ADVANCEMENT OF DATA-DRIVEN SHORT-TERM FLOOD PREDICTIONS ON AN URBANIZED WATERSHED USING PREPROCESSING TECHNIQUES.

MARINA ZISTLER

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ARTS

GRADUATE PROGRAM IN INFORMATION SYSTEMS & TECHNOLOGY

YORK UNIVERSITY

TORONTO, ONTARIO

July 2018

ABSTRACT

Supervised classification can be applied for short-term predictions of hydrological events in cases where the label of the event rather than its magnitude is crucial, as in the case of early flood warning systems. To be effective, these warning systems must be able to forecast floods accurately and to provide estimates early enough. Following the approach of transforming hydrological sensor data into a phase space using time-delay embedding, an attempt was made to improve the performance of the models and to increase the lead-time of reliable predictions. For this, the available set of attributes supplied by stream and rain gauges was extended by derivatives. In addition, imbalanced data techniques were applied at the data preprocessing step. The computational experiments were conducted on various data sets, lead-times, and years with different hydrological characteristics. The results show that especially derivatives of water level data improve model performance, increasingly when added for only one or two hours before the prediction time. In addition to that, the imbalanced data techniques allowed for overall improved prediction of floods at the cost of slight increase of misclassification of low-flow events.

DEDICATION

I would like to dedicate this thesis to my wonderful parents Klaus and Patricia, who have supported me throughout my academic career and always believed in me.

I would also like to thank my spouse, Mansour. I could not have finished this thesis without his incredible patience and encouragement.

A big thanks to my friend Sabine, who helped me proofreading and “hopefully” found all my spelling mistakes.

Additionally, I would like to thank Prof. M. Erechtchoukova for her support, effort, and guidance as my supervisor on this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

Machine learning as part of Artificial Intelligence is not only an area for scientific research (Breiman, 2001; ASCE, 2000 a; ASCE, 2000 b; Sebastiani, 2002) but also part of our daily lives. Modern spam filters such as used by Gmail apply machine learning techniques to detect spam emails while considering the personal preferences and characteristics of each user (Gmail, 2015). Financial institutions manage to determine whether a transaction is fraudulent or not without manually reviewing each of the millions of transactions that are made every day. The data analytics company FICO, for example, uses Neural Networks that assess whether a transaction is a fraud based on information such as transaction amount, transaction frequency, and transaction location (FICO, 2018).

Another area that machine learning has proven to be a viable technique for is the prediction of floods. Floods are one of the most severe natural catastrophes globally, causing both the loss of human life and economic damage. According to the Centre for Research on the Epidemiology of Disasters (CRED), floods caused an average of 5918 deaths and affected an average of 85,139,395 people each year between the years of 2005 and 2014. This ranks flood as the fourth highest disaster with regards to the cause of death and the highest for affecting human life. In addition to that, the CRED marked floods, such as the one in China in 1998 (caused $43 billion of damage) and the flood in Thailand in 2011 (caused $42 billion of damage), as disasters with the largest economic impact between 1980 and 2015 (CRED, 2016).

Flash floods are a severe form of floods with more than 5000 deaths annually and a mortality rate that is 4 times greater than any other type of flood (Jonkman, 2005). In the United States in 2016, 86 out of the 126 flood casualties were caused by flash floods accounting for about 68% of all flood casualties (National Weather Service, 2016). The American Meteorological Society defines flash floods as flooding caused by rapidly rising water level as a result of intense rainfall over a small area or because of moderate to intense rainfall over highly saturated or impervious land surfaces (AMS, 2000). Hapuarachchi et al. (2011) state that flash floods are mostly caused by excessive rainfall. Kelsch (2001) specifies the draining area for flash flood endangered regions as only a few hundred square kilometers. With the increasing intensity of rainfall in parts of the world and the increasing urbanization, Hapuarachchi et al. (2011) argue that flash floods will affect larger parts of the population. They generally occur within minutes to several hours of the rainfall event. This rapid onset limits the opportunity for an effective response and entails additional challenges for the prediction of flash floods. Another complicating factor is that for many of the small catchments not much information about the physical characteristics of the watershed is available. This limits the application of physically based models that require this information in form of parameters.

Early warning systems are part of an effective Flood Risk Management, which helps to improve the flood preparedness and, as a result, reduce the risk and impact of such. According to Plate (2002), a warning system must be able to forecast floods accurately and provide estimates early enough to mitigate the risk effectively. The United Nations (2004) agree that both forecast timing and accuracy are central for measuring the effectiveness of early warning systems and call early warning systems a cornerstone of disaster reduction. The most crucial information that operational flood early warning systems need to provide is whether a flood will occur at a specific location of interest. This enables officials to respond to hydrological events in an efficient fashion. Rather than predicting the actual magnitude of hydrological characteristics, each event can then be assigned a class label indicating whether it was a 'high' flow (flood) or 'low' flow (no-flood) event. This calls for the application of supervised machine learning algorithms and, specifically, classification algorithms. Predictions should be made as early as possible while ensuring at least 80% accuracy when predicting a flood. The model must be able to produce reliable predictions with input from only a few hydrological and meteorological sensors. This accounts for the limitation that small watersheds, prone to flash floods, are often poorly gauged.

How machine learning is applied depends on the problem to address. In the case of rainfall-runoff modeling the following considerations must be taken into account. The underlying processes are highly complex and exhibit nonlinear dynamics as they are influenced by many natural and anthropogenic factors such as soil moisture, land use, watershed geomorphology, evaporation, infiltration, distribution, and duration of the rainfall. The data collected by hydrological and meteorological sensors such as stream gauges or rainfall gauges can be represented as a time series. The subsequent elements of this time series are dependent on some preceding ones because earlier events may affect later events. Heavy rainfall, for example, causes a later rise in water level. A rise in water level at an upstream location will eventually travel downstream affecting the sensor results at the downstream locations. Another characteristic of data sets used for rainfall-runoff modeling is that they are often imbalanced. This means that there is only a small percentage of records representing the event that needs to be predicted, like a flood or draught, compared to the total number of records. Missing such an event like a flood can have serious consequences, while declaring a non-flood event as a flood event is less severe. Of course, too many of these false positives can cause the population to lose trust in the flood forecasting system, so the accurate prediction of non-flood events must not be ignored. As a result of the characteristics mentioned above, rainfall-runoff modeling shares many common issues with other application areas of machine learning. In the financial sector, machine learning is used for the characterization and prediction of complex, nonstationary time series events from the financial domain such as finding a trading-edge or stock market predictions (Povinelli, 2001). Machine learning can be used in the healthcare sector, for longer-term follow-up diagnosis, to determine if and when a transplant patient's body starts rejecting the new organ and to determine when to start a treatment with

immunosuppressive drugs. Another application area are cases in which patients have been admitted to the hospital following an accident and are monitored for organ failures (Lin et al., 2008).

This study presents the findings of two approaches to improve the methodology for the application of classification machine learning algorithms to hydrological predictions. The first approach is to extend the list of potential variables by adding derivatives of water level and rainfall magnitudes. Based on knowledge from the hydrological domain, the assumption is that, along with magnitudes of water level and rainfall, their rates of change over time also carry important information about current and future hydrological conditions at an investigated watershed. Rainfall-runoff processes display nonlinear dynamics. As a result, actual magnitudes observed at discrete moments are not sufficient to describe the necessary details of the underlying hydrological processes. The rate of change in magnitude, however, adds information that allows modelling this relationship. The rate of change in a process is mathematically described as the derivative of a variable. However, in the case of hydrological predictions, data of continuous processes such as rainfall or water level are available in the discrete form of a time series. As a result, an approximation using finite differences between the magnitude of one timestamp and that of the previous timestamp was implemented. These approximations use the smallest possible timestep, corresponding to the observation time interval and are further referred to as deltas. The effects of adding those deltas on the prediction ability of generated models are investigated for different lead-times and classification algorithms using exploratory computations. For the second approach, different imbalanced data preprocessing techniques, in the form of over-sampling and under-sampling techniques, are applied on data sets with and without the deltas and the same lead-times and algorithms. The results of those techniques are evaluated compared to each other and to the baseline. The investigations are conducted as a case study on a small stream – Spring Creek, Ontario, Canada. The results show that deltas can improve single classifier and classifier ensemble performance and allow extending the lead-time for hydrological predictions. The applied over-sampling and under-sampling techniques show that they can increase the ability of the classifier or classifier ensemble to identify flood events but introduce a higher rate of false alarms.

2 LITERATURE REVIEW

The chapter covers the theoretical background used for this research as well as its practical applications where appropriate. It first discusses the theories and methods used for machine learning in the context of hydrology and then generic machine learning topics such as classification algorithms, variable selection, and techniques for imbalanced data sets.

2.1 MACHINE LEARNING IN HYDROLOGY

Determining the relationship between rainfall and runoff for a watershed is an important problem that many hydrologists and engineers face. Although many watersheds have been gauged in the past to provide detailed records of stream flow and other information, hydrologists and engineers are often confronted with situations where little to no information is available (ASCE, 2000 b). As a result, over the past 60 years, the nonlinear relationship between rainfall and streamflow has received considerable attention from the academic world, leading to mathematical and computer-based models that range from complex physically based models to black-box representations (Young, 2003).

2.1.1 CLASSIFICATION OF RAINFALL-RUNOFF MODELS

Physically based models determine the relationship between rainfall and runoff based on the physical representation of the watershed using several physical parameters such as the digital elevation model (DEM), land-use, soil maps, and soil characteristics.

Lumped models treat the basin as one uniform entity, not accounting for the differences between the different areas of the watershed. This means that the model input and the model parameters are only considered on a global basin level. As a result, the limitation of the lumped models is their coarse resolution. However, lumped models are still used in operational flood forecasting systems due to their simplicity, computational efficiency, and lower data requirements (Hapuarachchi et al., 2011). Examples of these lumped hydrological models are the Stanford watershed model, a pioneer work by Crawford and Linsley (1966), and the HBV model by Bergström (1976) developed for Scandinavian catchments.

Due to the increased amount of data available, spatially distributed models, which are the result of the effort to overcome the limitations of the lumped models, gained popularity. While semi-distributed models divide the basin into sub-basins, fully distributed models discretize the basin in small units typically in grid form. Ciarapica and Todini (2002), for example, advanced the original TOPKAPI model that uses

evapotranspiration, snowmelt, soil water, surface water and channel water (Todini, 1995) into a distributed model and then added interception, infiltration, percolation, groundwater flow, and lake/reservoir routing. As a result, the new model requires input from 25 hydrological parameters. The huge computational effort of their model did not allow for any real-time application, but this will be part of their ongoing research (Liu et al., 2005). These distributed hydrological models overall provide better performance than the lumped models and can better utilize distributed rainfall input as well as represent the catchment's physical characteristics (Hapuarachchi et al., 2011). Figure 1 shows a schematic representation of the three physically based models to demonstrate their differences. The grids are drawn for demonstration purposes only and do not represent the actual divisions of a watershed into compartments.

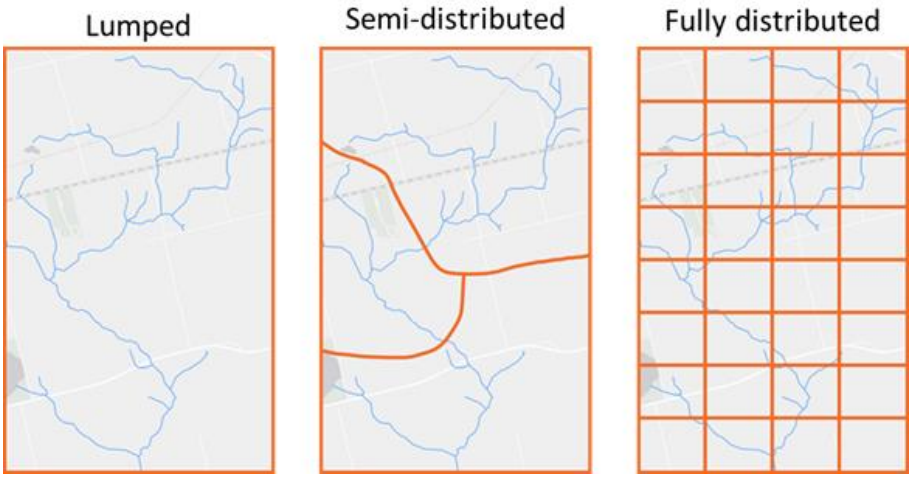

Figure 1*: Types of physically based models*

Data-driven modeling techniques use statistical relationships obtained from rainfall and river flow data to generate flow forecasts (ASCE 2000 a; ASCE 2000 b; Eagleson, 1972). These techniques have been expanded with the application of stochastic and data-mining algorithms resulting in black-box models. Black-box models try to find a relationship between historical inputs such as rainfall or temperature, and outputs such as watershed runoff or water level (ASCE, 2000 a). They produce a transfer function which can be applied on the input data to receive the desired hydrological output information (Erechtchoukova et al., 2016).

Artificial Neural Networks (ANNs) are a mature black-box model approach to hydrological problems. Their application to streamflow predictions has been heavily researched (Aqil et al., 2007; Hu et al., 2001) resulting in the ASCE Task Committee on Application of ANNs in Hydrology to create standards and frameworks on how to use ANNs in hydrology (ASCE 2000 a; ASCE 2000 b). Over the last few years, they evolved into various hybrid schemes such as Bayesian ANNs (Humphrey et al., 2016), neuro-fuzzy systems (Nayak et al., 2005), and deep learning tools (Li et al., 2016). Other common techniques used for

rainfall-runoff modeling are regression methods such as the M5 algorithm used by Solomatine and Dulal (2003) and genetic programming. Genetic programming is an evolutionary computing method that creates a structured representation of the rainfall-runoff system that can be understood by humans and offers additional information about the relationship between rainfall and runoff that would otherwise not be visible (Savic et al., 1999; Whigham & Crapper, 1999). For the black-box models depending on whether regression or classification algorithms are being used, the model will then either return the magnitude of the hydrological characteristics such as the water level or the class of the hydrological event such as whether a flood event or a non-flood event occurred.

The advantage of black-box models over physically based models is that they do not require extensive information about the characteristics of the watershed in the form of hydrological parameters. Obtaining these hydrological parameters can be a long and expensive process. On top of that, once the characteristics of the watershed change, for example when a new road is built next to the river, the physically based model might not function anymore, and the parameters need to be collected again. The data-driven models, on the other hand, require a larger amount of data. However, the data they require are often easy to acquire using hydrological sensors (e.g. rain gauge) or is already available (e.g. meteorological information). This gives them the ability to perform better in cases were the underlying complex system is poorly understood, noise is present, or the input is incomplete or ambiguous (Tokar & Johnson, 1999). Nevertheless, data-driven models also hold disadvantages. For instance, a large set of historical data are necessary to train the black-box model, so that its predictions are accurate enough. Physically based distributed hydrological models on the other hand mostly use parameters that have physical interpretation and therefore can be calibrated on relatively short records (Hapuarachchi et al., 2011). Another disadvantage of black-box models is as identified by the ASCE Committee the lack of physical concepts and relationships (ASCE, 2000 a).

Research is undertaken in how to combine both physically based and data-driven models to obtain the advantages of both while mitigating the disadvantages. These hybrid models are aiming to represent the physical characteristics of the catchment as well as spatial variations while realizing the ability of nonlinear mapping and pattern recognition (Corzo Perez, 2009; Chen & Adams, 2006). An example of this research is the rainfall-runoff forecasting procedure developed for the Apennines Mountains in Italy. This model achieved optimal results by using short-term rainfall forecasts obtained by ANNs from past rainfall magnitudes as the only input information. The short-term rainfall forecast is then routed through a physically based rainfall-runoff model to predict flooding (Toth et al., 2000). Another example is the research conducted by Chen and Adams (2006) that integrated ANNs with three different types of conceptual models. In this case, the catchment is divided into three sub-catchments. For each of them, the conceptual semi-distributed model is being applied and generates the runoff for the specific sub-catchment.

This results in three outflow outputs, which are then used as the inputs for the Artificial Neural Network that combines and weights them into a final output.

To find the best model for a given watershed, data availability, the complexity of the underlying hydrological processes, temporal scales, spatial scales, and the required output of the model must be taken into consideration.

2.1.2 RELATED WORK

This study aims to complement the wider on-going research to improve flood predictions generated by a classification algorithm trained on a phase space reconstructed from observation data.

The research conducted by Damle and Yalcin (2007) uses time series data transformed into a phase space. However, the predictions were based on daily discharge magnitudes and an unsupervised clustering algorithm was used to predict the flood and non-flood events. Segretier et. al. (2012; 2013) aim to find a transformation of the observation data that provides better prediction results. They do so by applying an evolutionary algorithm that searches for the best set of predictive variables and their derivatives. Later, the algorithm searches for the best classifier juries that classify the tuples using the majority vote. Furquim et al. (2014) applied machine learning to signals generated by wireless sensor networks, seeking to issue real-time flash flood alerts and 5-, 10- and 15-minute forecasts. Instead of simply dividing tuples into flood and no-flood events, they use the five class labels high increase, low increase, stable, low decrease, and high decrease.

The sensors producing the data sets for rainfall-runoff predictions can have significant uncertainty associated with them. Rainfall gauges contain uncertainty in their calibration, and in their tipping mechanism. They are a point measurement and assume a uniform distribution of rainfall in the respective area (McCulloch et al. 2008; Han et al. 2002). Habib et al. (2001) measure the sampling errors of tipping-bucket rain gauge at several time scales and found significant errors if the measurements are based on time scales less than 10 to 15 minutes. Because of these shortcomings, research is being conducted that aims to mitigate sensor uncertainty. McCulloch et al. (2008) researches the application of the Linguistic Decision Tree Induction Algorithm (LID3) to real-time flood forecasting. The model uses station water level and rainfall gauge measurements. They argue that both measurements have significant uncertainty associated with them. That is why instead of using the actual measured values they are labeled as high, normal, and low while at any point in time a measurement can have an association with two labels. The LID3 is applied to those linguistic labels. To ensure good performance over time they update the LID3 using an error-based approach and extend the domain of target focal elements. Han et al. (2002) use a fuzzy decision tree for

stream flow modeling. The trees are learned from data using the MA-ID3 algorithm, which allows using fuzzy attributes and class values for decision trees and provides a framework for representing linguistic rules. The data are first transformed into fuzzy sets. For rainfall, for example, the rainfall values are split up into five overlapping fuzzy sets described by labels such as very low, low, medium, high, and very high. The fuzzy ID3 algorithm is used to create a decision tree based on the fuzzy sets. Even though the MA-ID3 algorithm does not return results that are as good as the Neural Network benchmark, it allows to gain insight into the hydrological processes which is not possible with the Neural Network modeling approach. In the case of Han et al. (2002), the fuzzy decision tree showed that the rainfall values of four or five days before the prediction time were considered as more informative for the prediction than the more recent ones.

Using classifier ensembles is another area of machine learning that is being investigated to see if it can improve the prediction results of black-box rainfall-runoff models. The reasoning behind this is that combining multiple classifiers can reduce the generalization error and build a model that is more robust and returns better predictions than a single classifier. However, each classifier in the ensemble must have an error rate that is better than random guessing and the classifiers must be diverse, so they disagree with their prediction on the same tuples (Dietterich, 2000; Hansen & Salamon, 1990). Erechtchoukova et al. (2016) use synchronized time series precipitation and water level data transformed into a phase space based on a time delay embedded technique. This allowed for flood predictions using data from the recent past. An ensemble of five inducers was selected that determine the final class label by majority vote. This research resulted in a model that was able to generate 45-minutes and 60-minutes flood predictions with a Precision of more than 80%.

2.1.3 SCALING AND SYNCHRONIZATION OF HYDROLOGICAL DATA SETS

Depending on the nature of an observed hydrological parameter, the used sensor, and the responsible organization, the variables available for the rainfall-runoff modeling could be collected with different time granularities. In this case, scaling is used to synchronize the different variables of the data set.

Hydrological data sets can be described using the process scale, observation scale, and modeling scale. The process scale describes the lifetime (duration) of an intermittent process such as a flood, the period (cycle) of a periodic process such as snowmelt and the correlation length (integral scale) for a stochastic process displaying some sort of correlation. The observation scale describes the temporal extent: how long samples were recorded for or the finite number of samples, the resolution between samples describing the time between measurements, and the time constant of a sample showing the duration of a measuring process resulting in a single measurement.

Figure 2 shows three alternative definitions of the observation scale in space and time, according to Blöschl and Sivapalan (1995).

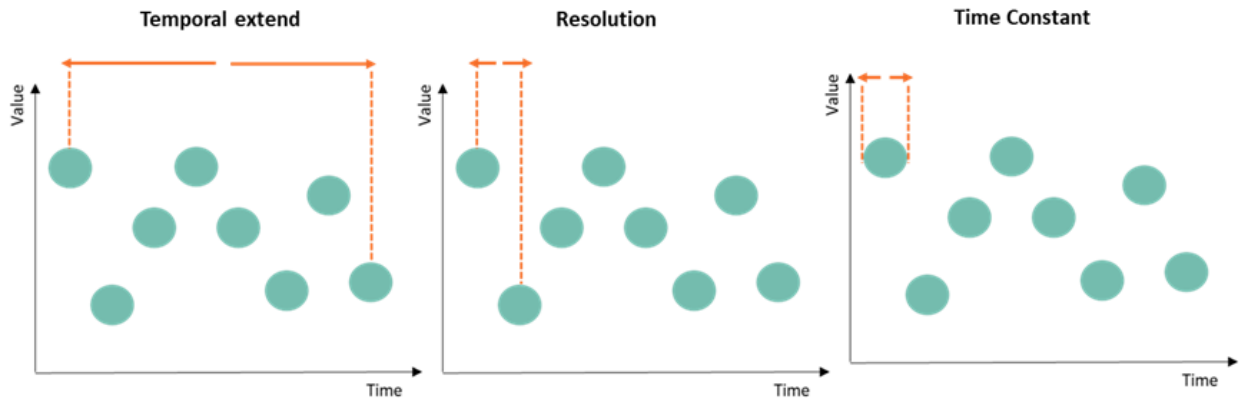

Figure 2: *Observation scale*

The modeling scale is only partly related to the hydrological process and depends on the hydrological model applied. Ideally, processes should be observed at the scale they occur, so the observation scale should be the same as the process scale. However, this is not always feasible because often only small-scale point samples are available and the same data sets are being used for different applications. In this case, scaling is necessary to bridge the gap between the process scale and modeling scale. In hydrology, upscaling is the process of transferring from a given scale to a larger scale while downscaling is the process of transferring from a given scale to a smaller scale (Blöschl & Sivapalan, 1995).

