the dataset was constructed in a way that each data point contains information about a specific node at a specific time step. So all the nodes at the same date of observation had the same weather information. After splitting features and target (state of the node), the data was divided into the training and testing datasets. In the first experiment, the data from the observations in the year 2018 served as training data, while the data from 2020 was used as a testing dataset. In the second experiment, observations from 2018 and 2020 were used to train and fine-tune the model, while the data from 2021 were used for testing purposes. Lastly, data were scaled using scikit-learn function MinMaxScaler() which transforms features

by scaling each feature to a given range that in this project was equal to feature range [12]. The transformation is given by:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

After preprocessing the merged dataset, it is evident that the data is unbalanced and the statistics for each year are as follows:

- 2018: 509 nodes x 8 observations = 4072 samples out of which 2314 were classified as wet and 1758 were classified as dry
- 2020: 509 nodes x 12 observations = 6108 samples out of which 3536 were classified as wet and 2572 were classified as dry
- 2021: 509 nodes x 7 observations = 3563 samples out of which 1849 were classified as wet and 1714 were classified as dry

3 Methods

3.1 Models

The main focus of the project was binary classification of the state of a specific node using time series data over graph data. It is the allocation of unknown samples to a known class-based feature vector [13]. The choice of a classification algorithm depends on the type of the problem and predictors. In this project, five models were trained and tested and they are: Logistic Regression, Decision Tree, Random Forest, k-Nearest Neighbors, and Support Vector Machine. In general, Logistic regression is a statistical method for predicting binary classes while Decision Trees are a non-parametric supervised learning method used for classification and regression which also serves as the building block of the Random Forest Classifier. KNN is Classifier implementing the k-nearest neighbors vote while Support vector machines are a set of supervised learning methods used for classification, regression, and outliers detection. In most machine learning tasks, the goal is not only to find the most accurate model of the response but also to identify which of the input variables are the most important to make the predictions, e.g., in order to lead to a deeper understanding of the problem under study[7]. With this being said, interpretability was also the of the project where it was explored and evaluated how the models perform with the different subsets of predictors and which features play the biggest roles.

3.1.1 Logistic Regression

Logistic regression sometimes called the logistic model or logit model, analyzes the relationship between multiple independent variables and a categorical dependent variable and estimates the probability of occurrence of an event by fitting data to a logistic curve [14]. This model can handle the non-linear relationship between independent and dependent variables because it is using non-linear log transformation of the linear regression. In this project, the focus was on binary logistic regression which is typically used when the dependent variable is dichotomous

and independent variables are either categorical or continuous. The explanation of the impact of the predictors can usually be found in the terms of odds because the LR calculates the probability that an event will occur over the probability that it will not occur. LR is modeling the response variable p in terms of explanatory variable x following the equation $p = \alpha + \beta x$. To make sure that extreme values of x still produce the result of the equation to falling between 0 and 1, the above-mentioned equation of odds is transformed using the natural logarithm where we model the natural log odds as a linear function of the explanatory variable as follows:

$$\log(\frac{p(x)}{1-p(x)}) = \alpha + \beta x$$

where p is the probability of an event occurring, x is the explanatory variable , and α and β are regression coefficients. These two coefficients are usually estimated using a method of maximum likelihood of observing sample values that will provide coefficient values which are maximizing the probability of obtaining the dataset [15].

In this project, 4 different parameters of LR models were fine-tuned while the default values were used for the rest of the parameters. These parameters include 'penalty', 'tol', 'C', and 'solver'.[12]

- 'penalty' - specifies the norm of the penalty and as input takes values:
  - None: No penalty is added
  - 'l2' (default): Add an L2 penalty term and uses the sum of the squares of the parameters (shrinks the coefficients to be close to 0)
  - 'l1': Add an L1 penalty term that limits the size of the coefficient and is equal to the absolute value of the magnitude of coefficients (can yield sparse models)
  - 'elasticnet': Both L1 and L2 penalty terms are added
- 'tol' - Tolerance which is a threshold for stopping criteria and stops the iterations of the algorithm when the threshold is reached
- 'C' - Inverse of regularization strength applied to the penalty term where smaller values specify stronger regularization
- 'solver' - Algorithm to use in the optimization problem which depends on the penalty chosen and as input takes values:
  - 'lbfgs' (default) - Supports 'l2' and None penalties
  - liblinear' - A good choice for small datasets; limited to one-versus-rest schemes; supports 'l1' and 'l2' penalties
  - 'newton-cg' - Supports 'l2' and None penalties
  - 'sag' - Faster for large datasets; supports 'l2' and None penalties
  - 'saga' - Faster for large datasets; supports 'elasticnet', 'l1', 'l2' and None penalties

3.1.2 Decision Tree

Decision tree is a model that predicts the target value by learning simple decision rules which are inferred from the data features. The DT consists of a root node, internal also referred to as test nodes, and leaf nodes also called terminal or decision nodes. These nodes together assemble a directed rooted tree where the root node has no incoming edge. On the contrary, internal and leaf nodes have exactly one incoming edge. The crucial difference between internal and leaf nodes is that internal nodes also have outgoing edges. Another characteristic is that each test node split the instance space into two or more sub-spaces according to a discrete function of the input features. Deeper trees produce models that are fitted better and implement more complex rules. Generally, less complex decision trees are considered to be more comprehensible; moreover, the complexity of the tree has an essential effect on the model accuracy[16]. The complexity is often measured by the total number of nodes, total number of leaves, tree depth, or total number of predictors. To reduce over-fitting which occurs when the model is too complex and does not generalize the data correctly, specifying the minimum number of samples at the leaf node or maximum depth of the tree may be used. Decision tree induction is tightly linked to rule induction because each path from the root node to one of the leaves can be converted into a rule. To transform a path into a rule we can combine the tests along the path from the root to the leaf to create the antecedent part. In this case, the predicted class of the leaf is considered to be the class value. Each leaf is assigned to exactly one class which represents the most fitting target value. Samples are classified using a top-down approach according to the outcome of the tests conducted along the path from the top (root) down to the bottom (leaves).[17] Decision tree uses a white box model which means that the explanation for the condition is explained by Boolean logic if a given situation is observable in the model.

Dts use the loss function that assesses the split based on the purity of the resulting nodes (ratio of the data that belongs to each class before and after the split). The default values were used for all the parameters except for "criterion", "max_depth", and "min_samples_split"[12].