2.1.4 TIME-DELAY EMBEDDING

To account for the dynamic nature of the underlying hydrological processes, the classification algorithms can be applied on a phase space, reconstructed from the collected time series using time-delay embedding. A reconstructed phase space is a Q-dimensional metric space in which a time series is unfolded. It is a vector space for the system so that specifying a point in this space specifies the state of the system at any given moment. In this case, it specifies the state of a sensor at a certain timestamp t (Povinelli & Feng, 2003; Damle & Yalcin, 2007). According to Takens (1980), the phase space is homeomorphic to the state space that generated the time series if Q is large enough. The Takens' Theorem guarantees that the dynamics of the reconstructed phase space are topologically identical to the true dynamics of the system if the embedding is performed correctly (Povinelli & Feng, 2003). The following section shows how to create a phase space using the time-delay embedding technique.

Hydrological data collected from an observation site *k* can be represented as a time series:

$$Y_k = \{y_{k,t}, t = 1, \dots, N\} \tag{1}$$

A reconstructed phase space can be created from this collected data using the time-delay embedding, which extracts *R* successive observations from the same observation site to create an element of this phase space:

$$x_{k,t} = (y_{k,t-(R-1)\tau}, y_{k,t-(R-2)\tau}, \dots, y_{k,t-2\tau}, y_{k,t-\tau}, y_{k,t}) \tag{2}$$

with τ being the time interval between the measurements (resolution) (Povinelli & Feng, 2003; Abarbanel, 2012; Erechtchoukova et al., 2016). As the occurrences of floods depend on several factors, the time-delay embedding approach was extended following Erechtchoukova et al. (2016) to contain measurements from multiple observation sites. This creates an element of a phase space that is constructed by concatenation of tuples $x_{k,t}, k = 1, \dots, M$:

$$z_t = (y_{1,t-(R-1)\tau}, \dots, y_{1,t}, y_{2,t-(R-1)\tau}, \dots, y_{2,t}, \dots, y_{M,t-(R-1)\tau}, \dots, y_{M,t}) \tag{3}$$

with the *M*-th observation site being the cross section of interest.

To provide a class label that allows the classification algorithms to connect a temporal pattern from past and present with a future event, an event characterization function is being used. This event characterization function represents the class of the future event for the present time index. What event characterization function to use is defined *a priori* and addresses the specific goal of the investigated problem (Povinelli & Feng, 2003). For the investigated problem, the event characterization function is used to differentiate between high flow (flood) and low flow (no-flood) events. Whether a high flow or low flow event occurs is determined by the observed magnitude at the cross-section of interest with an empirically derived threshold value H_{thresh}.

$$f_k(x_{k,t}) = \begin{cases} 'high', & y_{k,t} \geq H_{thresh} \\ 'low', & y_{k,t} < H_{thresh} \end{cases} \tag{4}$$

The augmented phase space follows from the definition of the phase space and the event characterization function. It is a *Q+1*-dimensional space resulting from the phase space being extended by the class label. The goal is to predict the event with a specified lead-time that can be defined as $i * \tau$ with i being the number of time steps that correspond the to the specified lead-time.

Therefore, the class labels can be assigned as follows:

$$s_t = \left(z_t, f\left(y_{M,t+i*\tau}\right)\right) \tag{5}$$

with s_t being an element of the augmented phase space. This augmented phase space now forms a set of samples, which can be divided into a training set, to develop the classification model, and into the testing set, to evaluate the model's performance (Povinelli & Feng, 2003; Erechtchoukova et al., 2016).

The concepts of time-delay embedding, and augmented phase space have not only been used for hydrological predictions (Erechtchoukova et al., 2016; Damle & Yalcin, 2007) but also for other areas such as the prediction of metal droplets from a welder (Povinelli & Feng, 2003) and for characterizing and predicting complex, nonstationary time series events from the financial domain such as finding a trading-edge and stock market predictions (Povinelli, 2001).

2.2 CLASSIFICATION ALGORITHMS

Multiple different classification algorithms were used for the experiments. They differ from the splitting criteria they use, whether it is a univariate or multivariate splitting criterion, and how they are being pruned and optimized. Based on the 'no free lunch theorem' it can be expected that each algorithm will perform differently on the same data set and that there is not one algorithm that will outperform all other algorithms consistently even on data sets of the same problem domain and that if one algorithm performs well on a certain class of problems it might not perform well on other problems (Wolpert, 1996).

Ross Quinlan proposed the C4.5 algorithm in 1993 as an extension of the ID3 algorithm. C4.5 is a classification algorithm that creates a top-down classification tree using the 'divide and conquer' approach. It uses univariant splitting criteria and chooses at each non-leaf node of the tree one attribute that splits the data set into two subsets while maximizing the information gain creating a binary tree. Subsequent pruning is used to mitigate tree overfitting (Quinlan, 1993). J48, the Weka implementation of the C4.5 classification algorithm, was used for the experiments in this study (Weka, 2016 a).

Breiman et al. (1984) created the CART algorithm, which also uses univariant splitting criteria. It is a decision tree algorithm that uses a binary split based on the Gini index meaning that the attribute is chosen for the split that provides the lowest Gini index and as a result creates the purest class for each leaf of the tree (Breiman et al, 1984). SimpleCart is the Weka implementation of the CART algorithm that uses minimal cost-complexity pruning and was used for this study (Weka, 2017 a).

The Reduced Error Pruning Tree (REPTree) creates multiple decision or regression trees in different iterations. At the beginning of the model preparation, REPTree sorts the values of numeric attributes once. Each tree is built using one attribute only for the split and information gain as the splitting criteria resulting in a binary split. After all trees have been created, it selects the best tree. REPTree uses reduced error pruning with back fitting. During reduced error pruning for each non-leaf subtree, the algorithm checks if replacing the subtree with a leaf node would result in the same or fewer errors. If this is the case, this leaf-node replaces the subtree. Otherwise, the subtree remains. This process is repeated until further pruning would increase the error rate (Witten & Frank, 2000; Quinlan, 1987).

Random Forest is a classifier consisting of a collection of decision trees. Each of these single decision trees is based on random vectors that are identically distributed amongst them causing each decision tree to vary from the others. Each tree is grown on only a subset of the training set with the tuples selected at random with replacement. For each split, the tree can only choose from a subset of all variables that are selected at random. In addition to that, no pruning is used causing the tree to be as large as possible. Then each decision tree casts a vote for the most popular class for a specific input tuple and the class with the majority of the votes is chosen as the final output (Breiman, 2001). Weka's implementation of Random Forest, which is based on the implementation by Leo Breiman, was used for this study (Weka, 2016 b).

NBTree is a hybrid algorithm with a similar structure as a decision tree but each leaf node contains a Naive-Bayes categorizer instead of a single class label. The Naive-Bayes categorizer uses Bayes rule to compute the probability of each class. To decide if and how to split a node the data are discretized and the accuracy of the estimate using Naive-Bayes at the node is calculated using 5-fold cross-validation. The overall utility of the split is then the weighted sum of the utility of the node where the weight is proportional to the number of instances that go down to that node. The NBTree classifier is, therefore, trying to estimate whether the generalization accuracy of Naive Bayes at each leaf is higher than a single Naive Bayes classifier at the current node. A split will only be initiated if the split results in a relative error reduction that is greater than 5% and if there are at least 30 instances in the node (Kohavi, 1996).

The ripple down rule learner, Ridor, first generates a default rule in Weka and after that creates exceptions for the default rule with the least error rate. Then it creates the best exceptions for each existing exception and repeats the process until pure (Weka, 2017 b). The exceptions are created using the ripple-down approach. Ripple-down rules form a decision tree which differs from standard decision trees in that compound clauses are used to determine branching, making Ridor a multivariate inducer. Another difference is that each rule clause does not need to exhaustively cover all cases, making it possible for a tuple to be classified at an interior node and not just at the root node as with decision trees. However,

similarly to standard decision trees, only one decision node is activated for each case (Gaines & Compton, 1995).

JRip is the result of the effort of Cohen (1995) to improve the Ridor algorithm further. It consists of three phases. The first phase is the growing phase in which conditions are greedily added to the rule until the rule achieves 100% accuracy. While doing that, the algorithm tries every possible value of each attribute and then selects the condition providing the highest information gain. In the next phase, the pruning phase, each rule is being incrementally pruned. After the first two phases have been repeated, resulting in an initial ruleset, the last phase, the optimization phase, is used to create a final rule-set with the shortest description length (Cohen, 1995; Weka, 2017 c).

2.3 VARIABLE SELECTION

In cases where a large number of variables are available, variable selection is used to decrease the risk of overfitting, provide faster and more cost-efficient predictors, and to improve data understanding (Guyon & Elisseeff, 2003). How to select the relevant variables and eliminate the irrelevant variables is a central problem in machine learning and has been researched for many years (Blum & Langley, 1997). Over the past years, the number of variables available for machine learning problems has increased calling for new techniques (Guyon & Elisseeff, 2003).

Machine learning algorithms respond differently when facing many variables that are not necessary to predict the output correctly. Top-down decision tree algorithms such as the C4.5 and the CART algorithm are known to degrade in performance when faced with many irrelevant variables due to the way they build and prune the decision tree. Other algorithms such as the Naïve-Bayes are more robust towards irrelevant variables but degrade in performance when introducing correlated variables (Kohavi & John, 1997; Langley et al., 1992). One of the problems of variable selection is closely related to the bias-variance dilemma. This dilemma describes the trade-off between the bias, which are errors resulting from wrong assumptions of the classifier also known as underfitting, and the variance, which are errors due to fluctuations in the training set also known as overfitting. These two types of errors are then the two components of the estimation error (German et al., 1992). In the context of variable selection, this means that there is a trade-off between estimating based on more variables (bias reduction) and correctly estimating these parameters (variance reduction). The second problem is that finding the actual best hypothesis rather than the best estimate adds computational complexity and is NP-hard. Therefore, to find the optimal subset of variables for a specific classification algorithm requires to consider its heuristics, biases, and trade-offs (Kohavi & John, 1997).

As the core of variable selection is to find and select relevant features, the definition of relevance becomes important. Blum and Langley (1997) and Kohavi and John (1997) discuss the different definition of relevance in the context of variable selection as well as the difference between relevance and usefulness. They both agree that there are multiple definitions of relevance depending on what the variables are relevant to. One definition, for instance, is defining variables that are relevant to the target, which is the class label in the case of classification:

"A feature x_i is relevant to a target concept c if there exists a pair of examples *A* and *B* in the instance space such that *A* and *B* differ only in their assignment to x_i and $c(A) \neq c(B)$." (Blum & Langley, 1997).

While this definition allows for the theoretical analyses of classifiers or other machine learning algorithms it has practical drawbacks. If the classifier, for example, is limited to a certain sample set it might not be able to determine whether a certain feature is relevant using the definition above (Blum & Langley, 1997). With respect to the practical nature of this case study as well as the fact that the data set is limited to two years of data, the following definition of relevance, which Blum and Langley (1997) name incremental usefulness, will be used in the context of this study:

"Given a sample of data *S*, a learning algorithm *L*, and a feature set *A*, feature x_i is incrementally useful to *L* with respect to *A* if the accuracy of the hypothesis that *L* produces using the feature set $\{x_i\} \cup A$ is better than the accuracy achieved using just the feature set *A*." (Blum & Langley, 1997)

Both, Kohavi and John (1997) along with Blum and Langley (1997), differentiate between strong and weak relevance. A feature is strongly relevant if it cannot be removed from the feature set without causing a decrease in model performance. On the other hand, a feature is weakly relevant if it is not strongly relevant but there is a subset of features that show better model performance with this feature than without. If a feature is neither strongly nor weakly relevant, the feature is irrelevant. As a result, strongly relevant features must be included in the feature set while weakly relevant features are optional and irrelevant features should be discarded (Kohavi & John, 1997; Blum & Langley, 1997).

While the above is true for the optimal Bayes classifier, machine learning algorithms that are trained on actual data sets do not have access to the underlying distribution of the data set and often use a restricted hypothesis space and cannot utilize all available features. This makes them suboptimal compared to the optimal Bayes classifier. As a result, relevance alone no longer determines whether to keep a variable but whether the variable is in the optimal feature subset. As Kohavi and John (1997) explain in their review article, the relevance of a variable does not imply its optimality and vice versa.

Variable selection can be represented as a heuristic search. During this search, each state in the search space represents a subset of possible variables. Figure 3 shows an example of such search space according to Blum and Langley (1997) with each state in the space specifying the attributes to use.

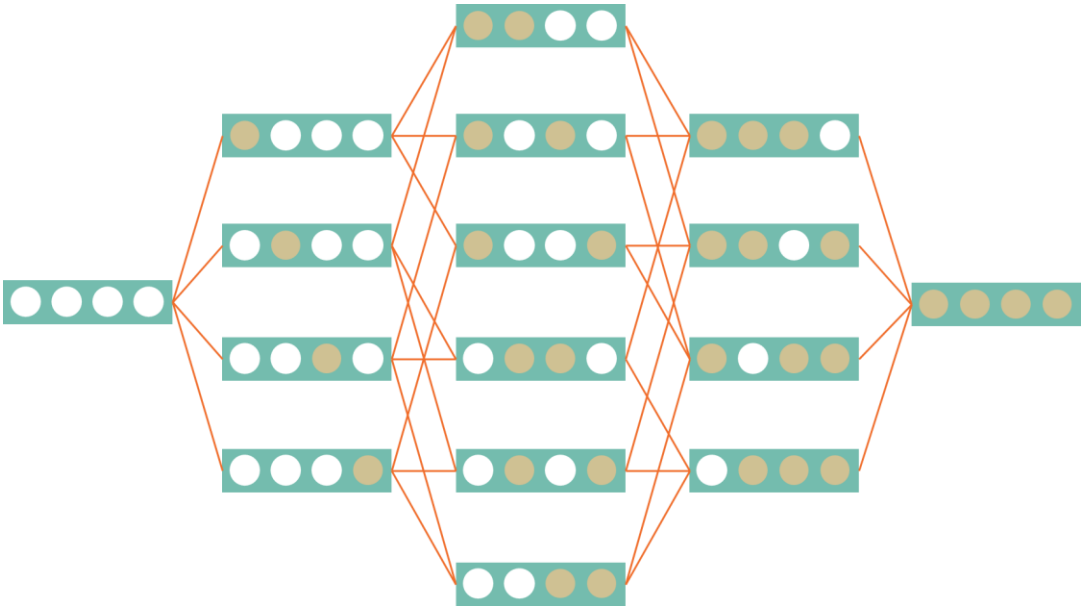

Figure 3: *Variable selection as a heuristic search*

The nature of the search is determined by the starting point of the search, the organization of the search, the strategy used to evaluate alternative subsets of variables and the stopping criteria (Blum & Langley, 1997).

Methods for variable subset selection can be divided into filters, wrappers, and embedded methods. The filter approach uses the variable selection as a preprocessing step (Kohavi & John, 1997) and, as a result, computes the evaluation measure for the variable selection directly from data without considering the predictor architecture that will later be applied to the created subset (Reunanen, 2003). For that reason, Taormina and Chau (2015) title this approach the model-free approach. Figure 4 shows a schematic overview of the filter approach according to Kohavi and John (1997).

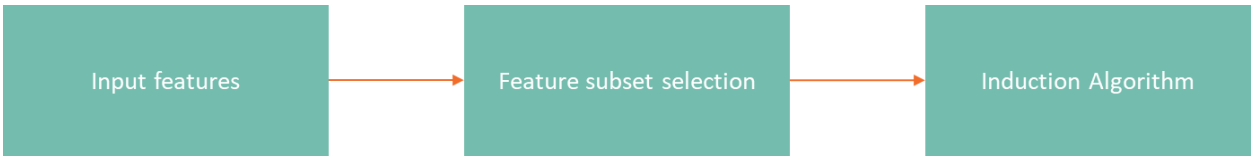

Figure 4: *The filter approach for feature subset selection*

The relevance of the input variables is estimated through statistical measures. The FOCUS algorithm is just one of multiple filter algorithms. It scans all available variables and selects a subset where there are no

examples in the subset that agree on all the features but have a different class label (Almuallim & Dietterich, 1991). The Relief algorithm (Kira & Rendell, 1992) estimates the relevance of a feature by randomly sampling instances from the training set and then calculating the difference between a selected instance and the two nearest instances of the same and the opposite classes. If the value of a variable shows differences between the selected instance and its two nearest instances while sharing the same class, then the relief score decreases for this variable. If a difference is observed and the class of the selected instance differs from the class of the two nearest instances, then the relief score increases.

The filter approach usually shows good generalization capabilities, as they are independent of the chosen data-driven model. As training a model is not necessary to find the variable subset for this approach, filters show high computational efficiency and consider the physical underlying processes of the problem they are applied to (Taormina & Chau, 2015). Their disadvantage is that they ignore the effect of the selected feature subset on the performance of the induction algorithms (Kohavi & John, 1997; Taormina & Chau, 2015). This disadvantage is addressed in the wrapper and embedded approach that is also known as model-based approaches (Taormina & Chau, 2015).

Using the wrapper approach (John et al., 1994), the feature subset selection is performed using the induction algorithm as a black box. It conducts a search in the space of available variables using the induction algorithms as part of the evaluation function to find a good subset. For the search, a state space, an initial space, a termination condition, and a search engine are required (Kohavi & John, 1997). Figure 5 shows a schematic overview of the wrapper approach according to Kohavi and John (1997).

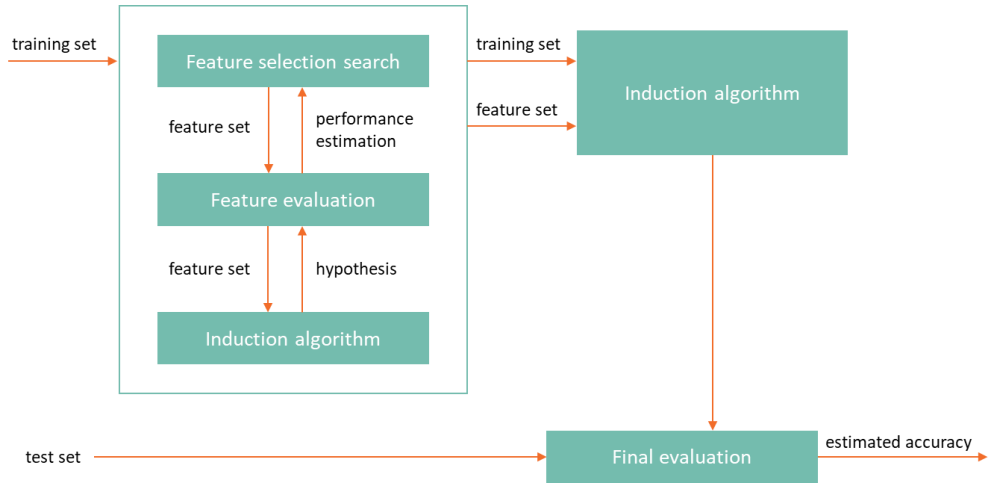

Figure 5: *The wrapper approach for feature subset selection*

If the number of available features is small enough, an exhaustive search can be done with good performance. However, the problem of finding the optimal subset of variables is still NP-hard (Amaldi &

Kann, 1998), quickly increasing in computational cost when the variable space increases. Therefore, research is conducted to find efficient search strategies without sacrificing model performance also in the context of rainfall-runoff modeling (Taormina & Chau, 2015; Maier & Dandy, 2000; Maier et al., 2010).

Embedded methods select variables during the training step and are usually specific to the machine learning algorithms (Guyon & Elisseeff, 2003). Figure 6 shows a schematic overview of the embedded approach.

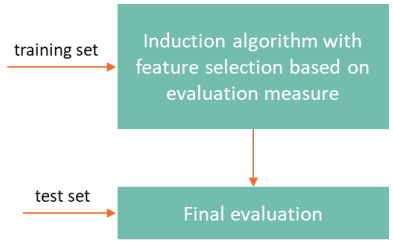

Figure 6: *The embedded approach for feature subset selection*

In the case of C4.5, for example, an information gain measure is used during the training step to select the relevant features (Martinez & Fuentes, 2005). The Random Forest framework mostly uses the mean of the error of the randomly permuted trees as the score of importance. In the case of classification, this is based on the misclassification rate (Genuer, Poggi & Tuleau-Malot, 2010). Embedded methods are understood to be more computationally efficient than wrappers as they purely focus on the development of a single final model. However, the search they perform to find the optimal subset has a higher complexity, which might outweigh the increased computational efficiency (Taormina & Chau, 2015).

Rather than choosing just one of the variable selection methods above, another approach is to combine them. One approach is to use a wrapper or embedded method with a linear inducer as a filter and then to train a more complex non-linear inducer on the resulting subset (Guyon & Elisseeff, 2003). One example of this approach is described in the paper of Bi et al. (2003).

Identifying the subset of variables that allows to fully model a hydrological process is a difficult task. Maier and Dandy (2000) as well as Maier et al. (2010) identify this as one of the most important steps in the development of an Artificial Neural Network for hydrological predictions including rainfall-runoff modeling. However, this applies as well to classification algorithms. The research conducted by Erechtchoukova et al. (2016) shows how the selection of sensors contributing to the data set used for the flood prediction task can have a high impact on the resulting classifier performance.

2.4 IMBALANCED DATA SETS

In an imbalanced data set classes are not represented equally. The data set contains only a small percentage of tuples representing the event of interest, compared to the number of tuples representing the remaining classes. Imbalanced data often cause bad performance as the classifier is trained more towards the majority class favoring this class over the minority class. In the case of decision trees, for example, each minority class tuple in the data set will eventually be represented by one branch of the tree and as a result, the area of the tree representing this class will be arbitrarily small. Tuples in an imbalanced data set can be classified into roughly four categories – class label noise, borderline tuples, redundant tuples, and safe examples. Class label noise tuples are of the majority class, close to one or many tuples of the minority class in many cases even the nearest neighbor. They can cause tuples of the minority class to be misclassified. Borderline tuples are tuples close to the boundary between the different classes. They are unreliable as even small noise can send the tuple to either side of the border. Redundant tuples are tuples mostly of the majority class that do not contribute to the correct fitting of the classifier as their part can be taken over by other tuples close to them. Tuples that fall into the safe examples category do not belong in any of the other classes and are worth keeping for the classification task (Kubat & Matwin, 1997). Figure 7 shows how the tuples of the four categories are located within the feature space according to Kubat and Matwin (1997).