- "criterion" - The function to measure the quality of a split and as input takes
  - "gini" - Gini index is an impurity-based criterion that measures the divergences between the probability distributions of the target attribute's values[17], or in another words, the variance across the different classes [18];

    $$G(node) = \sum_{n=1}^{c} p_n(1 - p_n)$$

    where p_n is the probability of picking value from class n, and $(1 - p_n)$ is the probability of not picking a value from class n
  - "entropy" - Information gain is an impurity-based criterion that uses the entropy measure (origin from information theory) as the impurity measure [19]

    $$Entropy(node) = -\sum_{n=1}^{c} p_n \log(p_n)$$

    where p_n is the probability of picking a value from class n
  - "log_loss" - The likelihood–ratio which is useful to measure the statistical significance of the infor-

mation gain criterion [17]

$$L_{log}(y, p) = -(y \log(p) + (1 - y)(\log(1 - p))$$

for each single sample with true label $y \in 0, 1$ and probability estimate $p = Pr(y = 1)$ [12]

- "max_depth" - The maximum depth of the tree (provided values for the grid search were between 2 and 40)
- "min_samples_split" - The minimum number of samples required to split an internal node (provided integer values for the grid search were between 2 and 600)

3.1.3 Random Forest

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [12]. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [7]. Aggregating numerous adaptations of estimation into an ensemble can produce significant improvement in accuracy because each tree classifier in the random forest casts a unit vote for the most popular class at input x. Since DTs usually have small bias and high variance which makes them very likely to benefit from the averaging process, they are the ideal choice to be used in the ensemble methods. [20] Random forests provide the better interpretability of the model because it is possible to identify which predictors are the most important to make the prediction via several mechanisms for feature importance measures. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them.[7]

The default values were used for all the parameters except for "criterion", "max_depth", and "n_estimators"[12].

- "criterion" - refer to the section 3.1.2
- "max_depth" - refer to the section 3.1.2
- "n_estimators" - The number of trees in the forest

3.1.4 K-nearest Neighbor

K-Nearest Neighbors is a very powerful non-parametric classification algorithm that classifies based on a similarity measure. This algorithm finds k-nearest neighbors in the training dataset that is closest to the test sample and by using majority voting assigns the most common class amongst its K-nearest neighbors to the test sample. The k-NN approach is also an example of a lazy learning technique, that is, a technique that waits until the query arrives to generalize beyond the training data. [21] It is computationally expensive (especially for large datasets) because

classifying one sample usually requires computing the distance of the test sample to all the objects in the training set.

The basic k-NN algorithm follows 5 steps:

1. Determine parameter K (number of neighbors)
2. Calculate the distance between the unlabeled sample and all the labeled data
3. Sort the distance and determine the nearest neighbors based on the k-th minimum distance
4. Gather the labels of the K nearest neighbors
5. Use majority vote to assign a predicted label

K is the parameter that controls the volume of the neighborhood and consequently, the smoothness of the density estimates is k number of neighbors.[13] Although larger values of k produce a smoother boundary effect, if k is too large, there could be too many points from other classes in the node neighborhood. On the contrary, if values are too small, then the model may produce a result that is sensitive to noise points. There are many ways to choose the optimal value of the key ([13][22]), but in this project, k was chosen empirically. The default values of the KNN model were used for all the parameters except for "n_neighbors", "weights", and "algorithm"[12].

- "n_neighbors" - Number of neighbors to use by default for k-neighbors queries
- "weights" - Weight function used in prediction which input takes
  - "uniform" - All points in each neighborhood are weighted equally (uniform weights)
  - "distance" - Weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away
- "algorithm" - Algorithm used to compute the nearest neighbors
  - "auto" - Attempts to decide the most appropriate algorithm based on the values passed to fit method
  - "ball_tree" - Uses BallTree algorithm for fast generalized N-point problems
  - "kd_tree" - Uses KDTree algorithm for fast generalized N-point problems
  - "brute" - Uses a brute-force search

3.1.5 Support Vector Machine

Support Vector machines are effective in high dimensional spaces, memory efficient, and versatile linear and maximal margin classifiers where the margin is a crucial geometric quantity associated with this algorithm. SVMs use a subset of training points in the decision function called support vectors.[12] In their most general formulation, SVM finds a hyperplane in a space different from that of the input data x. [23] It is a hyperplane in a feature space induced by a kernel K (the kernel defines a dot product in that space) and through kernel the hypothesis space is

defined as a set of "hyper-planes" in the feature space induced by K[24]. SVM is a discriminant technique aiming to find, based on an independent and identically distributed training dataset, a discriminant function that can correctly predict labels for newly acquired instances [25]. SVM in practice minimizes a trade-off between empirical error and complexity of hypothesis space. Formally this is done by solving the following minimization problems for classification[23]:

$$\min_f \| f \|_K^2 + C \sum_{i=1}^{l} |1 - y_i f(x_i)|_\xi$$

The SVM parameters that were fine-tuned are the following [12]:

- "C" - Strictly positive penalty parameter of the error term that adds a penalty for each wrongly classified data point; The strength of the regularization is inversely proportional to C
- "degree" - Non-negative degree of the polynomial kernel function 'poly'
- "gamma" - Kernel coefficient for 'rbf', 'poly' and 'sigmoid
- "kernel" - Specifies the kernel type to be used in the algorithm
  - "linear" - The simplest formulation of SVM where the hyperplane lies on the space of the input data x [23]
  - "rbf" - is the squared-exponential kernel that have as the basis functions radially symmetric function such as Gaussian, multi-quadratic, or different spline functions[26]

    $$k(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{2l^2}\right)$$

    where l is the length scale of the kernel and $d(\cdot, \cdot)$ is the Euclidean distance
  - "poly" - Similarity of vectors in the labeled dataset in a feature space over polynomials of the original variables used in the function

    $$k(x_i, x_j = (x_i \cdot x_j + 1)^d$$

    where d is the degree of the polynomial
  - "sigmoid" - Equivalent to activation of a two-layer perceptron

    $$\sigma(x) = \frac{1}{1 + e^{-(x)}}$$

    where x is the input to the function

3.2 Experiments

3.2.1 Description

The main goal was obtaining high accuracy which would mean that the model has a good performance and classifies the state of the node well. As mentioned before, the first big division of the datasets comes during the training

and testing datasets split. In the first experiment, data from 2018 was used for training purposes, while the data from 2020 was used to test the models. The second experiment included data from 2018 and 2020 which were training data in this case, and 2021 data which served as testing dataset. The additional features, such as the previous state of the node, local persistency, weighted class average, and weighed previous state average were calculated for each experiment separately according to the datasets used. The second dataset division comes from the weather data approximation, where 5 different datasets were created for each year (daily, weekly, biweekly, monthly, and all). These four datasets were merged with different subsets of the graph dataset. Firstly, they were merged with the basic dataset, and then gradually more temporal and special features were incorporated. Each new feature was tested with all temporal data and included all the combinations done previously. For instance, if we take into consideration one weather dataset, to create new subsets firstly all combinations of features that include information about the past were put together. Then, all these combinations were combined further with all the features that include spacial information about the nodes. All these different combinations created 24 different combinations.