Figure 7: *Tuples in an imbalanced data set*

Working with imbalanced data sets rises two questions. How to achieve good prediction results and how to measure the performance of the prediction. The following sections aim to answer these two questions.

2.4.1 PREPROCESSING STRATEGIES FOR IMBALANCED DATA SETS

Working with imbalanced data sets has been researched for all kinds of application areas. As a result, there are multiple approaches to dealing with imbalanced data. These approaches can be divided into the categories over-sampling and under-sampling. Over-sampling increases the number of tuples of the minority class in the training set and under-sampling decreases the number of tuples of the majority class in the training set (Batista et al., 2004).

One over-sampling technique is random over-sampling. This is a non-heuristic method that randomly replicates tuples from the minority class with the goal to balance the class distribution. The advantage of random over-sampling is that it is relatively easy to implement. However, overfitting is likely to occur as it uses the exact copy of the tuples (Batista et al., 2004). SMOTE, the Synthetic Minority Oversampling Technique, on the other hand, creates new tuples of the minority class by interpolating between several minority class tuples that are in close proximity. This approach addresses the issue of overfitting as new tuples are generated instead of replicating existing ones (Chawla et al., 2002). Another positive effect is that the decision boundaries for the minority class are spread further into the majority class space (Batista et al., 2004).

Random under-sampling is a non-heuristic method for under-sampling that randomly removes tuples from the majority class with the goal to balance the class distribution. The advantage of random under-sampling is, like random over-sampling, the easy implementation. Yet, randomly discarding tuples could cause tuples to be removed that are essential for the training process (Batista et al., 2004). The goal of the Condensed Nearest Neighbor Rule approach is to reduce the training set to a subset so that the classifier built from this subset can achieve results that are very close to the results from the full training set. This is achieved by eliminating the redundant tuples from the majority class that are distant from the decision border and being less relevant for the learning of the classifier (Hart, 1968). A shortcoming of this approach is that it contains random sampling to find a sufficient subset. This random sampling results in the retention of unnecessary samples but also the occasional retention of internal samples rather than samples that are close to the borderline (Tomek, 1976). Ivan Tomek (1976) introduced Tomek links as a modification of the Condensed Nearest Neighbor Rule to address its shortcomings. A Tomek link exists if there are two tuples from different classes that do not have any other tuple from their own class that is closer to them. In this case, these tuples are either noise or borderline tuples. The tuple of the majority class for each Tomek link is then removed. As a result, samples are no longer chosen randomly as with the Condensed Nearest Neighbor Rule but based on whether they are part of a Tomek link or not, making sure that mostly borderline samples are chosen.

There are also approaches that combine multiple of the above methods such as the one-sided selection approach, which first uses Tomek links to remove noise and borderline tuples followed by the Condensed Nearest Neighbor Rule, which removes all tuples that are distant from the border, and therefore not as relevant (Batista et al., 2004).

Saffarpour et al. (2015) investigated the issue of imbalanced data in the context of flood prediction. The data set used contained hourly data collected at the Spring Creek watershed from one rainfall sensor and one water level sensor for the years 2008, 2011, 2012, and 2013. Predictions were generated by an ensemble of the five classifiers C4.5, CART, REPTree, NNge, and JRip. The authors implemented three data preprocessing approaches to address the issue of imbalanced data. For the first approach, Saffarpour et al. (2015) maintained all low flow tuples of a year and oversampled the existing high flow tuples for the same year. For the second approach, they used a wet year with the highest number of high flow tuples as the training set for the other years. For the third approach, they again maintained the low flow tuples of each year and then combined the high flow tuples of all available years and added them to the training set for that year. They found the third approach to return the best prediction results except for the year 2012, which was the only dry year available. Their rational is that this is due to the difference in the flood patterns of wet and dry years so that the floods in the dry year cannot be well detected by classifiers that were mostly trained on high flow events from wet years. The study demonstrated that the traditional over-sampling approach did not produce sufficient results and the hydrological differences between floods in wet and dry years need to be considered when choosing imbalanced data strategies.

2.4.2 EVALUATING MODEL PERFORMANCE ON IMBALANCED DATA SETS

The most common way for evaluating classifier performance is to use the confusion matrix as shown in Figure 8 (Han et al., 2011).

		Predicted Class		
		Flood	No Flood	**Total**
Actual Class	Flood	TP	FN	**P**
	No Flood	FP	TN	**N**
	Total	**P'**	**N'**	**P+N**

Figure 8: *Confusion matrix*

Within the confusion matrix, P (Positives) represent the tuples of the class for which the model performance is being analyzed while N (Negatives) represent the tuples that do not belong to this class. Therefore, TP (True Positives) represent the number of positive tuples that have been correctly classified as such. FP (False Positives) are the negative tuples that have been incorrectly classified as positive. TN (True Negatives) are then all negative tuples that have been correctly classified as negative, while FN (False Negatives) are positive tuples that have been incorrectly classified as negative. P' are all tuples that have been classified as positive and N' are all tuples that have been classified as negative independently from their actual class.

With the help of the confusion matrix, it is possible to extract multiple simple numerical performance measures. However, when evaluating classification performance on imbalanced data sets, it is essential to choose the correct performance measurement as the use of an inappropriate measurement might lead to misleading conclusions. The following section covers the most common performance measures and evaluates them for their practicality on imbalanced data sets.

Accuracy/recognition rate is the proportion of tuples that are correctly classified.

$$Accuracy = \frac{TP + TN}{P + N} \tag{6}$$

Error rate/misclassification rate is the proportion of tuples that are incorrectly classified. Also known as re-substitution error when used on the training set instead of the test set (Sebastiani, 2002).

$$Error\ rate = \frac{FP + FN}{P + N} \tag{7}$$

Accuracy and error rate are highly impacted by class imbalance as they are strongly favoring the majority class. In a data set where the majority class takes up 99% of all tuples an accuracy of 99% could simply mean that all tuples are classified with the majority class and as a result, all tuples of the minority class are mislabeled. The same principle applied to the error rate (Batista et al., 2004). As a result, accuracy is not sufficient to measure the performance of a model with an imbalanced distribution of the class. In this case additional measures such as sensitivity and specificity can be used as an alternative (Rokach & Maimon, 2014).

Recall/completeness/sensitivity/true positive rate measures how well the model can recognize positive sample. It is defined as the proportion of positive tuples that are labeled as positive. A perfect Recall score of 1 means that every positive tuple was labeled correctly as positive (Rokach & Maimon, 2014; Sebastiani, 2002).

$$Recall = \frac{TP}{P} \tag{8}$$

Specificity/ true negative recognition rate measures how well the model can recognize negative samples. It is defined as the proportion of negative tuples that are correctly classified (Rokach & Maimon, 2014; Sebastiani, 2002).

$$Specificity = \frac{TN}{N} \tag{9}$$

Precision/exactness is the proportion of tuples labeled as positive that are actually positive. A Perfect precision score of 1 means that every tuple that has been labeled as positive was indeed a positive tuple (Rokach & Maimon, 2014; Sebastiani, 2002).

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

It is possible to combine multiple simple numerical performance measures into more complex numerical performance measures.

F-Measure: In many cases model performance shows a trade-off between Precision and Recall. Increasing one of the measures often leads to a decrease in the other one. The F-Measure combines both measurements into a single value using the harmonic mean (Rokach & Maimon, 2014).

$$F\ Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{11}$$

A high F-Measure ensures that both Recall and Precision are high. However, a low F-Measure does not necessarily show whether Precision or Recall is the limiting factor.

Evaluating the expected performance of a classifier in the context of limited resources requires cost-benefit considerations. In some cases, the limitation could be known to exist, but the size might not be known in advance. For these cases a performance measurement is needed that allows to evaluate the model performance while considering the cost-benefit trade-off (Rokach & Maimon, 2014).

Relative Operating Characteristics Curve (ROC Curve): Classifiers can be biased towards the majority class. Correctly classifying the tuples of the minority class can often be improved at the cost of correctly classifying the tuples of the majority class. The ROC curve displays this trade-off. It measures the false positive rate on the horizontal axis and the true positive rate on the vertical axis. The ROC curve shows how the true positive rate increases along the false positive rate (Swets, 1988; Kubat & Matwin, 1997).

Figure 9 shows an example of the ROC curve for a classifier compared to the random guessing referencing line.

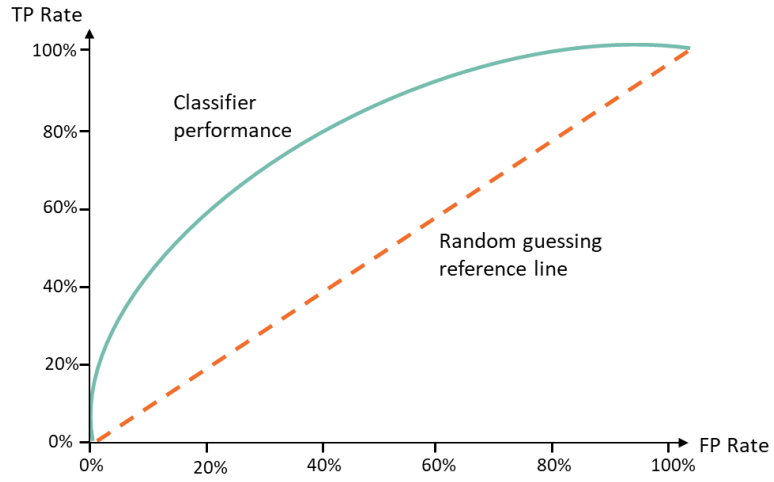

Figure 9: *ROC curve*

Area Under Curve (AUC): Using continuous measures like ROC curves can only answer the question of which model is best if the curve for one model is above the curves for all other model over the entire chart space. In practise, one model might outperform the others in certain areas but might show lower performance in others. If a complete order of model performance needs to be obtained the area under the ROC curve can be used. The bigger the AUC the better is the model performance (Rokach & Maimon, 2014).

3 CASE STUDY

The following chapter explains the characteristics of the watershed used for this study as well as the results of the preliminary data analysis.

3.1 WATERSHED CHARACTERISTICS

The experiments were performed based on data sets collected from the Spring Creek watershed in the Greater Toronto Area, Ontario, Canada. The Spring Creek is one of two tributaries of the Etobicoke Creek. It joins the Etobicoke Creek approximately 13.5 km upstream of Lake Ontario within the Toronto International Airport lands. The Spring Creek is over 23 km long and stretches through a watershed area of around 50 km^2. The main branch of the Spring Creek is relatively steep (TRCA 2006). The Spring Creek has 70 sub-catchments with an average catchment size of 71 hectare (MMM Group, 2013). Most of its watershed is completely urbanized which creates a highly impervious surface. As a result, heavy rainfalls generate quick runoff responses in very short periods, often even less than an hour. This leads to water inundation and flash flood events that occur from April and lasting until December (TRCA 2006).

The data used for the experiments consisted of two data sets, one for 2013 and one for 2014 (both April to December), and was collected by the Toronto and Region Conservation Authority (TRCA). The TRCA is one of 36 Conservation Authorities in Ontario that works with municipalities and other partners to look after the watersheds of the Toronto region and its Lake Ontario waterfront (TRCA, 2018 b). Each data set includes measurements from two observation sites at the Spring Creek watershed that measure the water level (Spring Creek North and Spring Creek South). These sites will be identified as sensor WN and WS. In addition to that, the data sets contained data from two observation sites that measure precipitation (Heart Lake and Mississauga Works Yard) which will be identified as RN and RS. Water level measurements were collected on a 15-minute interval while precipitation was collected on a 5-minute interval. Only in 2013 and 2014 data are available from all 4 sensors, limiting the experiments to those two years.

The TRCA defined observation site Spring Creek South to be the cross-section of interest for the flood prediction.

Figure 10 shows the Spring Creek Watershed and the observation sites managed by the TRCA. Furthermore, it indicates the observation sites used for this study (TRCA, 2018 a).

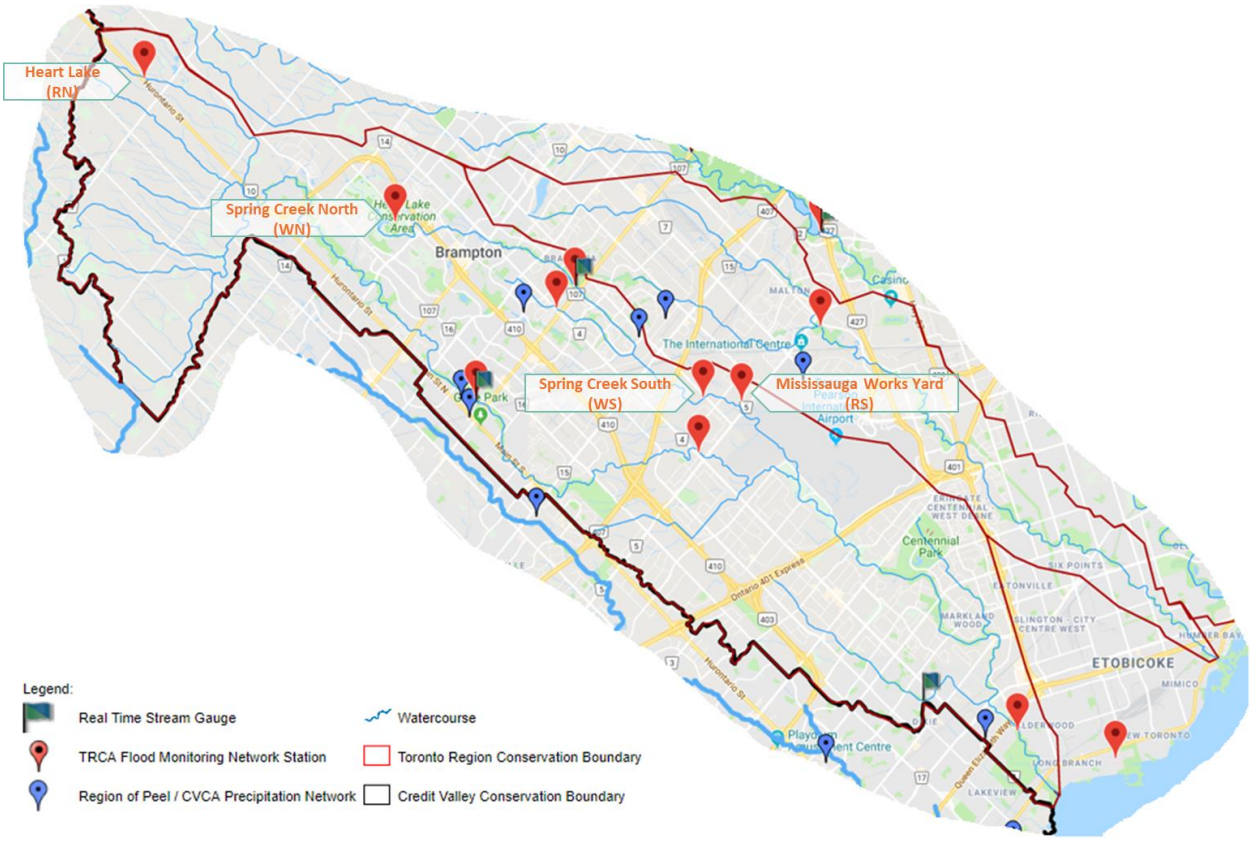

Figure 10: *Spring Creek watershed*

3.2 PRELIMINARY ANALYSIS

Between April 9th and December 3rd 2013, the Spring Creek received a total rainfall of 806.40 mm at sensor RN and 715.40 mm at sensor RS. Monthly rainfall in 2013 shows a median (only including full months May to November) of 99 mm for sensor RN and 80 mm for sensor RS. Monthly rainfall ranges from 33.40 mm in November to 184.20 mm in July at sensor RN and from 34.00 mm to 189.00 mm at sensor RS for the respective months. Figure 11 shows the rainfall by month for both sensors and their median.

Between April 24th and December 1st 2014, Spring Creek experienced a total rainfall of 544.60 mm at sensor RN and 501.80 mm at sensor RS. Monthly rainfall in 2014 shows a median (only including full months May to November) of 62.40 mm for sensor RN and 57.80 mm for sensor RS. Monthly rainfall ranges from 30.80 mm in August to 124.60 mm in September at sensor RN and from 33.20 mm to 118.00 mm at sensor RS for the respective months. Figure 12 shows the rainfall by month for both sensors and their median.

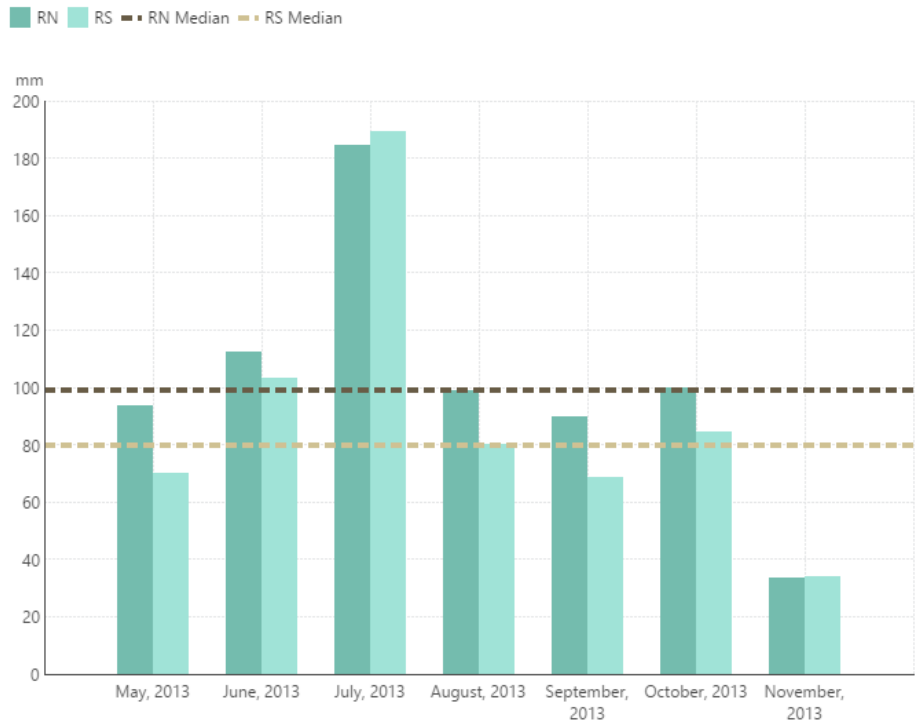

Figure 11: *Rainfall analysis 2013*

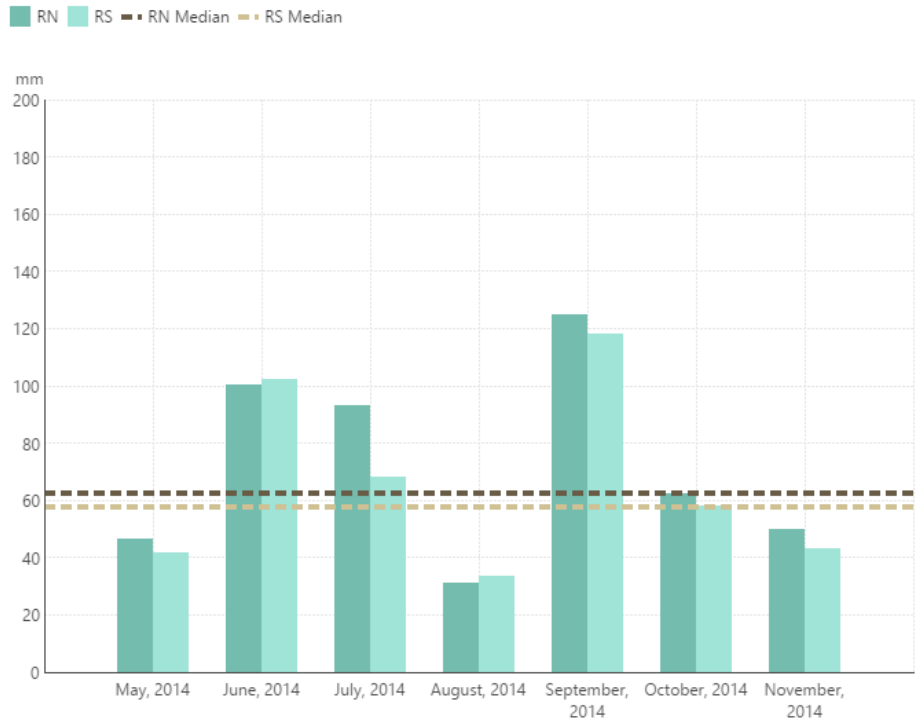

Figure 12: *Rainfall analysis 2014*

Monthly rainfall does not follow a periodical recurrence and usually spikes between the months of June and September.

Daily rainfall intensity in 2013 ranges from 0 mm to 20.40 mm for sensor RN and from 0 mm to 15.60 mm for sensor RS. In 2014 rainfall intensity ranges from 0 mm to 13.20 mm for sensor RN and from 0 mm to 13.60 mm for sensor RS. Based on this analysis, 2013 can be considered a wet year and 2014 a dry year.

Water level distribution in 2013 and 2014 is positively skewed. The median water level in 2013 was 209.51 cm for sensor WN and 172.14 cm for sensor WS. In 2014, the median water level was 209.48 cm for WN and 172.15 cm for WS. In 2013, the water level ranged from 172.16 cm to 212.18 cm for sensor WN and 172.11 cm to 174.60 cm for sensor WS. In 2014 the water level ranged from 209.43 cm to 210.07 cm for sensor WN and 172.10 cm to 173.75 cm for sensor WS. Instantaneous water discharge at sensor WS ranges from 0.1 m^3/s to 29 m^3/s in 2013 and 0.05 m^3/s to 9 m^3/s in 2014. Based on this analysis, 2013 can be considered a wet year and 2014 a dry year.

When analysing the dependencies between rainfall and water level it becomes apparent that extensive rainfall takes about 2 to 3 hours until the water level rises, and the flooding occurs. This analysis was conducted by visualizing the water level and rainfall values leading up to the flood events. Figures 13 and 14 present two examples of this visual analysis.