All 5 machine learning models which include LR, DT, RF, k-NN, and SVM were trained, fine-tuned, tested, and evaluated in both experiments using these 24 different versions of the dataset and using 5-fold cross-validation. The Python "sklearn" library was used for modeling and prediction. This library along with "matplotlib" was used to evaluate models and visualize the results. The function GridSearchCV() was used to fine-tune all the models besides SVM for which RandomizedSearchCV() function was used because of the time complexity. Both functions as input use the model, number of cross-validation folds, and scoring function. The only difference is that for digit data, GridSearchCV() takes as input the values, while RandomizedSearchCV() takes the range from which it randomly selects the data.

3.2.2 Evaluation

Since there were separate training and testing datasets and all 5 models were used for modeling 24 different variations of the datasets, there were 800 models to evaluate. Primarily, the models were trained and evaluated using the accuracy score function. Then, based on validation accuracy, for each year and each division of weather data approximation, the best model was selected for further evaluation. Initially, the influence of historical data was considered, and then the focus was on the best model among all temporal and spatial features subsets included. For instance, if we take into consideration the LR model, firstly we look at the model before and after incorporating data about the past. Next, we take a look at all 24 feature combinations where daily weather data is approximated, and out of these 24 models, the model with the highest validation accuracy is chosen. The procedure is the same also for weekly, biweekly, and monthly datasets. After this model elimination, there were 20 models left (10 for each of the two experiments). These 20 models were again evaluated using a classification report, confusion matrix, and ROC AUC score.

The classification report is based on ratios between True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) which produce accuracy, precision, recall, and F1 scores. The definitions and equations of these terms are:

- TP: Outcome where the model correctly predicts the positive class
- TN: Outcome where the model correctly predicts the negative class
- FP: Outcome where the model incorrectly predicts the positive class

- FN: Outcome where the model incorrectly predicts the negative class
- Accuracy: Measures the accuracy of all predictions (positive and negative) and gives the percentage of correct classification

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision: Percentage of correct positive predictions relative to total positive predictions

$$Precision = \frac{TP}{TP + FP}$$

- Recall: Also referred to as Sensitivity or True Positive Rate (TPR) and as outcome has percentage of correct positive predictions relative to total actual positives

$$Recall = \frac{TP}{TP + FN}$$

- F1 Score: A weighted harmonic mean of precision and recall (The closer to 1, the better the model)

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

- Specificity: The probability that the model predicts negative outcome correctly

$$Specificity = \frac{TN}{FP + TN}$$

A confusion matrix (also known as an error matrix) is a contingency table that is used for describing the performance of a classifier/classification system, when the truth is known.[27] It represents a summarized table of the number of correct and incorrect predictions in a form of an NxN matrix where N is the number of target labels. FP is also known as Type I error, while FN is referred to as Type II error.

The Receiver Operating Characteristic (ROC) curve displays the trade-off between sensitivity and specificity of a classification model and is very suitable when dealing with unbalanced data because it is dealing with the scores and not directly with the data. Plotting the ROC curve is a popular way for discriminatory accuracy visualization of the binary classification models and the area under this curve (AUC) is a common measure of its exact evaluation [28]. Points above the diagonal dividing the ROC space represent good classification results (better than random), while points below represent poor results (worse than random). [14] The area under this curve provides an overall measure of fit of the model [29]. The AUC varies from 0.5 (no predictive ability) to 1.0 (perfect predictive ability) and the larger the AUC score, the better the model performance. This curve is a two-dimensional plot that illustrates how well a classifier system works as the discrimination cut-off value is changed over the range of the predictor variable. Sensitivity and specificity are inversely related where

$$Sensitivity = TPR$$

and

$$Specificity = 1 - Sensitivity$$

4 Results

The main focus of this study was not only the model performance but also the exploration of feature importance and interpretability. In this chapter, the basic model is referred to as a model that did not have included any information about past or graph structure while training and fine-tuning. It will be explored how each of the predictors that were introduced affected the model performance. Firstly, information about the history of the nodes will be added, and later on, also information about the graph with be added. Different metrics will be used for evaluation such as accuracy, F1 score, confusion matrix, and ROC-AUC score. The main focus will be on the models that perform well in order to understand the reasoning behind their performance. If there are models with the same performance, the less complex models are explored for ease of interpretation and analysis. Lastly, if the models have the same complexity, the models that are easier to interpret are reported.

4.1 Baseline and Historical Data and Models

The training of the baseline models was done using the contributing area of the node and weather data. After, when training and testing the historical models, the first feature that was added to the training dataset included information about the past and in particular about the previous state of the node. Next, the local persistency was added and the baseline models were tested with only this additional feature, while at the end both features were included. The same approach was used in both experiments; moreover, in the first experiment data from 2018 was used to predict the state of the nodes in 2020 whereas, in the second experiment, the data from 2018 and 2020 was used to predict the state of the nodes in 2021.

The Table 4.1 shows the top 5 baseline models with the highest scores for each experiment. In the first experiment where data from 2018 was used to predict the state of the node in 2020, the model with the highest scores is DT Biweekly with an accuracy of 85%. DT classifier is not only the best model, but this classifier worked the best also on the daily and all weather data. The second and third places are RF weekly and biweekly classifiers with

BASELINE MODELS

Experiment I				
MODEL	**ACC**	**PREC**	**REC**	**F1 SCORE**
DT Biweekly	0.85	0.85	0.86	0.85
RF Weekly	0.83	0.84	0.82	0.83
RF Biweekly	0.83	0.83	0.83	0.83
DT All	0.82	0.82	0.81	0.81
DT Daily	0.81	0.81	0.82	0.81

Experiment II				
MODEL	**ACC**	**PREC**	**REC**	**F1 SCORE**
KNN Biweekly	0.89	0.90	0.89	0.89
KNN Weekly	0.88	0.88	0.88	0.88
KNN Daily	0.88	0.88	0.88	0.88
RF Daily	0.87	0.87	0.87	0.87
RF Biweekly	0.86	0.87	0.85	0.85

Table 4.1: Best Baseline Model Selection

HISTORICAL MODELS

Experiment I					Additional Features	
MODEL	**ACC**	**PREC**	**REC**	**F1 SCORE**	**Local Persistency**	**Previous State**
LR Biweekly	0.89	0.89	0.89	0.89	✓	
KNN Biweekly	0.87	0.88	0.86	0.87	✓	
LR Daily	0.85	0.85	0.85	0.85	✓	
LR Weekly	0.84	0.85	0.82	0.83	✓	
KNN Daily	0.83	0.83	0.81	0.82	✓	✓

Experiment II					Additional Features	
MODEL	**ACC**	**PREC**	**REC**	**F-1 SCORE**	**Local Persistency**	**Previous State**
LR Biweekly	0.92	0.92	0.91	0.92	✓	
LR Daily	0.91	0.91	0.91	0.91	✓	
KNN All	0.91	0.91	0.91	0.91	✓	
RF Biweekly	0.91	0.91	0.91	0.91	✓	
DT Biweekly	0.91	0.91	0.91	0.91	✓	✓

Table 4.2: Best Historical Model Selection

accuracy and an F1 score of 83%. It can be inferred that algorithms that learn simple decision rules from the data such as DT and RF are the most suited classification algorithms for this baseline experiment since these 5 models have the best scores when using 2018 and 2020 years data. In the second experiment, not only that the KNN biweekly model performed best with an accuracy of 89%, but also the top 3 models with the best scores are KNN trained and tested on different weather data. The higher accuracy in the second experiment can be explained by the size of the training dataset because in the first experiment only 7 observations of each node were used for the training, while in the second experiment, 19 observations were used.

The Table 4.2 shows the selection of the top 5 historical models with the best performance. In both experiments, the model with the highest scores is LR Biweekly with an accuracy of 89% in the first experiment and 92% in the second experiment. Additionally, the Table 4.2 shows that models classify better when additional information about the history is included. Regarding the first experiment, the DT and RF were the best baseline models but when the information about the past is included, LR and KNN were classifying the data more accurately. It is important to note how big a role the local persistency and previous state of the node have because almost all models that had signs of progression used at least one of them as a predictor. In the first experiment, only the previous state of the node was used by 4 models and only local persistency was used by 15 models, while both previous state and local persistency were used by 6 models. If we take a look at Experiment I in Table 4.2, it is shown that all models used local persistency as one of the predictors, whereas only one model also used the previous state. This implies that local persistency is the crucial additional information from which the model benefits.

To analyze deeper the influence features have on the model, the Figure 4.1 shows baseline DT Biweekly feature importance obtained from the coefficients. The higher the score, the more that specific feature will have an effect on the model. It is obvious that contributing area of the node and precipitation highly influence this classifier when no additional information is included. After them, with the much smaller coefficients are solar radiation and temperature. Features that have very low scores close to 0 such as relative humidity and snow are not very relevant to this model's prediction.

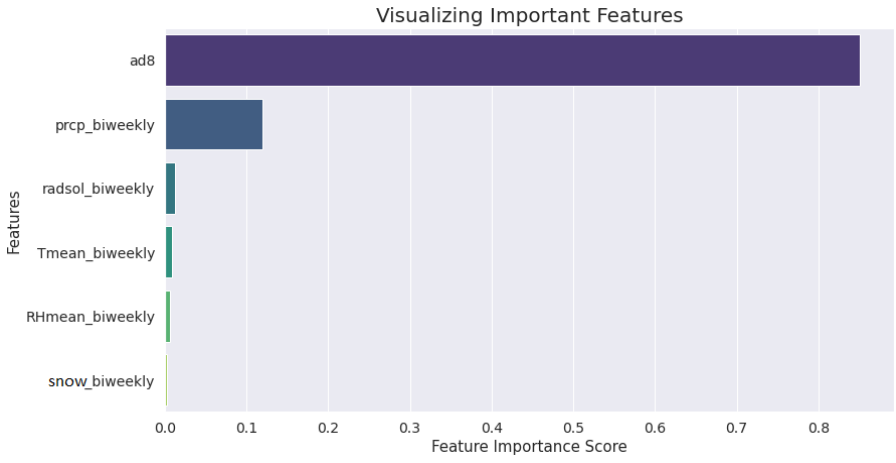
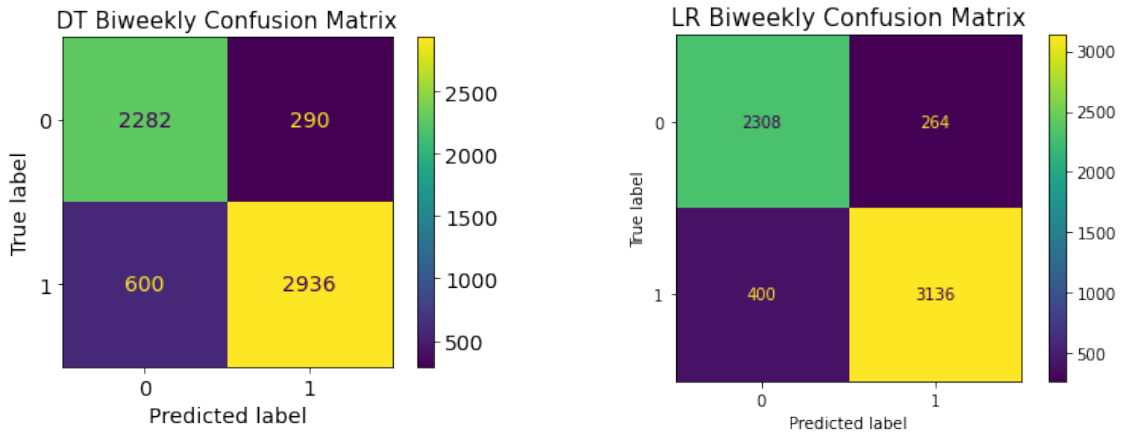


Figure 4.1: Baseline DT Biweekly Feature Importance

4.1.1 Experiment I

The average accuracy of all baseline models is 72%, while for the models including past data, predictors is 84%. Thus, in general, the accuracy increased for about 12%, and to better understand these differences, Figure 4.2 is presented. This figure is presented to compare the two best models in each category in the first experiment. It is clear that the baseline model makes a very big number of Type II errors (600) compared to the Historical Model (400). Both models have a higher number of FN than FP, but overall, the historical model correctly classified more nodes and has a higher number of TP and TN, hence the difference in the accuracy and F1 score. When temporal correlation through local persistency is introduced to the model as a feature, the prediction is more balanced and the number of labels that were FN is much smaller than in baseline models.