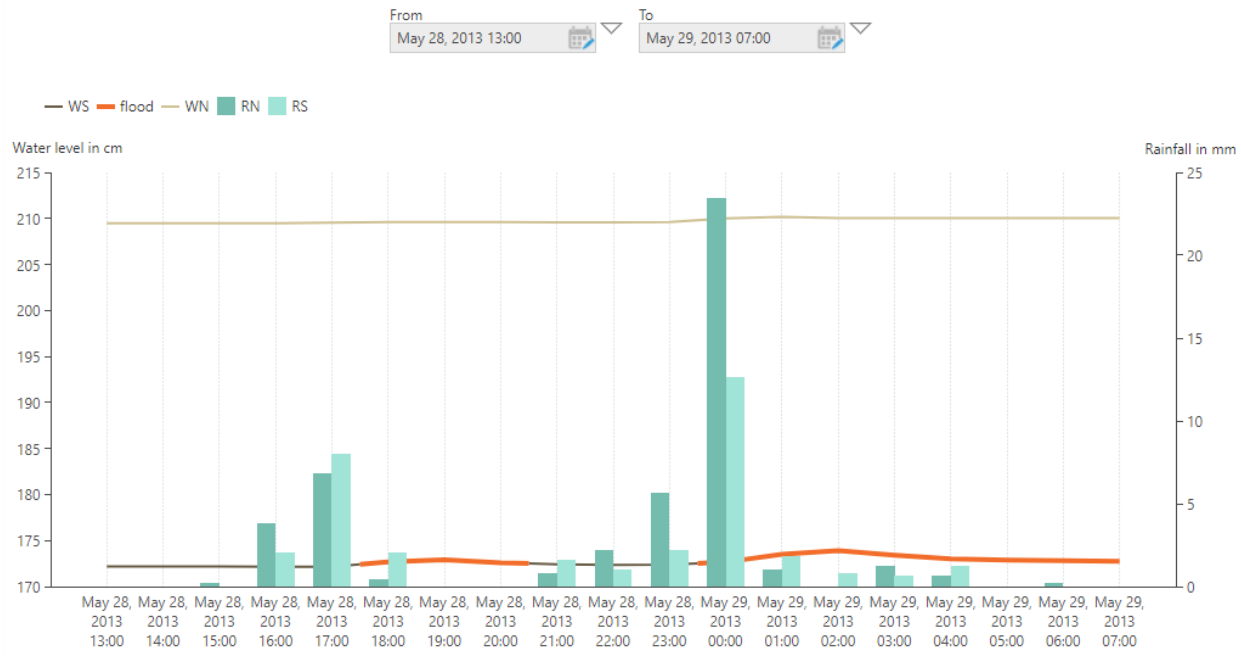

Figure 13: *Flood example 2013*

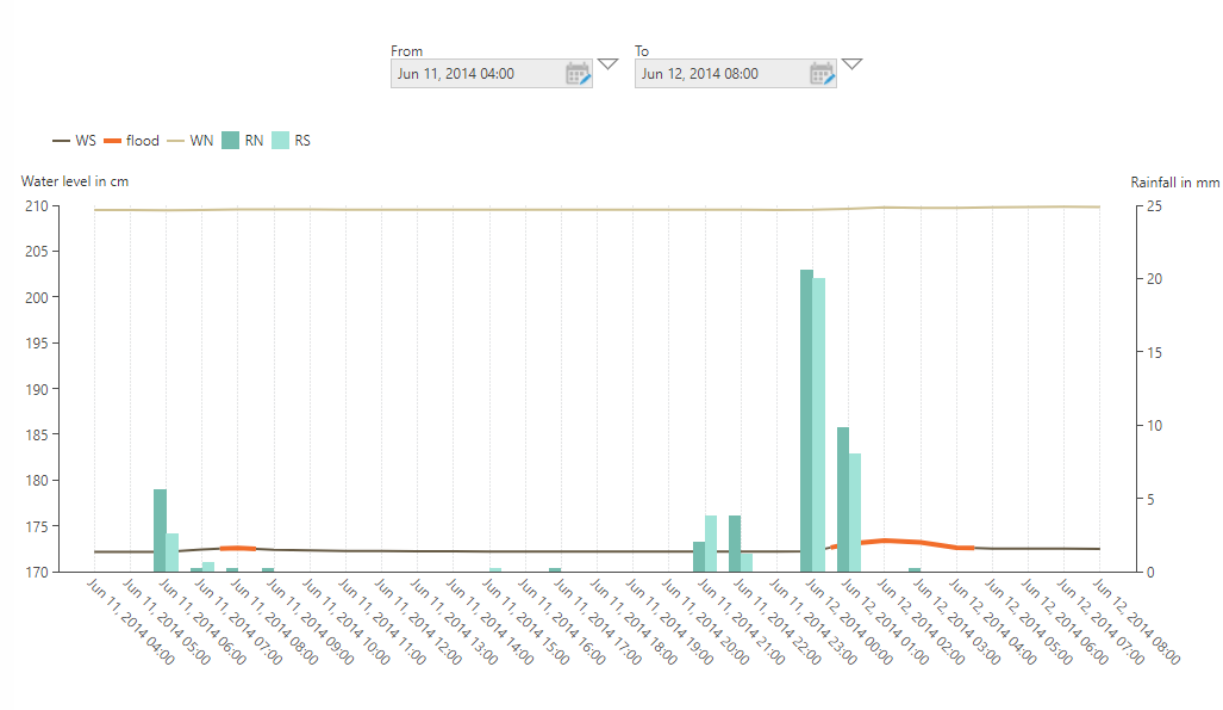

Figure 14: *Flood example 2014*

4 DATA PREPROCESSING

Data Preprocessing was performed in seven steps as outlined by Figure 15 using Excel as well as R (The R Foundation, 2018).

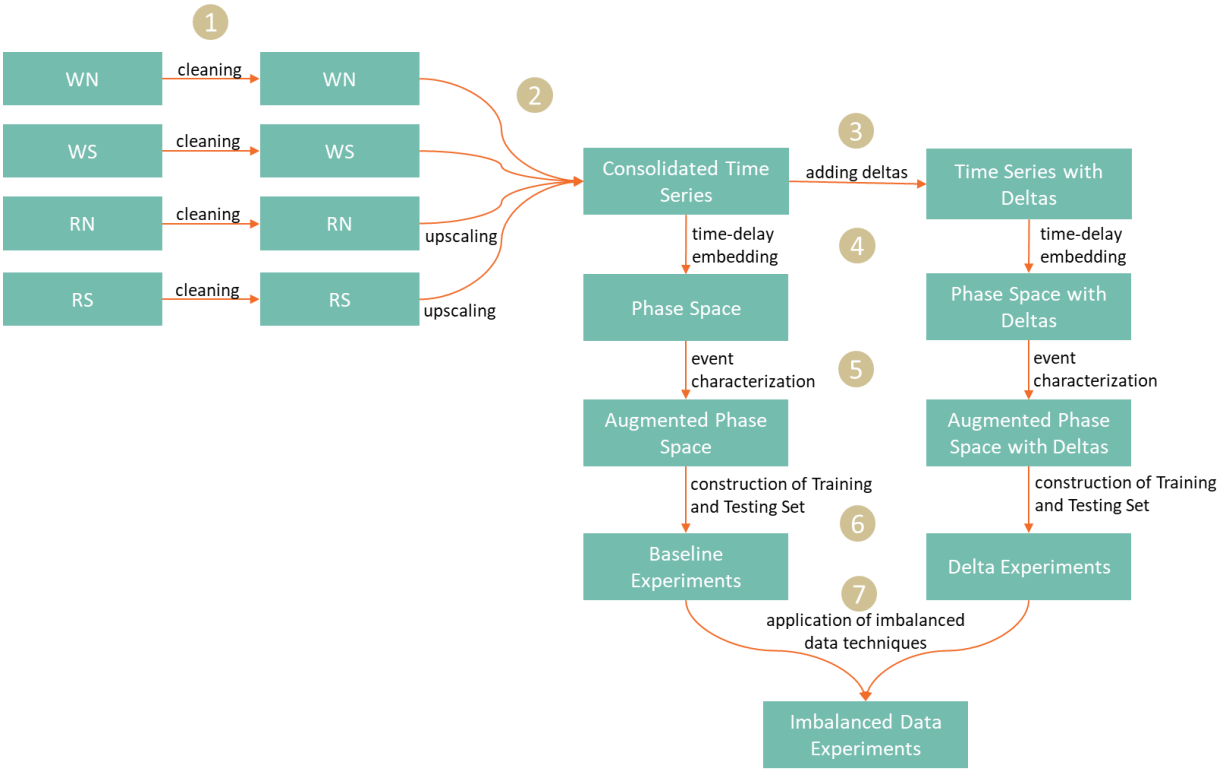

Figure 15: *Preprocessing steps*

Step 1: Cleaning

First, the data sets were cleaned by removing some of the rainfall records had a value of -999.99 indicating a missing value. During this process, less than 1% of all records had to be removed.

Step 2: Consolidation and upscaling

Data were provided in the time series form. As the data for the water level were measured on a 15-minute interval while the data for precipitation were measured on a 5-minute interval, the water level interval determined the lowest granularity for the prediction of the events. Classification algorithms can be applied on heterogenous data. Thus, it is possible to build a data set using both water level data and precipitation data at the granularity at which they were measured without synchronizing them. Erechtchoukova and

Khaiter (2017) studied the effect of data granularity on model performance. The results have shown that, model performance for lead times of 45 minutes and more increases when precipitation data are aggregated to 15-minute intervals compared to when unaggregated precipitation data are used. Further upscaling of the observation data would cause the prediction of the beginning of an event to become less accurate. Therefore, upscaling increases the total uncertainty of the hydrological prediction and must be considered carefully. Previous studies on the Spring Creek river have shown that flashy responses to water discharged into the watershed are typical. Often the water level rises quickly within less than half an hour. Hence, data sets of a 15-minute granularity were deemed the most suitable for the investigated watershed. As a result, only the precipitation data were upscaled to a 15-minute interval according to the hydrological characteristics of the tipping bucket rain gauges used for data collection. Then data from all sensors were synchronized and consolidated (Erechtchoukova & Khaiter, 2017).

Step 3: Adding the delta

So far, only the total precipitation and average water level had been used as part of the time series following the methodology of Erechtchoukova et al. (2016). For this study, delta derivatives, or simply deltas were added for attributes from all sensors including precipitation and water level. Deltas were computed as the difference between the water level or precipitation value of one timestamp to the previous one. The reasoning behind this is that the delta allows the model to consider the magnitude of the increase or decrease of the water level or rainfall and not just the actual value at time *t*. A delta attribute was added for each timestamp at each sensor. The delta of hydrological data collected from an observation site can then be represented as the following time series:

$$\Delta_{k,t} = \{(y_{k,t} - y_{k,t-1}), k = 1, \dots, M, t = 1, \dots, N\} \quad (12)$$

with $y_{k,t}$ being the value measured at time *t* by observation site *k* and *M* being the total number of observations that are included in the time series.

Step 4: Creating the phase space

With this extended time series, a phase space was reconstructed using the time-delay embedding technique from multiple observation sites as explained in chapter 2.1.4.

An element of this phase space can be described as below:

$$z_t = \begin{matrix}(y_{1,t-(R-1)\tau}, \Delta_{1,t-(R-1)\tau}, \dots, y_{1,t}, \Delta_{1,t}, y_{2,t-(R-1)\tau}, \Delta_{2,t-(R-1)\tau}, \dots, y_{2,t}, \Delta_{2,t}, \dots, \\ y_{M,t-(R-1)\tau}, \Delta_{M,t-(R-1)\tau}, \dots, y_{M,t}, \Delta_{M,t})\end{matrix} \quad (13)$$

with $y_{k,t}$ being the measured value by sensor k at time t, R being the number of successive observations in the time-delay window, and τ the time interval between measurements. $k = 1, \dots, M$ indicates the observation site with the M^{th} observation site being the cross-section of interest. Creating the phase space over a large window size (number of consecutive observations at different points in time) can add redundancy and noise. This can degrade the performance of the model and increase the computation time. Selecting a window size that is too small could leave out important information for the prediction, again decreasing the performance (Galka, 2000). Based on a previous analysis of the flood events, showing that extensive rainfall takes around 2 to 3 hours until the water level rises, the window size was set to 3 hours before the prediction time. This also ensures that for each lead-time the number of successive observations used in the phase space is the same keeping this independent variable static throughout all experiments.

Step 5: Event characterization function

The following event characterization function was used to assign the class label.

$$f_{WS}(x_{WS,0}) = \begin{cases} 'high', & y_{WS,0} \geq 172.75cm \\ 'low', & y_{WS,0} < 172.75cm \end{cases} \quad (14)$$

This function was applied to the water level value of sensor WS at time t. The threshold was 172.75 cm and provided by the TRCA, which monitors the hydrological conditions on Spring Creek watershed. After the event characterization function was applied, it became apparent that 2013 had 388 flood events and 22,365 non-flood events while 2014 had 171 flood events and 20,992 non-flood events. Therefore, flood events made up 1.7 % of all events in 2013 and 0.8% in 2014. Figure 16 shows the distribution of flood events across the months comparing 2013 and 2014, not showing any sessional recurrences.

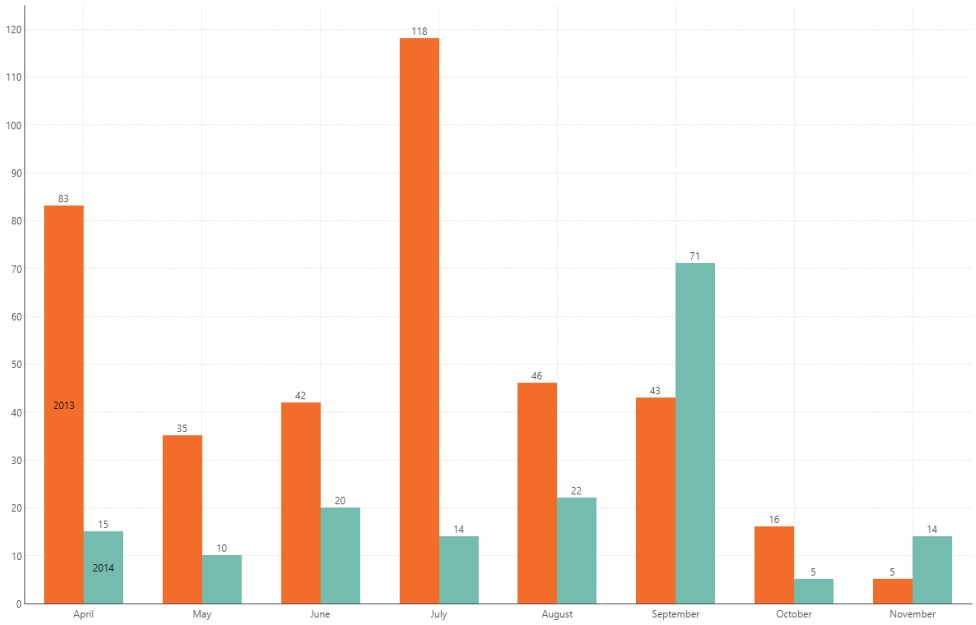

Figure 16: *Monthly distribution of flood events*

Step 6: Construction of training and testing set

To run the experiments, a training set is needed to train the model and a testing set to check how well the model is performing. Previous experiments conducted by Saffarpour et al. (2015) showed that training the classifiers on a wet year does not return good results when these classifiers are then applied to a dry year. These findings were confirmed during preliminary experiments, which used the 2013 data set as the training set and the 2014 data set as the testing set. In addition to that fewer high-flow tuples were present in the 2014 data set. Thus, the decision was made to enhance model performance by combining data from both years into the training and testing set. This allowed accounting for the different hydrological characteristics of wet years and dry years. It also allowed the predictors to be trained on a higher number of tuples than before. To prevent underfitting, most of the tuples should be used to train the model allowing it to see enough data so that it can extract all important underlying patterns and generalize to unseen data. For the experiments, the training sets contained 70% of the tuples and the testing sets contained 30% of the tuples. At the same time the training set contained 70% of all flood events while the testing set contained 30% of all flood events. As a result, the ratio of flood event tuples to no flood event tuples stayed constant. Using a combination of 2013 and 2014 tuples also allowed for the creation of multiple variations of training and testing sets. To get a better estimate on the generalization error of the inducers, five different testing and training sets were created each using the holdout method as implemented in the createDataPartition function as part of the caret package for R (Kuhn, 2017). For each training and testing set, the function was called

with a different seed ensuring that all data sets are different. It is understood that five different data sets are not enough to provide a valid statistical representation of the generalization error.

Step 7: Imbalanced data techniques

At this step, two different techniques to handle imbalanced data sets were applied. SMOTE was used to address the issue with the traditional over-sampling approach, where existing tuples are added multiple times to the training set, as identified by Saffarpour et al. (2015). As SMOTE creates new tuples from both 2013 and 2014 data sets, the new high flow tuples contain the hydrological pattern of both wet and dry years. This aims to improve the prediction of flood events in both wet and dry years. Previous research conducted on the Spring Creek watershed as well as the baseline experiments described in later chapters showed that misclassification occurs mostly as tuples approach the threshold for the event characterization function and are located at the border between high and low flow events. Tomek links were chosen to remove the no flood tuples too close to flood tuples so that the classifiers favor the flood class at this border. This is aligned with the nature of flood prediction where correctly predicting a flood is assigned higher importance than issuing false positives. In addition to applying each technique individually, experiments were also conducted on data sets using a combination of both techniques to determine if these two methods complement each other or if they should be used separately.

For the SMOTE technique, one new minority class tuple was created for each already existing one in the training set. This resulted in 392 new flood event instances and a total of 784 instances in the training set, about 2.6% of all training set instances. The ratio of one actual ‘flood’ tuple to one synthetic ‘flood’ tuple was selected to make sure that there would not be more synthetic instances than actual recorded ‘flood’ instances but still enough to impact the overall class distribution. Further research could investigate how this ratio affects the overall performance and if there is an ideal ratio. The new tuples were created using the SMOTE approach as introduced by Chawla et al. (2002) and then added to the training set. For this, the ubSMOTE function available from the unbalanced package in R was used. This function takes a data set and generates the synthetic flood event tuples. (Pozzolo et al., 2015).

For the Tomek links technique, first all Tomek links, as defined by Ivan Tomek (1976), were identified in the training set. Then the tuples of the majority class that are part of a Tomek link were removed. Tomek links were identified and removed using the ubTomek function available from the unbalanced package in R. This function takes a data set, finds the Tomek links within this data set and removes the no flood tuple of each found Tomek link (Pozzolo et al., 2015). When applied to the data sets containing all deltas the

algorithm found and removed between 157 and 176 instances of the majority class. When applied to the baseline data sets, the algorithm found and removed between 97 and 127 instances of the majority class.

Finally, both techniques were combined by applying the Tomek links techniques followed by the SMOTE technique.

5 EXPERIMENTS

Experiments were conducted using the Weka Experimenter (Hall et al., 2009) with seven inducers, five decision trees (J48, NBTree, Random Forest, SimpleCart, REPTree), and two rule-based classifiers (JRip, Ridor), as well as one classifier ensemble consisting of J48, SimpleCart, ReportTree, JRip and Ridor, combined using majority vote. Experiments conducted on these inducers showed how the applied techniques and modifications to the data sets affect the results of different inducers while the experiments conducted with the ensemble allows for more generic and robust insights. Each experiment was conducted with a lead-time of 15, 30, 45, and 60 minutes. To accommodate for the imbalance of the data set as well as the fact that misclassifying a no-flood event is not as dangerous as misclassifying a flood event, the Precision and Recall for flood events were defined as the main indicators of model performance. Both Precision and Recall allow evaluating prediction of high-flow events as opposed to other measures reflecting estimates of the generalization error averaged over two classes. Recall represents how well the model can identify the flood events which is the most important ability of a flood warning system. However, the false alarm rate, represented by Precision, should not be neglected to ensure public support of the system. That is why the Precision was analyzed alongside Recall for the experiments. Even though both measures are often combined into the F-Measure, this does limit the ability to see which of the two measures caused an increase or decrease. Therefore, the decision was made to mostly compare both measures separately but to consult the F-Measure in cases where a more consolidated view on the model performance is needed.

One round of initial baseline experiments was conducted without additional delta attributes or applied imbalanced data techniques. During the second round of experiments, delta attributes were added to the training and testing set over the full window size once for all attributes, once just for the rainfall attributes and once just for the water level attributes. These experiments assessed how the different types of deltas affect the prediction results. The experiments demonstrated that while delta attributes do have the ability to improve model performance, especially with higher lead-times deltas can introduce noise and lower model performance. Therefore, the third round of experiments was performed adding delta attributes only over the first or first two hours of the time series to select the variables, which carry more information than the others. For the imbalanced data experiments, the fourth round of experiments was performed on the training sets after removing the Tomek links to see how this affects model performance. Tomek links were removed once from the baseline training sets and once from the training sets containing all deltas over the full window size. This aims to show if this preprocessing technique affects prediction results differently when delta attributes are added. As removing Tomek links is a form of under-sampling for the fifth round of experiments, the SMOTE algorithm was used to add additional synthetic tuples of the minority class to the

training set. This round of experiments was implemented to see how this type of over-sampling affects model performance. SMOTE tuples were added once to the baseline training sets and once to the training sets containing all deltas over the full window size to see if the effect is different if delta attributes are introduced. To see how a combination of both techniques affect the results, a sixth round of experiments investigated the impact of both techniques combined on model performance.

6 RESULTS

The chapter presents and analyzes the results of all conducted experiments. The results of the baseline experiments are described. These results are compared to the delta and imbalanced data experiments.

6.1 BASELINE EXPERIMENTS

Figure 17 shows the Precision and Recall results for the ensemble over the different lead-times for data set 3. The ensemble baseline results for all data sets can be found in the Appendix A.