(a) Best Baseline model

(b) Best Historical model

Figure 4.2: Experiment I: Best baseline and historical models confusion matrices

The Figure 4.3 shows the ROC curves and AUC scores for the two best baseline and historical models in the first experiment. The blue line represents the ROC of the baseline model and the orange line represents the ROC of the best historical model. The difference between these two models' AUC scores corresponds to the difference in the accuracy and F1 score as well because the biweekly baseline model has worse performance than the biweekly historical model. Since the historical model has much higher scores than the baseline model, this means that the historical model will assign a higher probability to the random positive sample than a random negative sample. Since the historical model's scores are high, they also have a better ability to accurately classify the state of the node.

To illustrate how much actual influence local persistency has on the LR model, we will analyze the historical LR biweekly feature importance plot on the Figure 4.4. The bar plot shows that local persistency has the highest score which is more than 0.7, and second and third place take precipitation and contributing area with coefficients 0.5 and around 0.25 respectively. When it comes to other features, solar radiation has a coefficient of around 1.5, whereas snow and temperature are close to 1. The relative humidity is the only feature that has a negative coefficient and is close to 0.

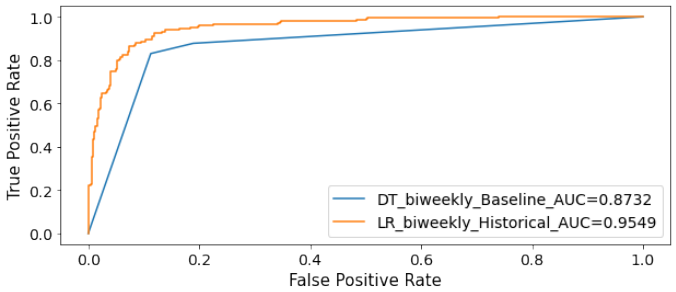


Figure 4.3: Experiment I: Best Baseline and Historical ROC and AUC

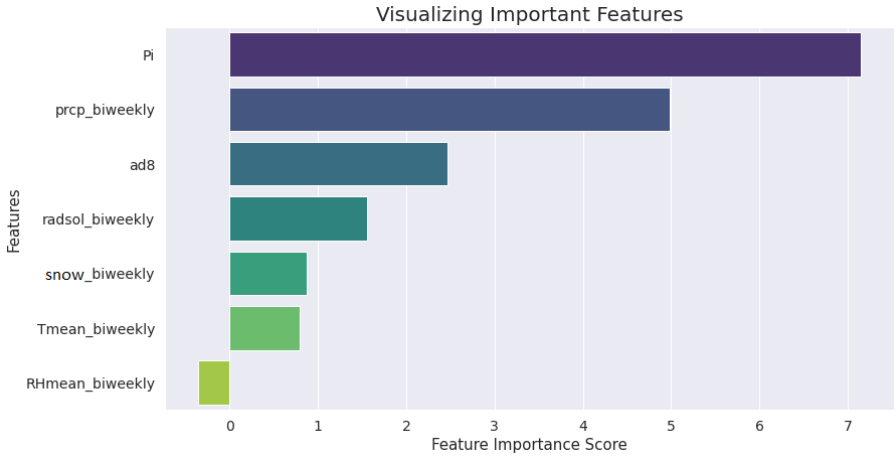


Figure 4.4: Historical LR Biweekly Feature Importance

4.1.2 Experiment II

The average accuracy is 89% for the historical data while it is 74% for baseline models. Again, an important role in the modeling has the local persistency and previous state of the node because 22 out of 25 models show improvement when information about the past is considered. In particular, 19 out of these 25 models show higher achievement when only local persistency is included, while the other 3 incorporate both historical predictors. Four out of the top 5 models with the best performance had the highest scores when only local persistency was included while one model also benefited from including the previous state of the node as the feature.

By looking at the Figure 4.5 for the second experiment, it is easy to see that the historical model has a higher probability to predict the positive class because it has higher scores. The blue line represents the ROC of the baseline model, while the orange line represents the ROC of the historical model. When the score is higher, it means that the model is more likely to make the correct prediction. The biweekly historical model has a much higher AUC score of 0.98 compared to the biweekly baseline model which has an AUC score of 0.92 which implies that the model better classifies unbalanced data.

Lastly, if we look at the F1 score for the second experiment, again, across historical and baseline models, it is almost always the same as accuracy values. The confusion matrices of these two best models KNN biweekly

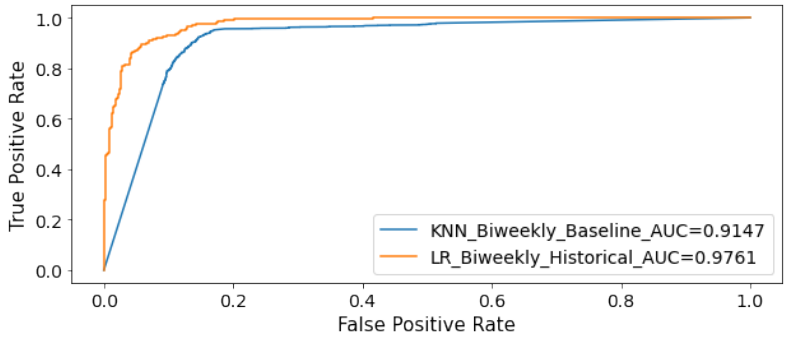


Figure 4.5: Experiment II: Best Baseline and Historical ROC and AUC

baseline and LR daily historical are analyzed on the Figure 4.6. Both models have more FP than FNs which is understandable due to the unbalanced data. Baseline KNN has slightly more FN than historical LR, and it also has more FP. There is a bigger difference in FPs, so from this fact comes that the historical model has somewhat higher accuracy because it predicts dry nodes better even though in general there are more wet nodes. Since we have unbalanced data where more nodes are wet than dry, it can be inferred that this imbalance leads the models to make wrong predictions.

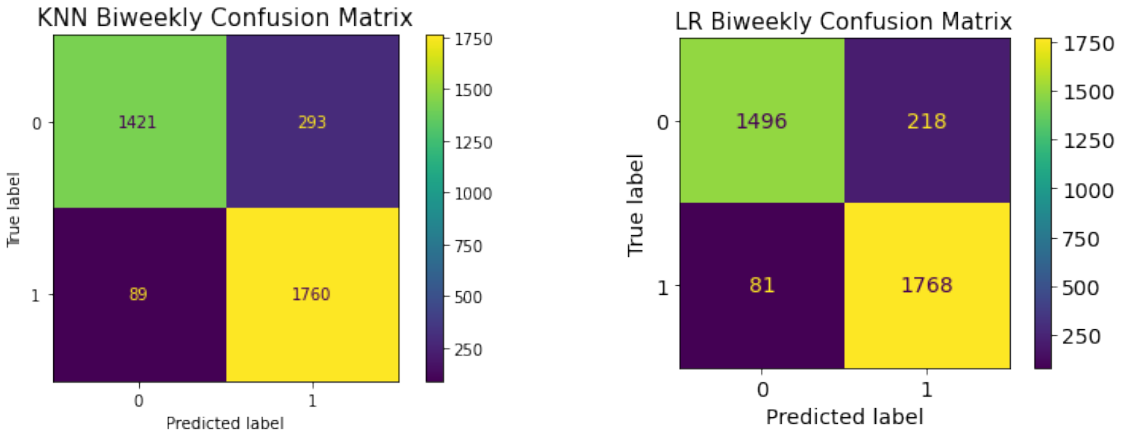


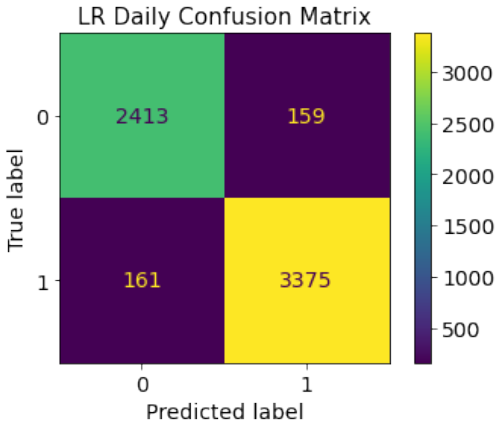
Figure 4.6: Experiment II: Best baseline and historical models confusion matrices

4.2 All Data and Models

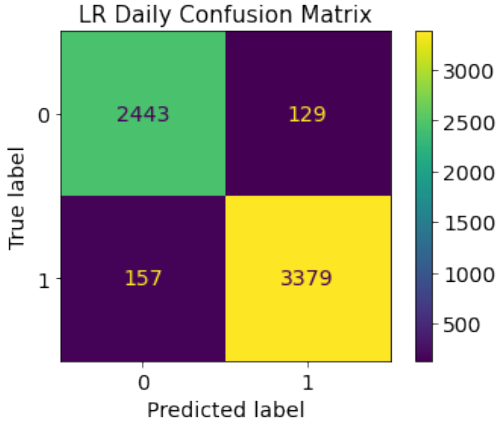
The Table 4.3 shows the best models for each dataset and for both experiments. In the first experiment and the second experiments, the daily LR predictor has the highest scores of 95% in both experiments. Even though the models have the same accuracy, precision, recall, and F1 scores are higher for the biweekly LR model in the second

experiment. In the first experiment, to have a more accurate prediction, all models included weighted averages of the current state of the node while the top 2 models with the best performance included also the weighted average of the previous state of the node. However, in the second experiment, all the best 5 models which were included in the table included both additional features. This implies that the model benefits from the assumption that the nodes behave similarly if they have similar local persistency. This information combined with the historical data affects the model to predict more accurately.

Because the dataset is not balanced, it is important that we evaluate the Figure 4.7 which shows the comparison of confusion matrices between 2 models with the overall highest accuracy for each experiment. Even though both models have really high numbers of correct predictions, the Figure 4.7a shows that the model predicted more FNs (161) in the first experiment and Figure 4.7c shows that the model predicted more FNs (143) also in the second experiment. However, the reason why the LR classifier in the second experiment makes a slightly more accurate prediction is that it has more TR and TN predictions overall even though it predicts more negative class when it is actually positive. Both models make more Type II errors, and the difference in the scores comes from the difference in the FP number because the difference between FN is only 4. This proves that the classifiers benefit from more samples in the training dataset because they can more accurately classify dry nodes given that there are more wet nodes in the dataset overall and with this make fewer FP predictions.



(a) Experiment I: Best Model Confusion Matrix



(b) Experiment II: Best Model Confusion Matrix

(c)

Figure 4.7: Best models confusion matrix comparison

ALL MODELS

Experiment I					Additional Features			
MODEL	ACC	PREC	REC	F1 SCORE	Local Persistency	Previous State	Weighted Avg. Current State	Weighted Avg. Previous State
LR Daily	0.95	0.94	0.94	0.94	✓		✓	✓
LR Biweekly	0.94	0.94	0.94	0.94	✓		✓	✓
RF Biweekly	0.94	0.93	0.94	0.94	✓	✓	✓	
DT Weekly	0.93	0.94	0.92	0.93	✓		✓	
KNN Weekly	0.93	0.92	0.93	0.92	✓	✓	✓	

Experiment II					Additional Features			
MODEL	ACC	PREC	REC	F-1 SCORE	Local Persistency	Previous State	Weighted Avg. Current State	Weighted Avg. Previous State
LR Daily	0.95	0.96	0.95	0.95	✓		✓	✓
KNN Biweekly	0.94	0.95	0.94	0.94	✓		✓	✓
RF Weekly	0.94	0.94	0.93	0.94	✓		✓	✓
KNN Weekly	0.94	0.94	0.93	0.93	✓	✓	✓	✓
KNN Monthly	0.94	0.94	0.93	0.93	✓	✓	✓	✓