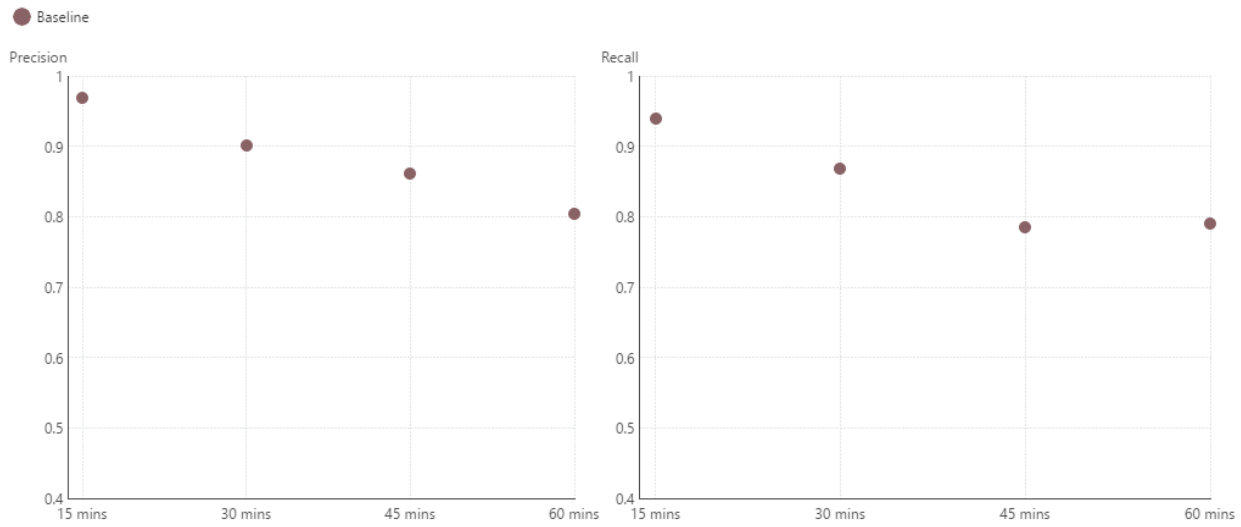

Figure 17: *Baseline precision and recall for ensemble (data set 3)*

The baseline results for the classifier ensemble show that both Precision and Recall decrease with increasing lead-time. Overall, across all data sets the decrease is in an almost linear fashion with some exceptions. It is also visible that overall Recall decreases at a higher rate than Precision. For data set 1, Precision decreases from 0.96 for 15 minutes lead-time to 0.87 for 60 minutes lead-time while Recall starts similar to Precision with 0.95 for 15 minutes lead-time but then drops to 0.75 for a lead-time of 60 minutes. The results of the other data sets support this statement.

The results of the single classifiers were considered as well. Figure 18 shows these results for data set 3. The results for all data sets can be found in Appendix B.

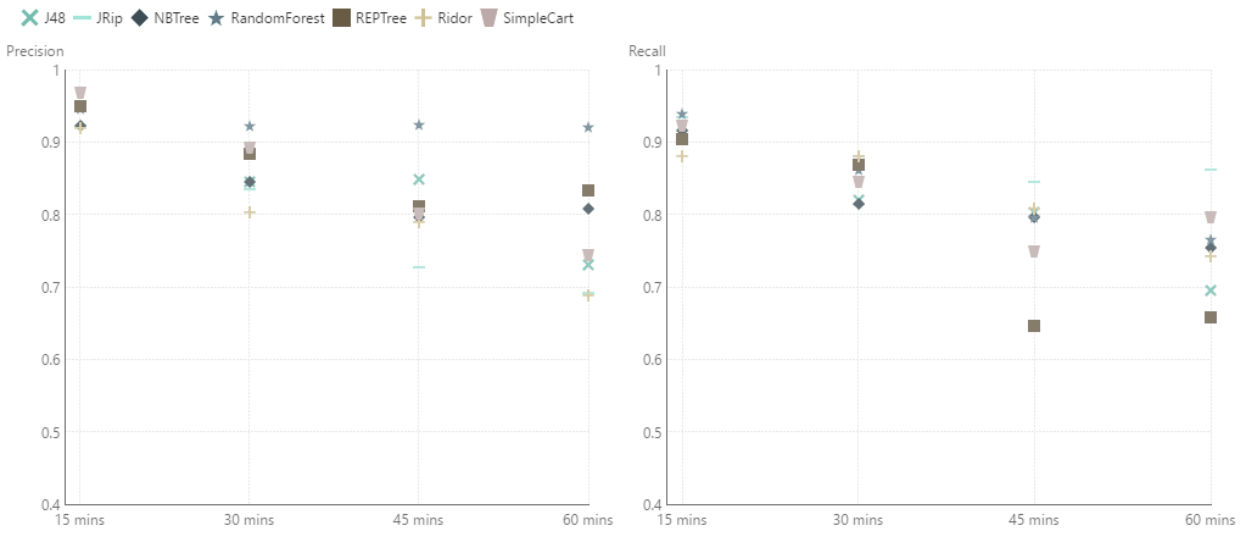

Figure 18: *Baseline precision and recall for single classifier (data set 3)*

When comparing the different classifiers, Random Forest achieves the highest Precision for all lead-times and shows consistent Precision with increasing lead-times. Recall results on the other hand decrease significantly for longer lead-times showing that Random Forest favors the majority class with increasing uncertainty. REPTree shows a similar trade-off between Precision and Recall especially for longer lead-times and data sets 3 and 4. For the other data sets, REPTree shows a more balanced behavior between Precision and Recall. JRip shows consistently across all data sets and lead-times one of the highest Recall results as the only classifier that achieves a Recall of higher than 0.83 for 60 minutes lead-time. However, it achieves these high Recall results by favoring the minority class, which is reflected in its Precision results causing a high number of false alarms, especially for longer lead-times. J48, NBTree, and SimpleCart show results that are more balanced between Precision and Recall and both performance measurements decrease evenly for increasing lead-times. Ridor shows inconsistent results to whether it favors the majority or minority class but a drop in Precision is closely related to an increase in Recall and vice versa. For data set 4, for example, Ridor achieves a Recall of 0.93 for 30 minutes lead-time and then drops to 0.59 for 45 minutes lead-time. After that, Ridor, achieves a Recall of 0.83 for 60 minutes lead-time. At the same time, Ridor shows only a Precision of 0.8 for 30 minutes lead-time, increases to a result of 0.86 for 45 minutes lead-time, and then drops to 0.67 for 60 minutes lead-time.

In order to see what variables the classifiers selected and their importance, the decision trees and rules were inspected. For the baseline experiments, the classifiers assign the highest relevance to the water level sensor WS at the earliest available timestamp. This is expected as the value of WS at time t determines the class label. The water level sensor WN is not included as often as WS especially for the shorter lead-times but appears more often for longer lead-times. Based on the inspected decision trees and rules, rainfall variables

seem to be as relevant to the prediction as the water level variables. However, while water level variables seem more relevant closer to the actual event, the rainfall variables are selected from timestamps further away from the time of the event. It is also apparent that the number of rules and their complexity as well as the number of leaf nodes and the tree size increase with increasing lead-times. Figure 19 shows the rule-set defined by the JRip classifier for data set 3 and a lead-time of 15 minutes while Figure 20 shows the rule-set for a lead-time of 60 minutes.

```
(WS(t-15) >= 172.721) and (WS(t-15) >= 172.804) => Class=High (461.0/13.0)

(WS(t-15) >= 172.7102) and (RN(t-75) >= 0.4) and (WS(t-15) >= 172.76822) => Class=High (21.0/0.0)

(WS(t-15) >= 172.656) and (WS(t-15) >= 172.74296) and (RS(t-150) <= 0.2) and (WS(t-105) <= 172.81618)
and (WS(t-180) >= 172.77216) => Class=High (18.0/1.0)

(WS(t-15) >= 172.656) and (WS(t-45) <= 172.59279) and (WS(t-30) <= 172.56016) => Class=High (21.0/2.0)

(WS(t-15) >= 172.68333) and (WS(t-15) >= 172.7627) and (RN(t-150) <= 0.6) and (WS(t-30) <= 172.88149)
=> Class=High (21.0/3.0)

(RS(t-60) >= 1.2) and (WS(t-15) >= 172.46) and (RS(t-60) >= 2.2) and (WN(t-15) >= 209.5381) => Class=High
(9.0/1.0)

(WS(t-15) >= 172.68333) and (WS(t-75) <= 172.52421) and (RN(t-15) >= 0.2) => Class=High (11.0/0.0)

(WS(t-15) >= 172.63886) and (WS(t-30) <= 172.6397) and (WS(t-15) >= 172.721) => Class=High (4.0/0.0)

(RS(t-60) >= 1) and (RS(t-45) >= 7.2) => Class=High (6.0/2.0)

(WS(t-15) >= 172.63886) and (RS(t-75) >= 0.8) and (RN(t-150) <= 0.2) and (RN(t-135) >= 0.2) => Class=High
(5.0/1.0)

(RS(t-60) >= 1.6) and (RN(t-180) >= 0.6) => Class=High (2.0/0.0)

 => Class=Low (43335.0/3.0)

Number of Rules : 12
```

Figure 19: *JRip rules baseline 15 mins lead (data set 3)*

```
(WS(t-60) >= 172.656) and (WS(t-60) >= 172.814) and (RS(t-120) >= 0.6) => Class=High (194.0/7.0)

(WS(t-60) >= 172.65616) and (WS(t-60) >= 172.76817) and (WN(t-225) >= 209.67742) and (WN(t-75) >= 209.92887) and (WS(t-60) >= 172.83375) => Class=High (54.0/0.0)

(WS(t-60) >= 172.57523) and (RN(t-120) >= 0.6) and (RN(t-75) >= 0.4) and (RN(t-120) >= 1.2) => Class=High (34.0/0.0)

(WS(t-60) >= 172.588) and (WS(t-60) >= 172.74723) and (RS(t-195) <= 0.2) and (WN(t-195) >= 209.71209) and (WN(t-135) <= 209.78143) => Class=High (29.0/0.0)

(WS(t-60) >= 172.47728) and (WS(t-60) >= 172.74723) and (WS(t-60) >= 173.0798) and (WN(t-225) >= 209.509) => Class=High (36.0/3.0)

(RS(t-75) >= 0.4) and (RS(t-90) >= 1.2) and (RS(t-60) >= 1) and (RN(t-135) >= 0.4) => Class=High (30.0/1.0)

(WS(t-60) >= 172.42499) and (RN(t-90) >= 0.4) and (RN(t-135) >= 0.8) and (WS(t-60) >= 172.814) => Class=High (7.0/0.0)

(WS(t-60) >= 172.42499) and (RS(t-75) >= 0.6) and (RS(t-105) >= 1.2) and (RN(t-105) >= 0.6) => Class=High (15.0/0.0)

(WS(t-60) >= 172.6302) and (RS(t-105) >= 0.4) and (RN(t-75) >= 0.4) and (WN(t-90) >= 209.635) => Class=High (13.0/3.0)

(WS(t-60) >= 172.42499) and (WS(t-60) >= 172.74723) and (WN(t-60) >= 209.86307) and (RN(t-165) >= 0.4) => Class=High (15.0/2.0)

(WS(t-60) >= 172.42649) and (RS(t-90) >= 0.8) and (RS(t-90) >= 2) and (WN(t-60) >= 209.5176) and (WS(t-60) >= 172.46299) => Class=High (11.0/0.0)

(WS(t-60) >= 172.3937) and (RN(t-90) >= 0.6) and (RN(t-105) >= 1) and (WN(t-60) <= 209.5349) => Class=High (9.0/2.0)

(WS(t-60) >= 172.42499) and (WS(t-60) >= 172.8517) and (RS(t-180) <= 0) and (WN(t-135) >= 209.78439) => Class=High (12.0/3.0)

(RS(t-75) >= 1) and (RS(t-60) >= 3.2) and (RN(t-75) >= 1.4) => Class=High (12.0/0.0)

(RN(t-120) >= 0.4) and (RS(t-75) >= 1) and (RS(t-105) >= 1.2) => Class=High (10.0/3.0)

(WS(t-60) >= 172.42499) and (RN(t-120) >= 0.6) and (WN(t-60) >= 209.64842) and (RN(t-90) >= 0.6) => Class=High (11.0/2.0)

(RS(t-60) >= 0.4) and (RS(t-60) >= 2.2) and (WS(t-60) >= 172.2121) => Class=High (22.0/9.0)

(WS(t-60) >= 172.6192) and (WS(t-75) <= 172.6152) and (WS(t-60) >= 172.677) => Class=High (10.0/3.0)

(RN(t-120) >= 0.4) and (RN(t-120) >= 1.4) and (WN(t-90) >= 209.5349) and (WN(t-150) <= 209.539) => Class=High (9.0/3.0)

(WS(t-60) >= 172.43639) and (WS(t-60) >= 172.6152) and (WS(t-225) >= 172.89684) and (WS(t-165) <= 172.91077) and (WN(t-60) >= 209.77646) and (WN(t-105) <= 210.03572) => Class=High (9.0/1.0)

(RN(t-105) >= 0.6) and (RN(t-120) >= 2.8) and (RS(t-105) >= 1.2) => Class=High (4.0/0.0)

(WS(t-60) >= 172.38199) and (RN(t-90) >= 2) => Class=High (6.0/2.0)

(RS(t-90) >= 1) and (RN(t-105) >= 3.2) => Class=High (8.0/2.0)

(WS(t-60) >= 172.43639) and (WS(t-60) >= 172.80918) and (WS(t-135) <= 172.205) and (RN(t-135) <= 0) => Class=High (8.0/2.0)

(RN(t-60) >= 0.2) and (RS(t-75) >= 4.4) => Class=High (4.0/0.0)

(RN(t-60) >= 0.2) and (RN(t-60) >= 4.8) and (WS(t-60) >= 172.1954) => Class=High (6.0/2.0)

(WS(t-60) >= 172.63886) and (RS(t-210) >= 5.4) and (WS(t-90) <= 173.2354) => Class=High (3.0/0.0)

(WN(t-60) >= 209.72) and (WN(t-105) <= 209.72189) and (WN(t-135) >= 209.71) => Class=High (8.0/3.0)

Class=Low (43325.0/23.0)

Number of Rules : 29
```

Figure 20: *JRip rules baseline 60 mins lead (data set 3)*

To see where the misclassifications occur the misclassified tuples were recorded. Figure 21 displays a histogram indicating the water level value at the cross-section of interest for all misclassified tuples in the baseline experiments.

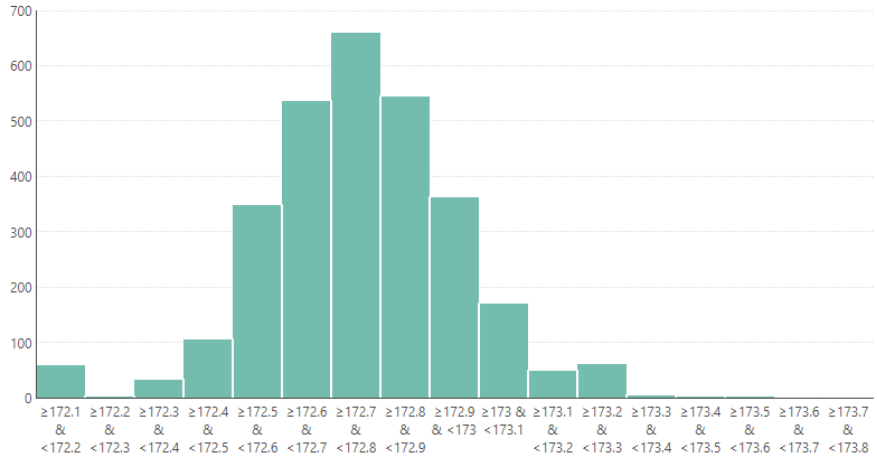

Figure 21: *Baseline misclassification histogram*

The results of this analysis show clearly that tuples are mostly misclassified as they approach the threshold for the event characterization function, which was 172.75 cm for this case study. The histogram is normally distributed, which also shows that an equal number of misclassifications occur above and below the threshold.

6.2 DELTA EXPERIMENTS

The following sections describe all experiments conducted using deltas. The results of adding deltas over the full window size are stated. Then the results for only partially added deltas are outlined.

6.2.1 DELTA OVER FULL WINDOW SIZE

Figure 22 shows the Precision and Recall results for the ensemble over the different lead-times for data set 3 after adding water level deltas, rainfall deltas and a combination of both (termed all deltas). The full results for all data sets can be found in the Appendix C.

When comparing the ensemble results from the baseline experiments with the results after deltas were added it is visible that adding deltas does affect the model performance. Considering all data sets, the results show

that especially all deltas or just the water level deltas have a positive effect on model performance while rain deltas can decrease model performance going below the baseline for both Precision and Recall.

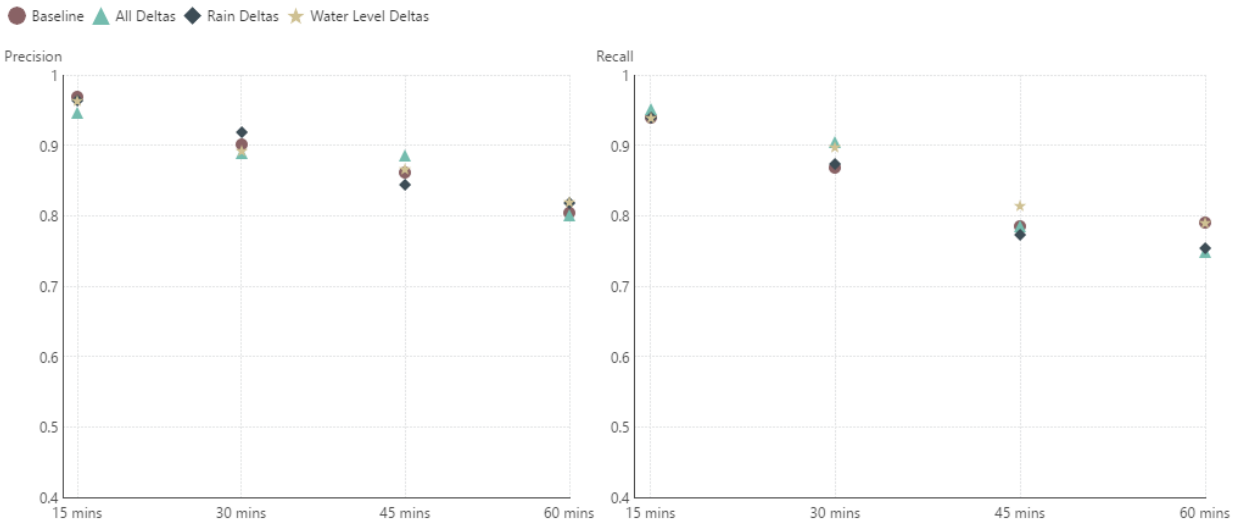

Figure 22: *Full window size delta precision and recall for ensemble (data set 3)*

All data sets confirm that the ensemble shows very robust results on data sets with and without deltas. However, in some cases, adding deltas was able to impact model performance to a great extend, e.g. the results for data set 5. Adding all deltas to the data set increases Precision from 0.88 to 0.93 and Recall from 0.81 to 0.86 for 30 minutes lead-time. For a lead-time of 60 minutes, water level deltas boost Recall from 0.67 to 0.75 while only showing a drop in Precision from 0.89 to 0.87.

The results also show that the positive effect is greatest for the lead-times of 30 and 45 minutes. The reason for this could be that for a 15 minutes lead-time not much uncertainty is present so that the classifiers do not require the additional delta information to correctly predict the events. For a lead-time of 60 minutes, deltas could introduce additional noise that decreases the model performance.

When looking at single classifier performance, adding deltas over the full window size shows the results as presented by Figure 23 for data set 3. The results for all data sets can be found in Appendix D.

Often an improvement in Recall comes at the cost of a decline in Precision and vice versa. However, the results show that through adding deltas it is possible to improve Recall without lowering Precision at the same time. SimpleCart, JRip, and J48 show the biggest improvements in both Precision and Recall after adding deltas. SimpleCart shows the greatest improvement with only water level deltas but also improvements when using all deltas. The improvement is visible for all lead-times and both Precision and Recall, however it is more consistent and apparent in the Recall results.

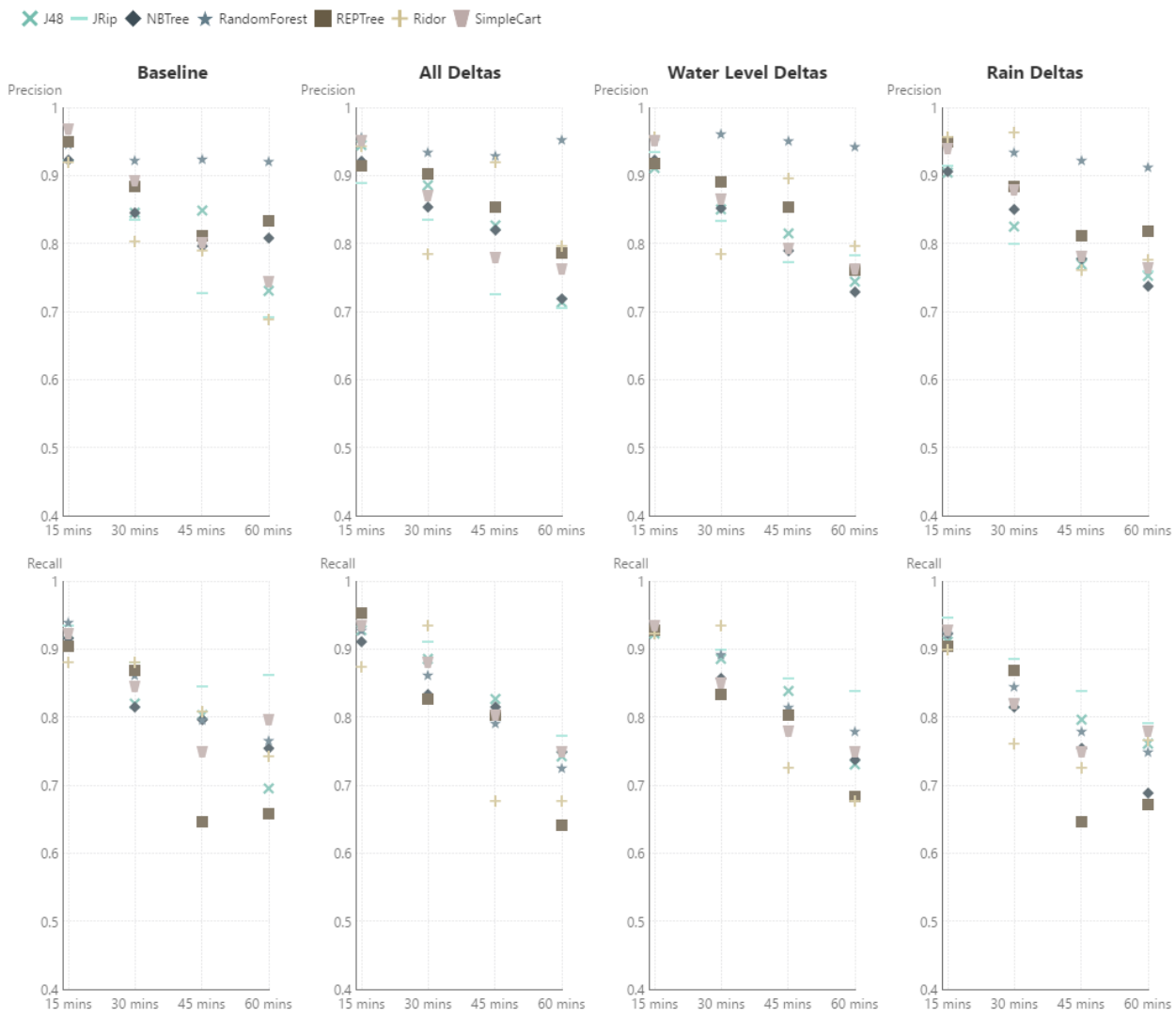

Figure 23: *Full window size delta precision and recall for single classifier (data set 3)*

For data set 5, both all deltas and water level deltas show the same Recall as the baseline for 15 minutes but then exceed the baseline for lead-times greater than 30 minutes by between 1.2% and 14%. Figure 24 shows the Precision and Recall results obtained by SimpleCart for the baseline and full water level delta experiments as a scatter plot for all data sets and lead-times. It shows improved Recall after water level deltas were added over the full window size especially for the longer lead-times. Adding water level deltas allows to achieve a Recall of 0.8 for longer lead-times which would allow to issue flood warnings earlier while still accomplishing the necessary accuracy.

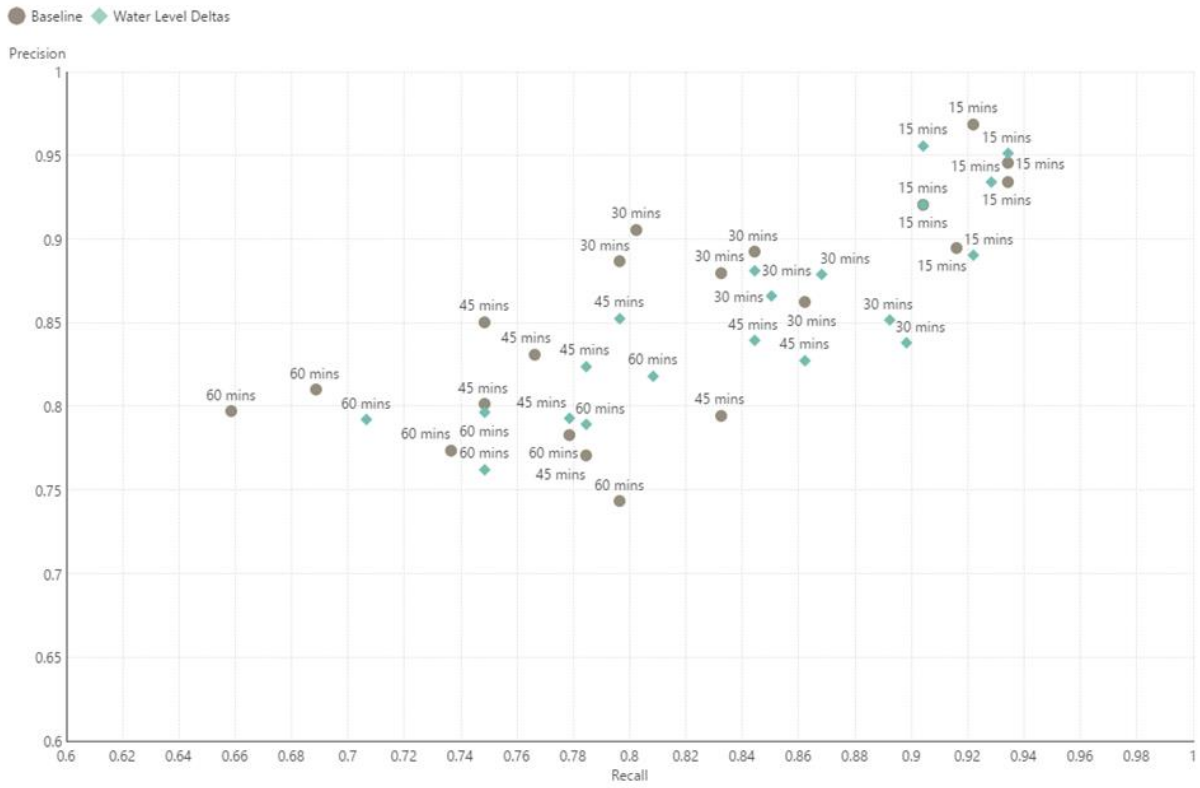

Figure 24: *SimpleCart results for baseline and full water level deltas and all data sets*

JRip shows improvements like SimpleCart in a sense that it achieves the best Precision and Recall results when using water level deltas, but still shows improvements when using all deltas. For data set 3, JRip exceeds the baseline Precision of 0.69 for 60 minutes lead-time with 0.78 when using water level deltas. However, unlike SimpleCart this increase in Precision often causes a drop in Recall and vice versa. For data set 5, JRip shows a Precision of 0.74 for the baseline and with water level deltas a result of 0.81, but a drop in Recall. When equally combining Precision and Recall, the F-Measure still shows mostly improvements for all deltas and water level deltas. For data set 3, water level delta shows a higher F-Measure than the baseline for all lead-times.

J48 also shows improvements for Precision and Recall when using all deltas or just the water level deltas for all lead-times. While the improvement is not as consistent across the different lead-times and data sets as for SimpleCart and JRip, it still has potential to boost Precision and Recall with a high magnitude. For data set 2, water level deltas show better Recall results than the baseline for all lead-times with an improvement of between 1% and 8.5 %.

Precision stays consistently at a high level for Random Forest whether deltas were added to the data sets or not. However, especially full water level deltas boost Recall performance across all data sets and lead-

times. For data set 5, water level deltas allow for higher Recall results than the baseline for all lead-times with the highest increase for the lead-time of 45 minutes. Here water level deltas show a Recall of 0.77 with a baseline of 0.72. This is an increase of 6.9 % while at the same time Precision also increases by 2.2 % from 0.92 to 0.94.

For REPTree, Precision results are inconsistent with whether adding deltas improves the performance. For Recall, overall water level deltas and all deltas seem to improve performance. In the case of data set 3, water level deltas achieve better results than the baseline for all lead-times except 30 minutes. For 45 minutes lead-time, Recall even improves from the baseline of 0.65 to 0.8 with added water level deltas. This is an increase of 23.1% while Precision increases at the same time by 4.9% from 0.81 to 0.85.

Ridor does overall improve Precision magnitude at the cost of decline in Recall when adding deltas. Especially for the longer lead-times, the added deltas cause Ridor to favor the majority class more, which causes high Precision results but low Recall. For data set 3 and a lead-time of 60 minutes, the baseline shows a Precision of 0.69. With all deltas or water level deltas the Precision reaches 0.8. However, at the same time Recall drops from the baseline of 0.74 to 0.68 for both all deltas and just water level deltas.

For some data sets, NBTree does benefit strongly especially from all deltas but also only water level deltas. For data set 2, all deltas show higher Precision results than the baseline except for the 15 minutes lead-time with the biggest increase of 9.3 % from 0.75 to 0.82 for the 60 minutes lead-time. Data sets 1 and 5 show similar results, the other data sets, however, do not confirm this trend and therefore leave the overall effect of adding deltas for NBTree on Precision results inconclusive. NBTree also shows improved Recall results mostly with all deltas however not for all data sets. Recall drops when deltas are added especially for a lead-time of 60 minutes.

Throughout the delta experiments, the constructed rule-sets and decision trees were analyzed so that they could be compared to the rule-sets and decision trees of the baseline. Figure 25 shows the rule-set defined by the JRip classifier for data set 3 and a lead-time of 15 minutes when all deltas were available, while Figure 26 shows the rule-set created for a lead-time of 60 minutes.

```
(WS(t-15) >= 172.74296) and (WS(t-15) >= 172.85114) => Class=High (394.0/5.0)

(WS(t-15) >= 172.7102) and (WS(t-15) >= 172.76619) and (WSDELTA(t-15) >= -0.0185) => Class=High (93.0/3.0)

(WS(t-15) >= 172.656) and (WSDELTA(t-15) >= 0.043) => Class=High (51.0/13.0)

(WS(t-15) >= 172.7102) and (RNDELTA(t-135) >= 0) and (WS(t-15) >= 172.77886) and (WSDELTA(t-60) <= 0.021) => Class=High (15.0/1.0)

(WS(t-15) >= 172.69236) and (WSDELTA(t-15) >= 0.0005) and (WNDELTA(t-165) >= 0.00294) => Class=High (5.0/0.0)

(WS(t-15) >= 172.46) and (RS(t-60) >= 2.6) => Class=High (9.0/3.0)

(WS(t-15) >= 172.75923) and (WN(t-15) >= 209.9018) => Class=High (8.0/1.0)

 => Class=Low (43339.0/10.0)

Number of Rules : 8
```

Figure 25: *JRip rules for full all deltas and 15 mins lead (data set 3)*

```
(WS(t-60) >= 172.6622) and (WS(t-60) >= 172.90734) and (RS(t-120) >= 0.6) => Class=High (152.0/2.0)

(WS(t-60) >= 172.58287) and (WS(t-60) >= 172.79015) and (WN(t-225) >= 209.5937) and (WN(t-105) >= 209.9338) => Class=High (73.0/3.0)

(WS(t-60) >= 172.5865) and (WSDELTA(t-60) >= 0.00101) and (RN(t-105) >= 0.6) and (RN(t-120) >= 1.2) => Class=High (52.0/3.0)

(WS(t-60) >= 172.45137) and (WS(t-60) >= 172.7232) and (RN(t-90) >= 0.4) => Class=High (70.0/21.0)

(WS(t-60) >= 172.42499) and (WS(t-60) >= 172.7498) and (WN(t-195) >= 209.71209) and (WSDELTA(t-60) >= -0.00699) => Class=High (44.0/8.0)

(RS(t-75) >= 0.6) and (RS(t-90) >= 1.2) and (WSDELTA(t-60) >= 0.004) => Class=High (65.0/14.0)

(WS(t-60) >= 172.42649) and (RN(t-75) >= 0.6) and (WSDELTA(t-60) >= 0.04899) => Class=High (23.0/4.0)

(WS(t-60) >= 172.38199) and (WS(t-60) >= 172.80918) and (WN(t-195) >= 209.721) and (WSDELTA(t-60) >= -0.0405) => Class=High (21.0/2.0)

(WS(t-60) >= 172.3937) and (WNDELTA(t-60) >= 0.00594) and (WSDELTA(t-60) >= 0.068) and (RNDELTA(t-135) >= 0) and (WNDELTA(t-165) >= -0.00106) => Class=High (13.0/3.0)

(RS(t-60) >= 0.6) and (RS(t-60) >= 2.2) and (RN(t-75) >= 2.2) => Class=High (18.0/4.0)

(WS(t-60) >= 172.495) and (WS(t-60) >= 173.14811) => Class=High (25.0/10.0)

(WS(t-60) >= 172.38199) and (RS(t-75) >= 0.8) and (WNDELTA(t-195) >= 0.00401) and (WSDELTA(t-60) >= -0.022) => Class=High (10.0/2.0)

(WS(t-60) >= 172.399) and (WS(t-60) >= 172.66885) and (WSDELTA(t-150) <= -0.0215) and (WSDELTA(t-225) <= 0.0175) and (WS(t-210) >= 172.87983) => Class=High (12.0/3.0)

(RN(t-105) >= 0.4) and (RS(t-105) >= 1.2) and (RNDELTA(t-120) >= 0.8) and (RS(t-150) >= 0.2) => Class=High (9.0/1.0)

(RS(t-60) >= 0.8) and (RS(t-60) >= 3.2) and (RNDELTA(t-105) <= -0.2) => Class=High (4.0/0.0)

(RN(t-105) >= 0.6) and (RS(t-90) >= 1) and (RNDELTA(t-90) <= -1.6) => Class=High (12.0/5.0)

(RN(t-75) >= 0.2) and (RN(t-90) >= 0.8) and (RS(t-75) >= 2.6) => Class=High (6.0/1.0)

(WS(t-60) >= 172.42499) and (RSDELTA(t-60) >= 2.2) => Class=High (4.0/1.0)

(WSDELTA(t-60) >= 0.0015) and (WS(t-120) >= 172.44697) and (WS(t-210) <= 172.46088) and (WS(t-210) >= 172.43132) => Class=High (8.0/2.0)

(RN(t-120) >= 0.4) and (WS(t-60) >= 172.6485) and (RN(t-75) >= 0.8) => Class=High (4.0/0.0)

 => Class=Low (43289.0/23.0)

Number of Rules : 21
```

Figure 26: *JRip rules for full all deltas and 60 mins lead (data set 3)*

Both rules, as well as the other rule-sets and decision trees analyzed, confirm that delta variables are heavily used by the classification algorithms. Especially water level deltas at the cross-section of interest are selected which is in line with the results obtained from the baseline. For the shorter lead-times, rain deltas are not as important but become more important and more present in the rule-sets and decision trees with increasing lead-times.

6.2.2 DELTA OVER PARTIAL WINDOW SIZE

Instead of adding deltas over the full window size, deltas were added only over the first or first two preceding hours. Figure 27 shows the Precision and Recall results for the ensemble over the different lead-times for data set 3. Each chart shows the results for the baseline and the deltas added over the full window size compared to the same deltas added only over the first or first two hours. The respective results for all data sets can be found in Appendix E.

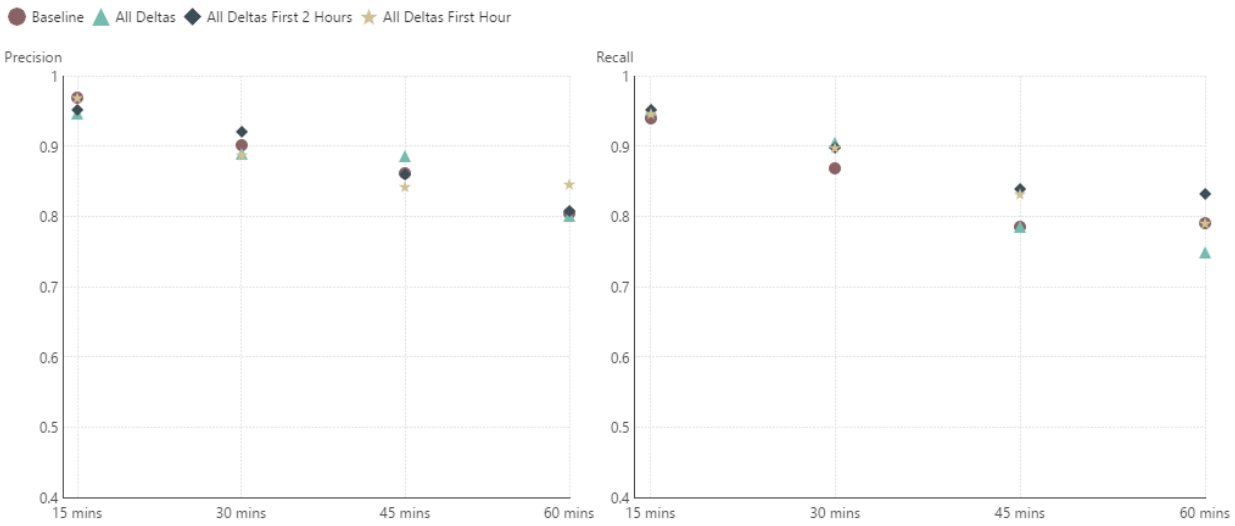

Figure 27: *Partial window size all delta precision and recall for ensemble (data set 3)*

Adding all deltas only for the first or first two hours clearly outperforms adding all deltas over the full window size. The difference is especially apparent in the Recall results where the partial delta results, unlike the full window size deltas, stay above the baseline for all lead-times. Adding all deltas only partially seems to negate the drop of performance that occurs towards longer lead-times when using deltas added over the full window size. For data set 5 and a lead-time of 60 minutes, adding all deltas over just the first two hours achieves a Recall of 0.74 exceeding the baseline of 0.67 and all deltas over the full window size with a Recall of 0.71. Another example of how partial deltas prevent a performance drop for larger lead-times is data set 1. While adding full deltas exceeded the baseline for a lead-time of 30 and 45 minutes, it fell below

the baseline for a lead-time of 60 minutes. Adding all deltas over only the first two hours, however, improves the Recall of the baseline for this lead-time by 5.3% from 0.75 to 0.79.

Adding only rain deltas, while improving model performance for some data sets and lead-times, does not seem to offer an overall improvement. Adding rain deltas only partially does not change this behavior as indicated in Figure 28 even though adding rain deltas only over the first 2 hours does improve both Precision and Recall performance compared to adding it over the full window size.

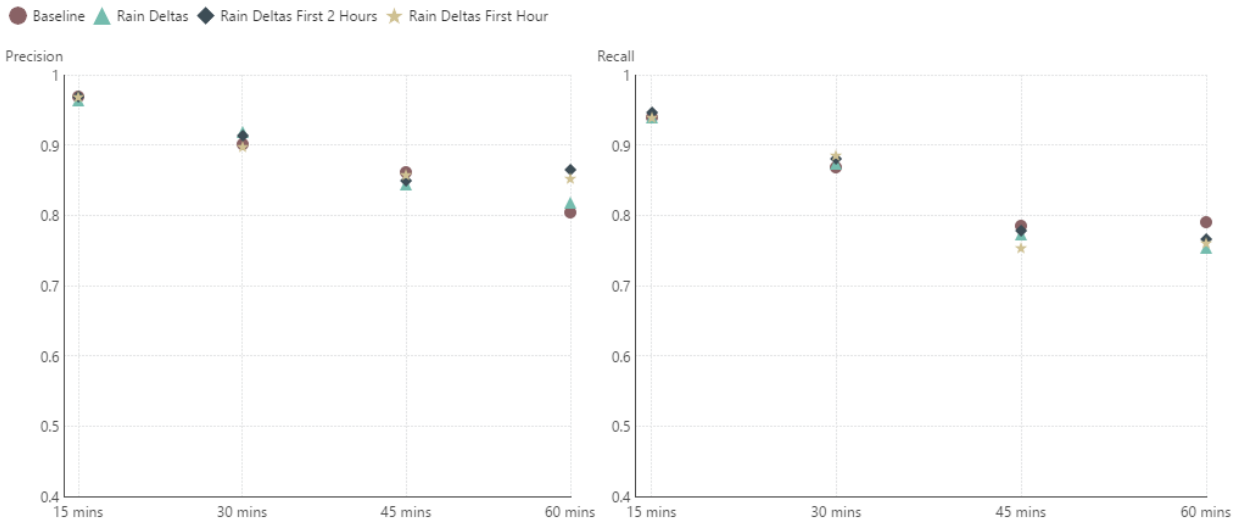

Figure 28: *Partial window size rain delta precision and recall for ensemble (data set 3)*

Data set 3 shows that rain deltas added over the first two hours outperform deltas added over the full window size for both Precision and Recall over all lead-times. In this case, the rain deltas over the first two hours show even higher Recall and Precision results than the baseline for some lead-times. The other data sets do show similar tendencies however not with the same consistency as data set 3.

Adding only water level deltas over the first or first two hours achieves results for data set 3 as presented by Figure 29. The results for the partial water level deltas show patterns that are similar to the results obtained from the partial all deltas. For data set 3, adding water level deltas over only the first two hours improves both Precision but also Recall results compared to the baseline and compared to adding the water level deltas over the full window size. Especially the Recall results show a great improvement as the water level deltas over the first two hours show consistently better results than the baseline over all lead-times. For a lead-time of 60 minutes, it allows a Recall of over 0.8 which is the threshold defined by the TRCA for operational flood management systems.

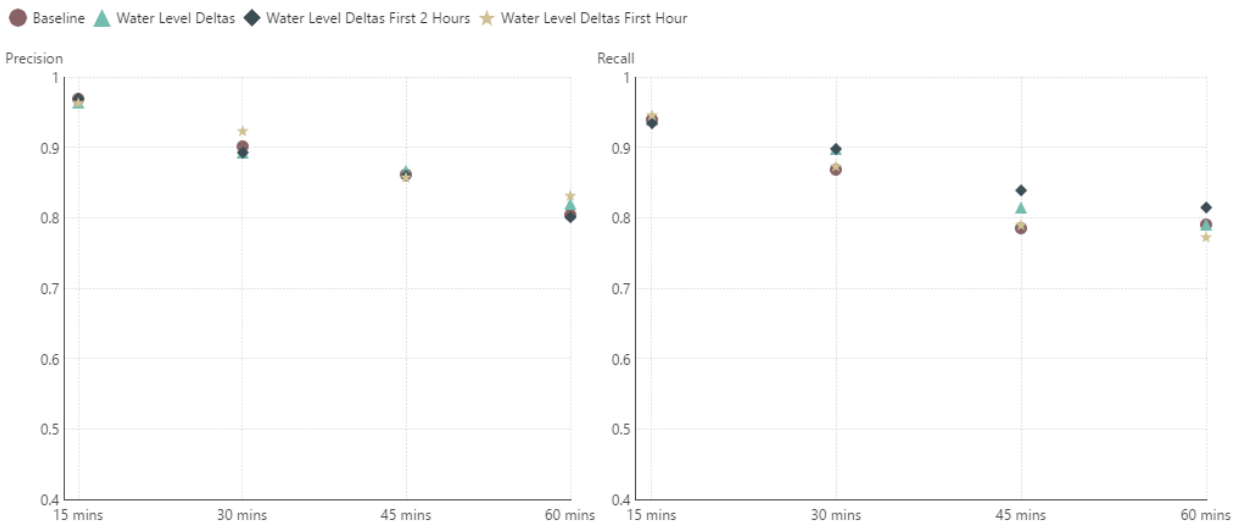

Figure 29: *Partial window size water level delta precision and recall for ensemble (data set 3)*

While the water level deltas over only the first hour also show promising results, the improvements are at a lower magnitude. The other data sets confirm this trend. For a lead-time of 30 and 45 minutes the water level deltas over the first two hours achieve the best results while for the other lead-times stay near the full window size water level deltas.

This section discusses the results of the individual classifiers. All partial delta results for these classifiers can be found in tabular form in Appendix F. The single classifier results show that adding all deltas over the first or first two hours does improve the model performance especially for Recall and a lead-time of 60 minutes, preventing the drop of performance that the full window size deltas are experiencing. Adding only rain deltas over the first or first two hours can improve Precision and Recall performance mostly for the 45 minutes lead-time. The results still stay below the baseline for most classifiers, lead-times, and data sets. This again shows that rain deltas on their own rather contain noise than useful information to the underlying hydrological processes. However, they show improved results when combined with the water level deltas. Adding water level deltas only over the first or first two hours shows better improvements on the Recall results than on the Precision results. The biggest improvement is visible for Recall and water level deltas for a lead-time of 60 minutes. Although the overall results do not necessarily show an improvement compared to the full water level deltas for all lead-times they still stay above the baseline.