Table 4.3: All Best Models

To understand which features are crucial, the feature importance of the best model among models and experiments will be analyzed. The Figure 4.8 shows the coefficients of each predictor that was used to train and test the best model which is the daily LR model from the second experiment. The plot is based on the model's coefficients, and if the model assigns a large negative or positive coefficient to each input value, the more influence that feature has on the model. It is evident that the most important feature among others is local persistency with a coefficient of around 3, whereas the least important is the weighted average of the previous state of the node with a coefficient close to 0. Also, a feature that is very important for accurate prediction is the weighted average of the current states of the neighboring nodes which has a coefficient close to 2.5. This feature has weights calculated based on local persistency similarity between the two nodes and includes the weighted average of neighboring nodes calculated for each node at each time step. Thus, the additional features representing spatio-temporal correlations among the nodes in the network and history were necessary to achieve good results. In particular, the correlations are based on the similarity of local persistency between pairs of nodes and states of the neighboring nodes. Precipitation and contributing areas have a much smaller influence on the model when all additional information is included.

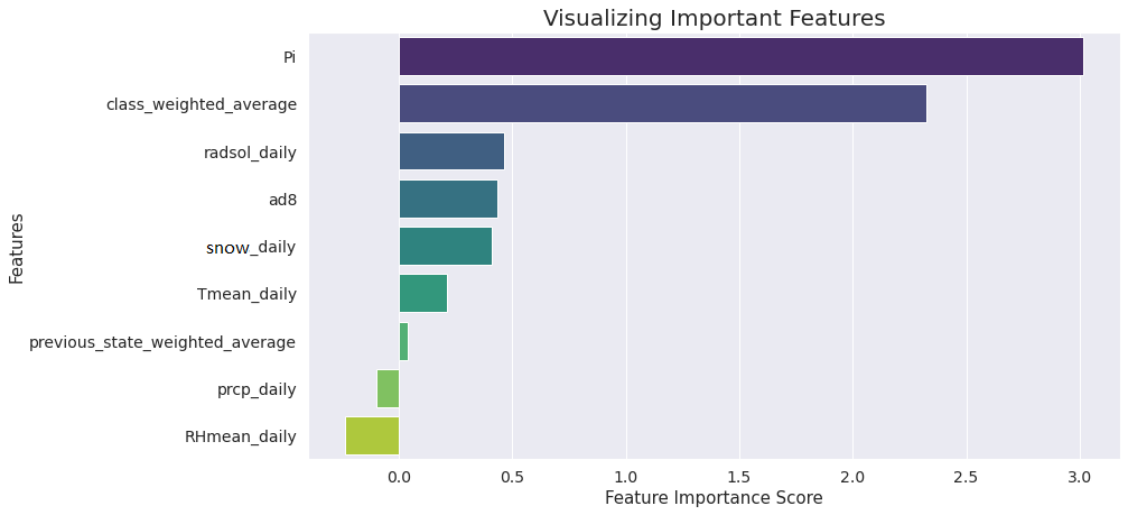


Figure 4.8: The feature importances for the best classifier among all models and experiments

The Figure 4.9 shows the ROC and AUC of the two best models in both experiments. The LR model in the second experiment does not only have slightly higher scores as shown in the Table 4.3, but it also has a higher AUC score. The orange line represents the ROC of the LR classifier from the second experiment while the blue line represents the ROC of the classifier in the first experiment. These good scores for the daily models mean that these models handle better the unbalanced data as they will assign a higher probability to the random label to be dry. This is the behavior that is important to have when dealing with unbalanced data and we see that both models handle unbalanced data well with a score very close to 1.

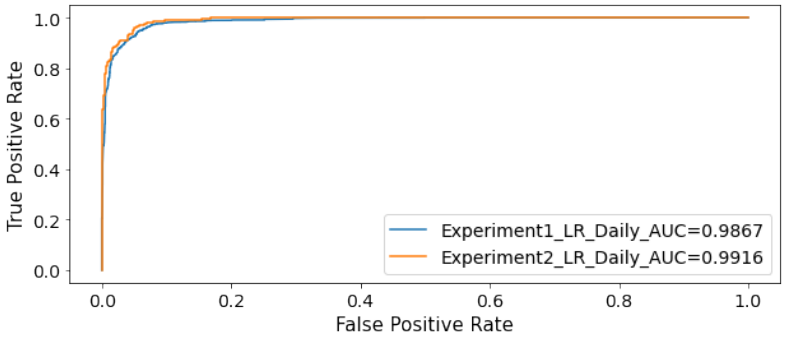


Figure 4.9: ROC curve for two best models in each experiment

5
Conclusion

At the core, machine learning is a set of tools that allows us to build and train models that extract and reproduce the spatial and temporal patterns in the datasets they encounter [30]. Different ML algorithms such as LR, DT, RF, KNN, and SVM were used to predict the dynamics of the Valfredda river network. The study catchment is specific because of the temporary streams which are defined by periodic flow cutoffs and may encounter limited or complete loss of water on the surface. The models were trained and tested on various combinations of features and different weather data approximations (daily, weekly, biweekly, and monthly). There were two main experiments and divisions: one in which the data from 2018 was used to train the model and predict the state of the node in 2020, and the second one where models were predicting the node state in 2021 using the data from 2018 and 2020. To evaluate the performance of these models, diverse scores and techniques were used and they include accuracy, precision, recall, F1 score, ROC and AUC score, confusion matrix, and feature importance coefficients. To improve accuracy, feature engineering was a crucial method because through it the models' performance was boosted. To include historical information previous state of the node and local persistency calculated using past information were introduced. Then, to include information about the graph, the weighted average of the state of the node and of the previous state of the node were included where the weight was based on the local persistency of the node. The main assumption is that the nodes with the same or similar local persistency behave in the same manner.