In the following section, the effect of adding partial deltas is analyzed for every single classifier. Figure 30 shows the Precision and Recall results for the different variations of all deltas for data set 3.

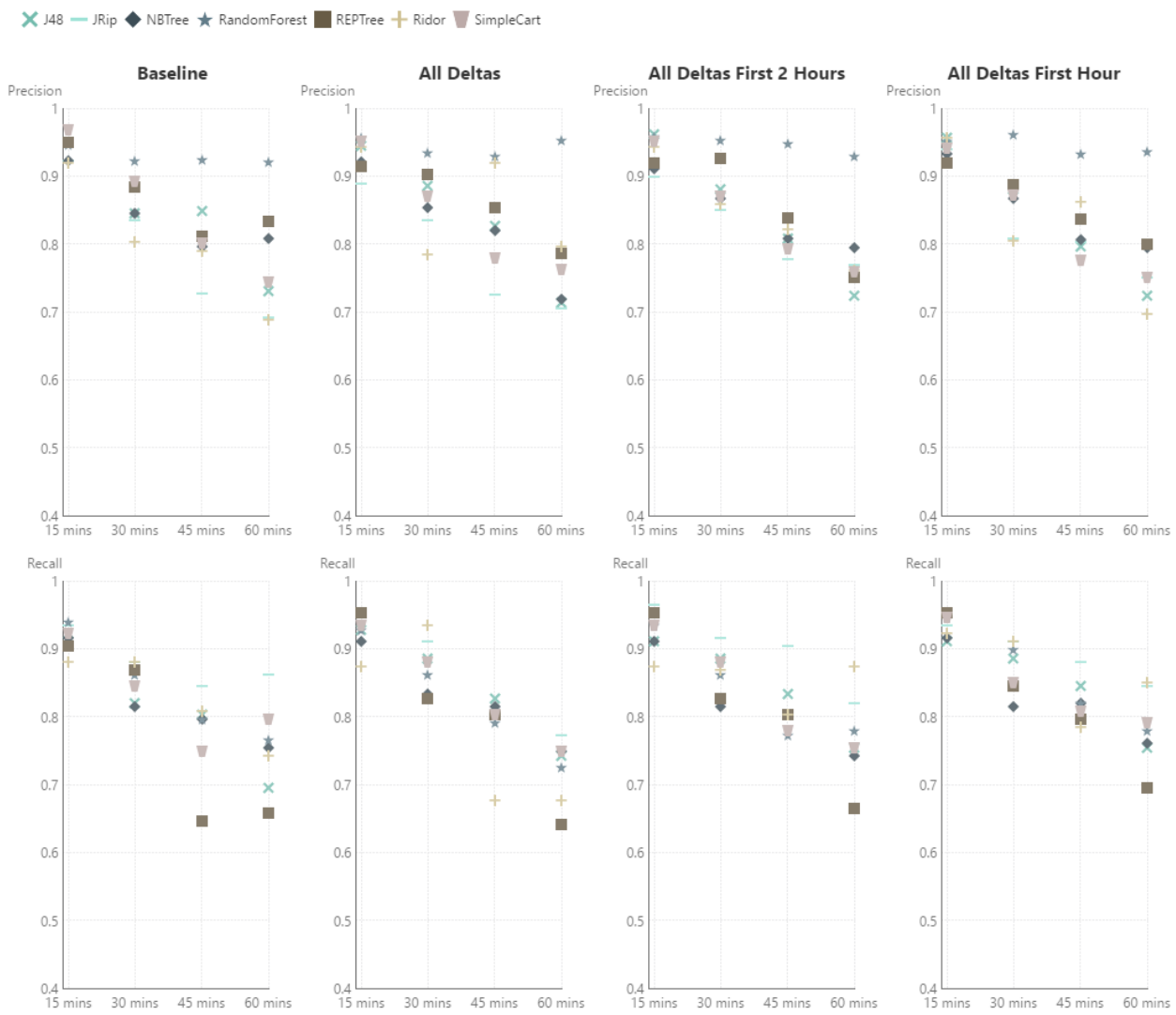

Figure 30: *Partial window size all delta precision and recall for single classifiers (data set 3)*

Both JRip and SimpleCart show clear improvements for both Precision and Recall when adding all deltas only over the first and first two hours. For data set 3, when applying JRip on all deltas over the full window size, Recall stays above the baseline for a lead-times of 15 and 30 minutes but falls below the baseline starting with a lead-time of 45 minutes. With the partial all deltas, both Precision and Recall increase and Recall for example stays above the baseline for all lead-times except 60 minutes. For a lead-time of 45 minutes for example Recall increases from the baseline of 0.84 to 0.90 for deltas over the first two hours. SimpleCart shows similar results. For data set 5 and a lead-time of 60 minutes, all deltas over the full window size achieve a Precision of 0.78 compared to the baseline of 0.8. Deltas only over the first two hours improve the baseline by 3% while also achieving the highest Recall of 0.74 compared to the baseline of 0.66 and the full window size results of 0.69. Noticeable is that partial deltas are especially improving

SimpleCart's Recall for the longer lead-times of 45 and 60 minutes. For data set 4, using partial deltas allows Recall to stay on or above the baseline for all lead-times. With the full window size deltas, this was not the case for a lead-time of 45 minutes. For this lead-time, partial deltas improved Recall from 0.75 to 0.83, which is an increase of more than 10%.

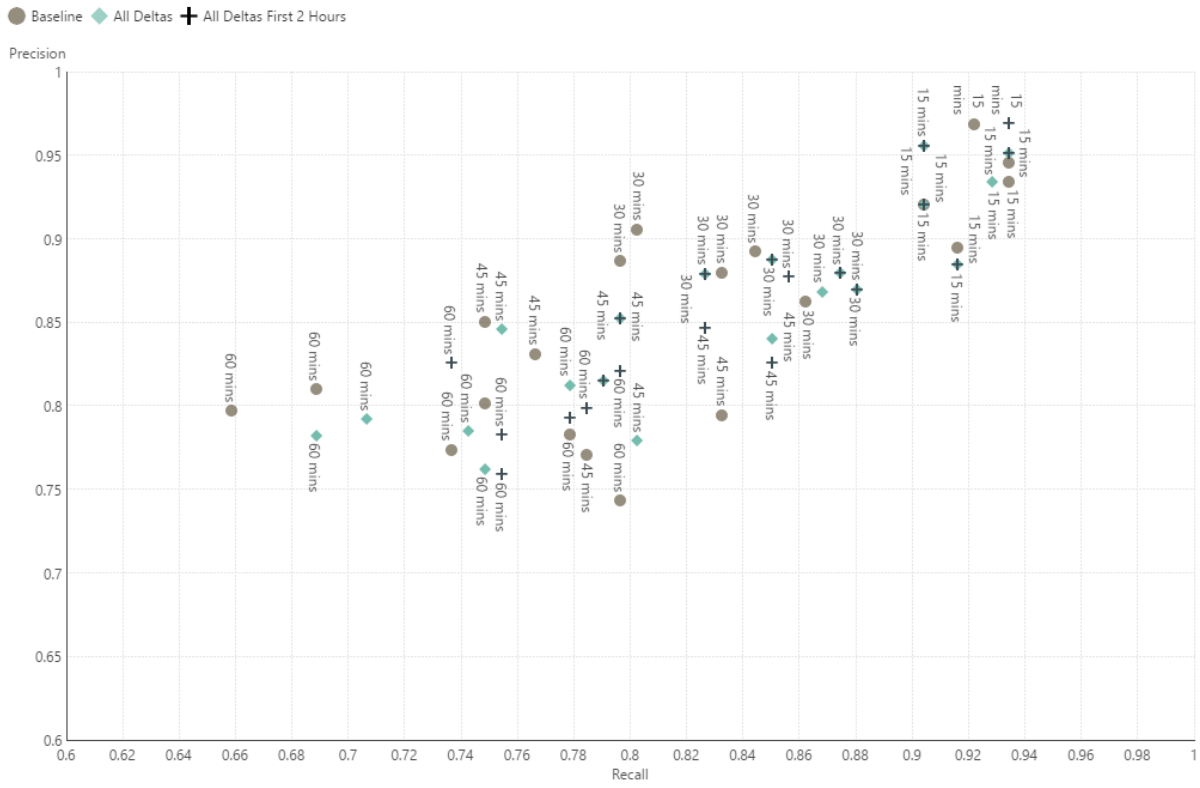

Figure 31: *SimpleCart results for baseline, full and partial all deltas and all data sets*

Figure 31 clearly supports the statement that partial all deltas improve performance for SimpleCart. They allow higher Recall results for longer lead-times even beyond the improvements already achieved by using all deltas over the full window size.

For REPTree and NBTree, partial deltas especially improve results for longer lead-times of 45 and 60 minutes. For NBTree, the drop in performance previously discussed for data set 3 and a lead-time of 60 minutes decreased using partial deltas. Where before all deltas over the full window size achieved a Precision of 0.72 that stayed below the baseline of 0.81, the partial deltas achieve a Recall of 0.79 and are significantly closer to the baseline. This allows the deltas over the first hour to stay above the baseline for all lead-times but 60 minutes, while staying near the baseline results for 60 minutes lead-time and at the same time showing a higher Recall for this lead-time. As a result, deltas over the first hour show a higher F-Measure than the baseline and the same F-Measure for 60 minutes lead-time. For REPTree adding only

partial deltas does not allow Precision to go over the baseline in cases where the deltas over the full window size did not. On the other hand, partial all deltas improve Recall especially for the lead-time of 60 minutes. All deltas over the first hour consistently across all data sets shows Recall results above or on the baseline for 60 minutes lead-time while all deltas over the full window size showed lower or the same Recall results.

Like REPTree, Random Forest's Precision does not benefit from partial all deltas and mostly shows best results with all deltas over the full window size. However, Random Forest's Recall magnitudes increase after adding deltas only partially. Using all deltas over the first or first two hours causes Recall results to stay above the baseline for almost all data sets and lead-times consistently. With all deltas over the full window size, data set 3 shows lower Recall results than the baseline. With partial deltas, Recall is above the full deltas for all lead-times, but 30 minutes. The other data sets confirm these findings. As a result, Random Forest shows consistently better F-Measure results than the baseline except for the lead-time of 45 and 60 minutes of data set 2 and for the lead-time of 45 minutes of data set 4.

For J48, partial deltas can improve Recall, however, not to the same extent as shown by the other classifiers mentioned above. Recall seems to improve especially for the longer lead-times. For data set 3, the lead-time of 15 minutes shows best results for all deltas over the full window size with a Recall of 0.93 and the partial deltas tie with the baseline at 0.91. For 30 minutes lead-time, all delta variations tie at 0.89 while the baseline achieves a Recall of 0.82. For 45 and 60 minutes, the partial deltas achieve a Recall of 0.75, which is above the baseline of 0.69 and the full window size deltas results of 0.74. Ridor does not show any improvement for Recall or Precision and displays similar inconsistent results as when using all deltas over the full window size.

The same analysis as for partial all deltas was conducted for partial rainfall deltas. Figure 32 shows the results of this analysis on data set 3. Even though partial rain deltas do not achieve the same Precision and Recall results as all deltas or just the water level deltas, for the classifiers J48 and REPTree they show prediction improvements compared to adding rain deltas over the full window size. For REPTree, for example, rain deltas over the first hour show the biggest improvements. Mostly Recall increases, but not enough to achieve results above the baseline. For data set 4 and a lead-time of 15 and 30 minutes, all rain delta variations show the same Recall as the baseline. For a lead-time of 45 and 60 minutes, rain deltas over the first hour show the highest Recall results with 0.78 and 0.72 compared to the baseline of 0.77 and 0.69 and the full window size results of 0.78 and 0.69. For JRip, NBTree, Random Forest, and SimpleCart neither Precision nor Recall results improve when adding rain deltas only over the first or first two hours of the window size.

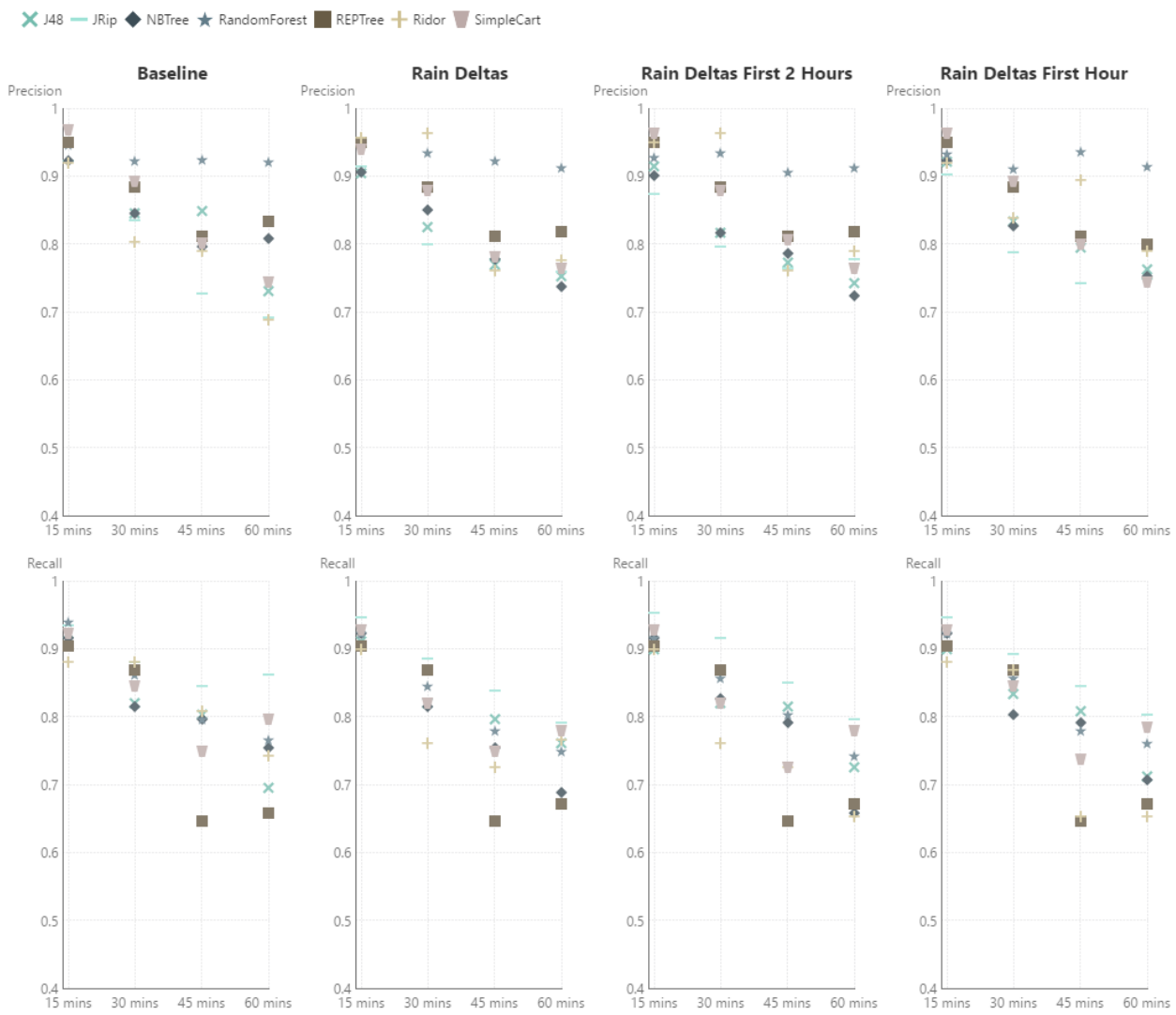

Figure 32: *Partial window size rain delta precision and recall for single classifiers (data set 3)*

Overall, partial water level deltas similar to all deltas do have the ability to improve prediction results for the single classification algorithms. Figure 33 shows the Precision and Recall results for the different variations of water level deltas for data set 3. Similar to the partial all delta results, JRip shows improvements when adding water level deltas only over the first and first two hours compared to the full window size. The improvement is greater for water level deltas over the first two hours for both Precision and Recall but water level deltas over the first hour improve performance as well. For data set 1, the F-Measure for the full water level deltas is below the baseline for all lead-times. When adding deltas only partially over the first two hours it stays above the baseline for all lead-times with the biggest improvement for lead-time 30 mins and 45 mins of 2.3%.

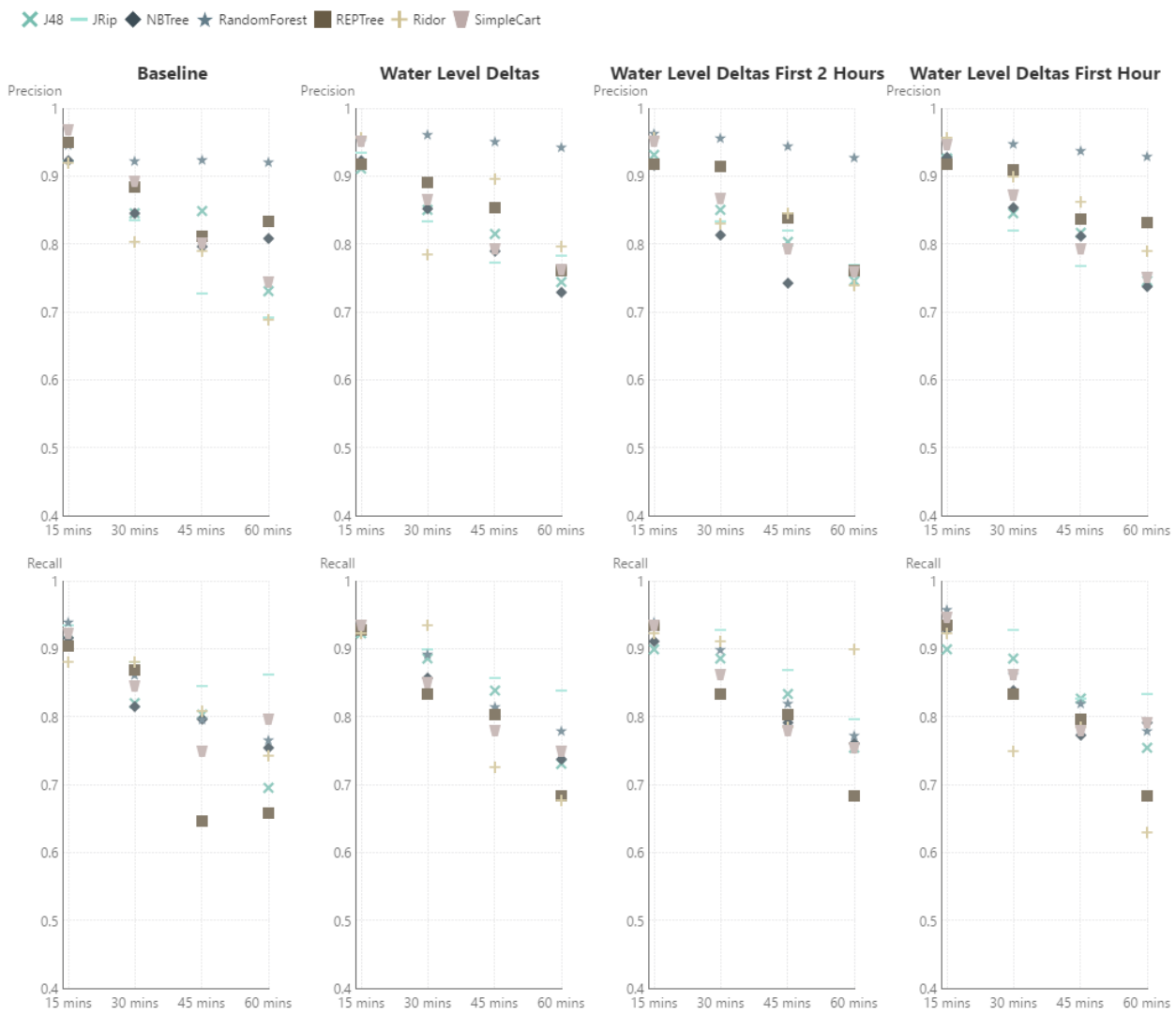

Figure 33: *Partial window size water level delta precision and recall for single classifiers (data set 3)*

Like the partial all delta results, SimpleCart does improve both Precision and Recall performance through the use of both variations of partial water level deltas. However, the improvements have a lower magnitude than the ones experienced when using all deltas. For data set 1 and a lead-time of 60 minutes, water level deltas over the first two hours achieve a Recall of 0.75 improving the baseline of 0.69 as well as the full window results of 0.71, while also achieving the highest F-Measure results. Figure 34 shows the Precision and Recall results for SimpleCart comparing the baseline with the full water level deltas and the water level deltas over the first two hours. Figure 34 confirms the previous analysis and shows that water level deltas over the first two hours improve especially Recall performance for longer lead-times.