In general, across all datasets and experiments, models in the second experiment showed better performance. This can be related to the number of training samples since in the second experiment there were 19 observations while in the first experiment, there were only 7. Also when it comes to the testing dataset, data is more balanced in 2021 than it is in 2020 which means that the difference in the number of wet and dry nodes is higher in 2020. Additionally, among all models, the performance is proven to be better when additional features such as local persistency and weighted averages of the current and previous state of the nodes were included. The predictors with the highest coefficients given by baseline models are contributing area and precipitation, while when temporal and spatial data are included, local persistency has the highest score and is followed by weighted average of the

current state. The precipitation is assigned much higher coefficients by baseline models, but when historical and spatial predictors are included, it is assigned lower values because additional features have higher coefficients and influence models to make more accurate predictions. The majority of best models include local persistency and all spatial data, while the others include different combinations of these data. Models which include historical data usually have higher AUC scores and give a higher probability to the new random samples are dry. This is a very important characteristic of the best-performing models because the datasets are unbalanced and there are more wet nodes than dry ones. It is also evident that even though the data is unbalanced and there are more wet nodes, the models predict a larger number of dry nodes when the training dataset is larger.

In future research, different approaches should be considered. One of them may be to take all years together into consideration and look at each node separately. Also, the local persistency adjacency matrix which serves as weights for calculating weighted average could be calculated at each time-step and not approximated over all training datasets. Next, since it is evident that models benefit from a higher number of training samples, it should be considered to gather and use larger datasets. This will also be beneficial to explore further ML algorithms and include more complex deep learning models.

References

[1] O. McDonough, J. Hosen, and M. Palmer, "Temporary streams: The hydrology, geography, and ecology of non-perennially flowing waters," *River Ecosystems: Dynamics, Management and Conservation*, pp. 259–290, 01 2011.

[2] R. Stubbington, J. England, P. Wood, and C. Sefton, "Temporary streams in temperate zones: Recognizing, monitoring and restoring transitional aquatic-terrestrial ecosystems," *Wiley Interdisciplinary Reviews: Water*, vol. 4, 05 2017.

[3] N. Durighetto and G. Botter, "On the relation between active network length and catchment discharge," *Geophysical Research Letters*, no. 49, 2022.

[4] J. Prancevic and J. Kirchner, "Topographic controls on the extension and retraction of flowing streams," *Geophysical Research Letters*, vol. 46, 02 2019.

[5] H. Lange and S. Sippel, *Machine Learning Applications in Hydrology*, 02 2020, pp. 233–257.

[6] H. Ren, X. Song, Y. Fang, Z. Hou, and T. Scheibe, "Machine learning analysis of hydrologic exchange flows and transit time distributions in a large regulated river," *Frontiers in Artificial Intelligence*, vol. 4, 04 2021.

[7] L. Breiman, *Machine Learning*. Kluwer Academic Publishers, 2001, ch. Random Forests, pp. 5–32.

[8] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 161–168. [Online]. Available: https://doi.org/10.1145/1143844.1143865

[9] N. Durighetto, F. Vingiani, L. E. Bertassello, M. Camporese, and G. Botter, "Intraseasonal drainage network dynamics in a headwater catchment of the italian alps," *Water Resources Research*, no. 56, 2020.

[10] G. Botter, F. Vingiani, A. Senatore, C. Jensen, M. Weiler, K. McGuire, G. Mendicino, and N. Durighetto, "Hierarchical climate-driven dynamics of the active channel length in temporary streams," *Scientific Reports*, no. 11, 2021.

[11] S. W. Menard, "Applied logistic regression analysis (quantitative applications in the social sciences)," *WaterResources Research*, no. 56, 2001.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] M. Melek, N. Melek, and T. Kayikcioglu, "A novel simple method to select optimal k in k-nearest neighbor classifier," *International Journal of Computer Science and Information Security,*, vol. 15, pp. 464–469, 12 2017.

[14] H.-A. Park, "An introduction to logistic regression: From basic concepts to interpretation with particular attention to nursing domain," *J Korean Acad Nurs*, vol. 43, no. 2, 2013.

[15] M. Scott, *Applied Logistic Regression Analysis (Quantitative Applications in the Social Sciences)*, 2nd ed. Thousand Oaks, CA: Sage Publications., 2001.

[16] L. Breiman, *Classification and Regression Trees*. Routledge, 1984.

[17] L. Rokach and O. Maimon, *Data Mining and Knowledge Discovery Handbook*. Springer Science and Business Media, 2005.

[18] F. Sohil, M. Sohail, and J. Shabbir, "An introduction to statistical learning with applications in r: by gareth james, daniela witten, trevor hastie, and robert tibshirani, new york, springer science and business media, 2013, eisbn: 978-1-4614-7137-7," *Statistical Theory and Related Fields*, vol. 6, pp. 1–1, 09 2021.

[19] J. Quinlan, "Simplifying decision trees," *International Journal of Man-Machine Studies*, no. 27, 1987.

[20] G. Louppe, "Understanding random forests (from theory to practice)," Ph.D. dissertation, University of Liège, 2014.

[21] M. Steinbach and P.-N. Tan, *The Top Ten Algorithms in Data Mining*. Taylor and Francis Group, LLC, 2009, ch. kNN: k-Nearest Neighbors, p. 12.

[22] M. Azadkia, "Optimal choice of k for k-nearest neighbor regression," *arXiv: Statistics Theory*, 2019.

[23] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," vol. 2049, 09 2001, pp. 249–257.

[24] G. Wahba, *Spline Models for Observational Data*. SIAM, 1990, vol. 59, ch. Series in Applied Mathematics.

[25] M. Awad and R. Khanna, *Support Vector Machines for Classification*, 04 2015, pp. 39–66.

[26] V. Kecman, *Support Vector Machines – An Introduction*, 05 2005, vol. 177, pp. 605–605.

[27] S. Yang and G. Berdine, "The receiver operating characteristic (roc) curve," *The Southwest Respiratory and Critical Care Chronicles*, vol. 5, p. 34, 05 2017.

[28] K. Gajowniczek, T. Ząbkowski, and R. Szupiluk, *Estimating the ROC Curve and Its Significance for Classification Models' Assesment*, 2014, vol. 2, p. 382 – 391.

[29] V. Bewick, L. Cheek, and J. Ball, "Statistics review 14: Logistic regression," *Critical care (London, England)*, vol. 9, pp. 112–8, 03 2005.

[30] C. Shen, X. Chen, and E. Laloy, "Editorial: Broadening the use of machine learning in hydrology," *Frontiers in Water*, vol. 3, 05 2021.

Acknowledgments

I would like to thank my parents, my brother, and Antonije for their constant encouragement, unconditional support, and motivation in moments of doubt, even though I never achieved to fully explain my research. I would also like to thank my colleagues and dear friends Nora, Stefanija, and Sercan for their endless help, stimulating discussions, and insightful conversations.