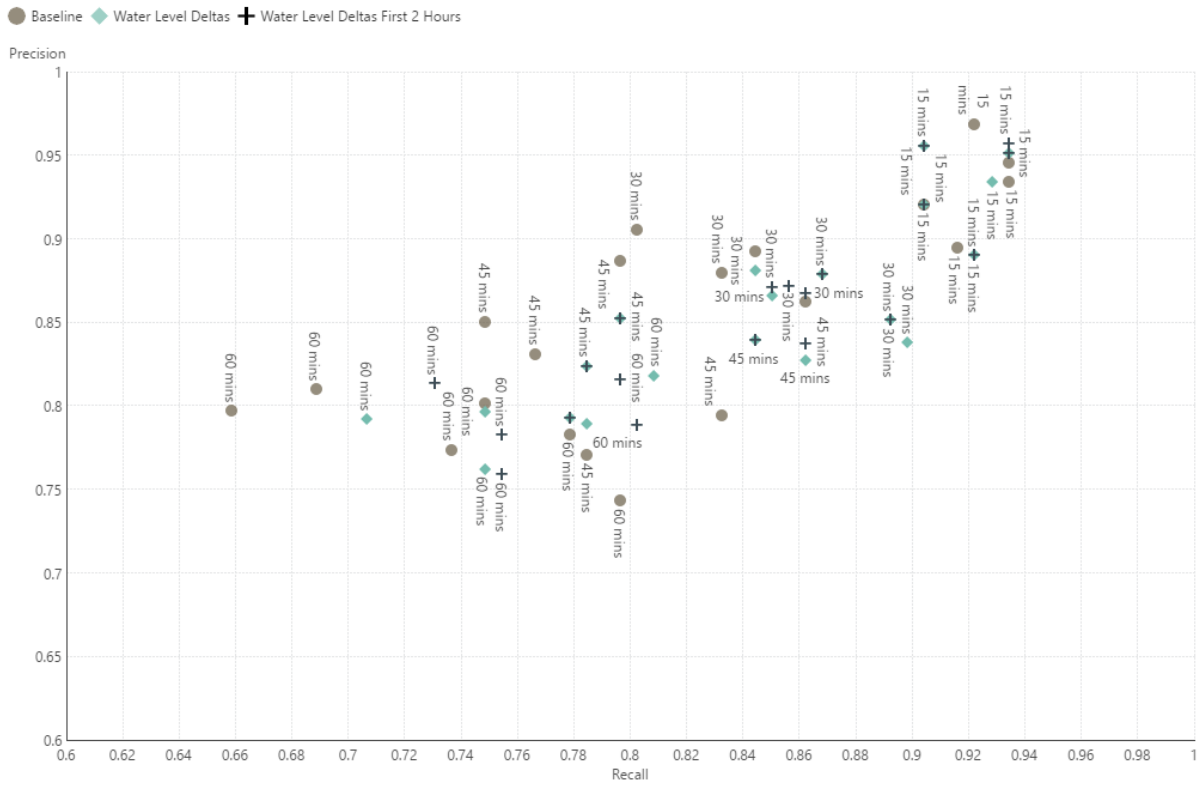

Figure 34: *SimpleCart results for baseline, full and partial water level deltas and all data sets*

Random Forest shows a similar trend for partial water level deltas than it does for partial all deltas as described earlier. While Precision stays constant, Recall shows a clear improvement for the longer lead-times compared to the full window size water level deltas. Adding deltas over the first and first two hours show similar improvements. For data set 1, Recall for full water level delta was below the baseline for 15 and 60 minutes lead-times. With both partial water level deltas, Recall stays above baseline for all lead-times. For 45 minutes, Recall improves from the baseline of 0.72 and the full water level delta results of 0.75 to 0.77 for water level deltas over the first two hours.

Unlike for the all delta experiments, Ridor does show improved performance when adding water level deltas only over the first or first two hours. Especially water level deltas over the first hour improves the Recall results increasingly with increasing lead-time. Yet, this also causes a decrease in Precision in some cases. For data set 1, water level deltas over the first hour show the best F-Measure results across all lead-time. The highest improvement is achieved over a lead-time of 60 minutes where water level deltas over the first hour achieve an F-Measure of 0.77 with a baseline of 0.74 and the full water level deltas at 0.72. This is the result of an increased Recall while keeping the Precision close to the baseline. For the lead-time of 60 minutes, water level deltas over the first hour show a Recall of 0.78, which is an improvement of 25.8% to the full water level deltas and 13% to the baseline.

While partial water level deltas show some improved performance for NBTree, the results are inconsistent over the different data sets and lead-times. For data set 3 and a lead-time of 60 minutes, NBTree using water level deltas over the full window size achieves a Recall of 0.74, which is below the baseline of 0.75. Water level deltas over the first hour achieves a Recall of 0.79. For a lead-time of 45 minutes in contrast, water level deltas over the first hour achieve the lowest Recall of 0.77 with a baseline of 0.8.

For both REPTree and J48, partial water level deltas neither increase nor decrease the model performance.

6.3 INFORMATION GAIN AND RELIEF ALGORITHM FOR VARIABLE SELECTION

To support the results of the exploratory computations, two filter approaches for subset selection, Information Gain and Relief, were applied. Both approaches were readily available from the R package mlr using the generateFilterValuesData function (Bischl, 2018). Information Gain is a common measure for variable relevance that is either applied using the filter approach or embedded in classification algorithms such as J48 and JRip. The Relief algorithm has successfully been applied to variable selection problems in different areas (Gore & Govindaraju, 2013; Koutanaei et al., 2015). Both Information Gain and Relief results support the assumption that the delta variables supply relevant information for the prediction task.

For both Information Gain and Relief water level variables from the cross-section of interest, sensor WS, achieve the highest results within 60 minutes before the actual event. The closer they are to the actual event, the higher is their relevance. As a result, the measured water level values 15 minutes before the event show the highest results with an Information Gain of 0.0615 and a Relief value of 0.1911. Close to the results of the actual water level values are the results of the deltas derived from the same location and time stamps. For Information Gain deltas from sensor WS 30 minutes before the event achieve the sixth highest results with 0.0355. For Relief, deltas from sensor WS and 15 minutes before the event achieve the third highest results of 0.1174. Both filter methods also agree that water level deltas from sensor WS have a higher importance than variables from sensor WN. While Information Gain shows moderate to low results for variables from WN, Relief clearly identifies this location to be the one producing the lowest ranking variables overall for all time stamps and both actual water level values and their deltas.

Both methods also agree that rainfall variables and their deltas are lower ranked than the water level variables and deltas from the cross-section of interest at location WS. It is also apparent that rainfall variables and their deltas show bigger Information Gain and Relief values from 45 minutes to 2 hours before the actual event. Before and after this period they show lower results. The rainfall variable with the highest Information Gain was measured at the location RS, 105 minutes before the event, with an overall rank of 19 and an Information Gain of 0.0251. At the same time, the rainfall variable with the lowest Information

Gain was measured 15 minutes before the event at sensor RN. This can be explained by the fact that the result of a heavy rainfall at location RN must first flow south before it can result in a flood event at location WS which takes longer than 15 minutes. Relief supports these findings with the highest-ranking rainfall variable measured at location RS, 45 minutes before the event, with a rank of 7 and a Relief value of 0.0735. The lowest-ranking rainfall variable was measured at the same location, 240 minutes before the event, with a Relief value of 0.0013. In this case, heavy rainfalls cause the flash floods to occur less than 240 minutes before the event so that earlier rainfall measurements are not relevant. For both Information Gain and Relief, rainfall variables from both locations achieve similar results. For both locations, the derived delta variables show lower Information Gain and Relief values compared to the actual rainfall values. For Information Gain, the highest rainfall delta variable is from location RS, 90 minutes before the event, with a rank of 38 and an Information Gain of 0.0214. For Relief, the highest-ranking rainfall delta variable is at location RN, 45 minutes before the event, at a rank of 26 and a Relief value of 0.0307.

The overall findings that water level deltas achieve higher Information Gain and Relief results than rainfall deltas support the earlier findings that showed a better model performance when water level deltas were added than rainfall deltas. Furthermore, the results show that water level deltas achieve highest Information Gain and Relief values closest to the actual event and decrease in relevance the further they are in the past. This supports the findings of the partial delta experiments that showed an improvement in model performance when water level deltas were only added over the beginning of the window size closest to the event. It also explains why doing the same with rainfall deltas did not result in a similar increase as rainfall deltas show highest Information Gain and Relief towards the middle of the window size between 45 minutes and 2 hours before the event. Especially for the short lead-times, this means that many of the non-relevant rainfall deltas closer to the event could have introduced noise, affecting model performance. The complete Information Gain and Relief results can be found in Appendix J and K respectively.

6.4 IMBALANCED DATA EXPERIMENTS

This chapter presents the results of the imbalanced data experiments. All experiments were conducted on data sets containing no deltas (baseline) and all deltas. The reason for this is to see if the methods applied cause different results on data sets containing delta variables and on those who do not. The previous experiments have shown that the combination of water level and rainfall deltas provides important information about the underlying hydrological processes. Therefore, the decision was made to use all deltas for the imbalanced data experiments. The following results show that in both cases the applied methods affect model performance in a similar way. Even though future experiments could include applying the

methods to data sets with only water level deltas, the results are not expected to vary from the results retrieved so far.

Figures 35 and 36 show the impact of the SMOTE and Tomek links technique on the ensemble results as well as the results for both techniques combined. While Figure 35 shows the results on the baseline data sets, Figure 36 shows the results on the data sets containing all deltas over the full window size. The ensemble results for all data sets are listed in Appendix G.

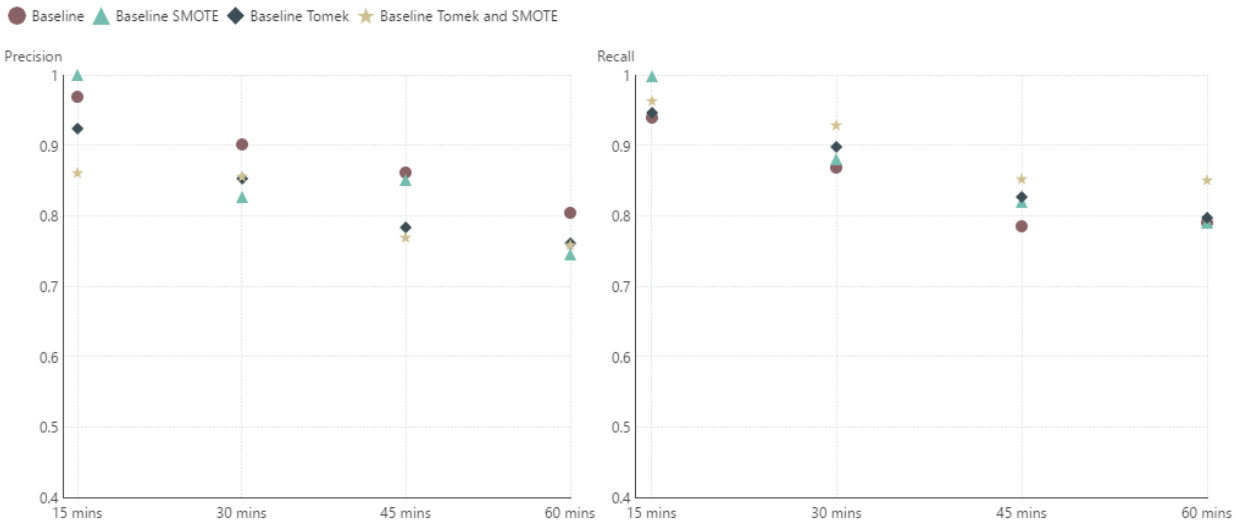

Figure 35: *Precision and recall for imbalanced data techniques on baseline (data set 3)*

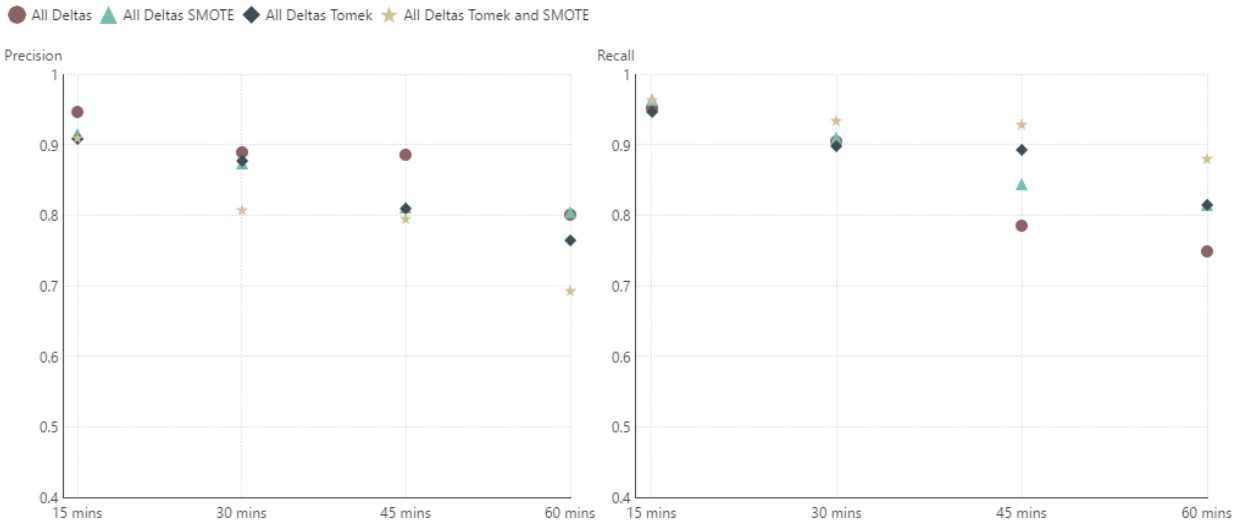

Figure 36: *Precision and recall for imbalanced data techniques on full all deltas (data set 3)*

Both figures show clearly that applying the SMOTE or Tomek links techniques cause an increase in Recall at the cost of a decline in Precision. Across all data sets, combining both methods shows the highest Recall improvement as well as the lowest Precision decline. Comparing the SMOTE and Tomek links techniques they achieve similar results and there is no clear winner over the different lead-times and data sets. Both SMOTE and Tomek links, therefore, cause the classifier to favor the minority class, which explains the drop in Precision and the increase in Recall. Furthermore, there is no difference between the effect of Tomek links and SMOTE on the data sets without any deltas and the data sets that contained all deltas over the full window size.

The results of the single classifiers show clearly that applying the Tomek links and SMOTE methods have the same effect on the single classifier than they have on the ensemble. It also seems that the longer the lead-time, the higher is the increase in Recall and the decrease in Precision. Overall SMOTE shows slightly better Recall and Precision results than Tomek links.

To see how the different classification algorithms react to the applied techniques, the figures 37 and 38 show the results for the single classifier on data set 3. Figure 37 shows the affect on the baseline and Figure 38 on data sets using all deltas over the full window size. The results for the remaining data sets can be found in Appendix H for the baseline results and in Appendix I for the all delta results.

Most classification algorithms react to the SMOTE and Tomek links method in a similar way. JRip, NBTree, Random Forest, REPTree and SimpleCart show improved Recall at the cost of decreased Precision across all data sets and lead-times.

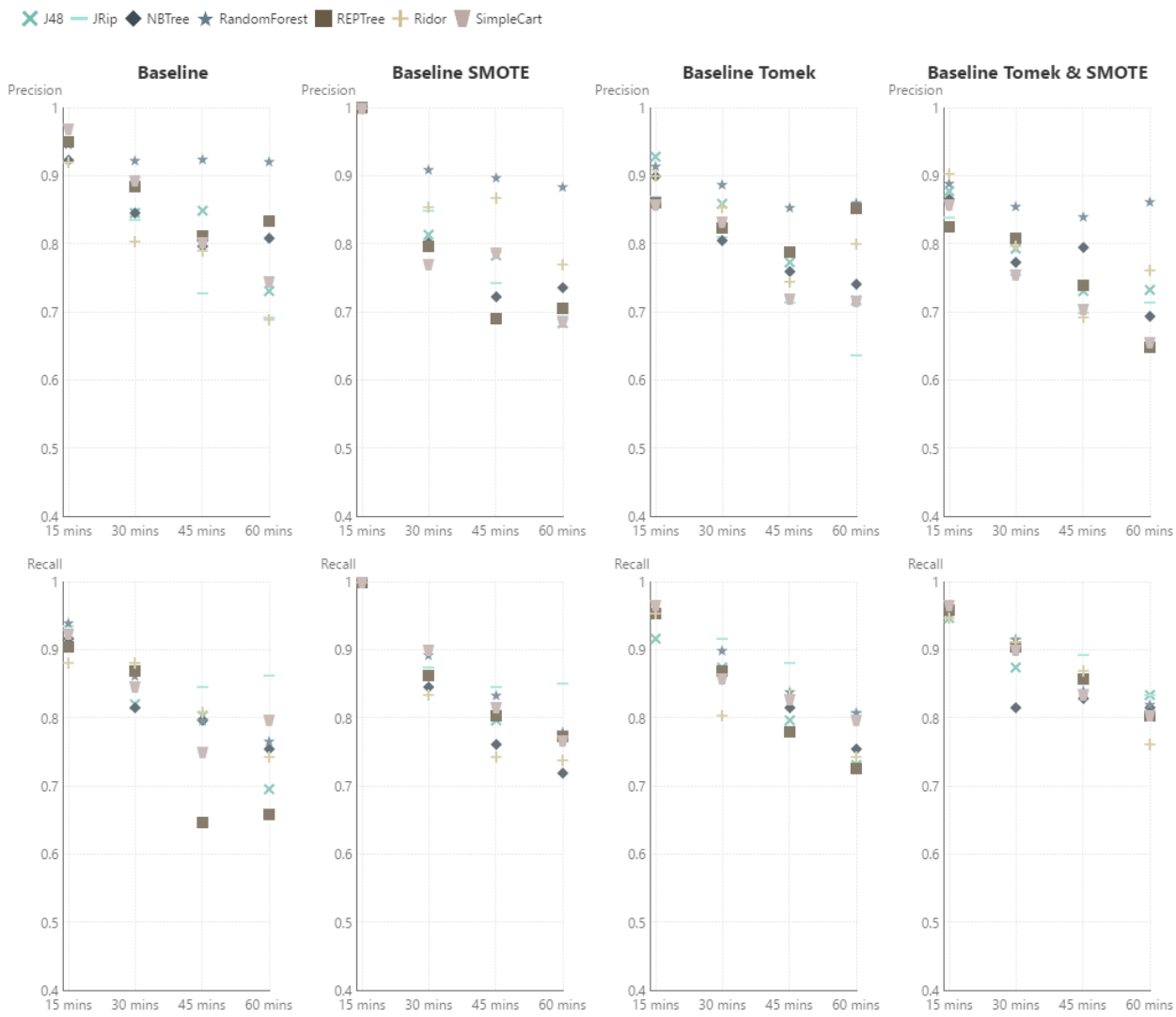

Figure 37: *Baseline precision and recall for single classifiers after applying imbalanced data techniques (data set 3)*

Overall, SMOTE provides better results than Tomek links when comparing the F-Measure. However, for specific classifiers, this changes depending on the lead-time, data set, and whether no deltas or all deltas are being used. Therefore, there is no clear winner for a single classifier. REPTree shows the biggest impact for both techniques with the highest increase in Recall but also the highest decrease in Precision. For data set 3 and a lead-time of 60 minutes, REPTree shows a Recall of 0.81 when applying SMOTE on the all deltas data set compared to the initial all delta result of 0.64. This equals an increase in Precision of 26.6%. On the other side, Precision decreases from 0.79 to 0.75 by 5%. For JRip and the same data set and lead-time, SMOTE increases Recall by 7.8% from 0.77 to 0.83 while also increasing Precision from 0.7 to 0.72 by 2.9%. For NBTree SMOTE increases Recall by 4% from 0.75 to 0.78 while increasing Precision from 0.72 to 0.77 by 6.9%.

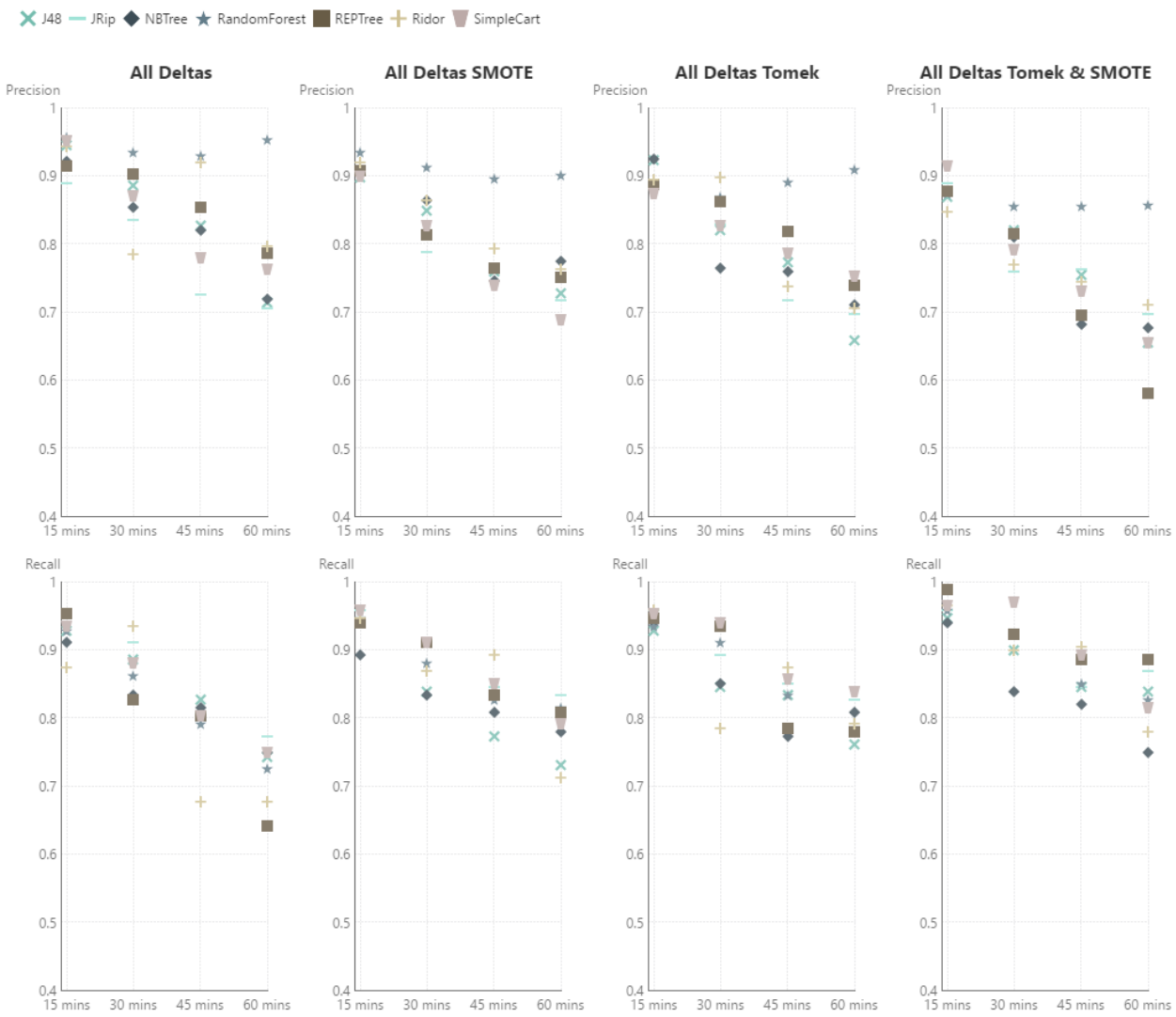

Figure 38: *All deltas precision and recall for single classifiers after applying imbalanced data techniques (data set 3)*

For SimpleCart SMOTE increases Recall by 5.3% from 0.75 to 0.79 and but decreases Precision from 0.76 to 0.69 by 10.1%. Figure 39 shows how applying the SMOTE and Tomek links techniques affects the prediction results for SimpleCart. It confirms the positive effect on SimpleCart's Recall when using SMOTE and Tomek links for all lead-times. This allows even predictions on a lead-time of 60 minutes to be close to or above the TRCA threshold of 0.8.

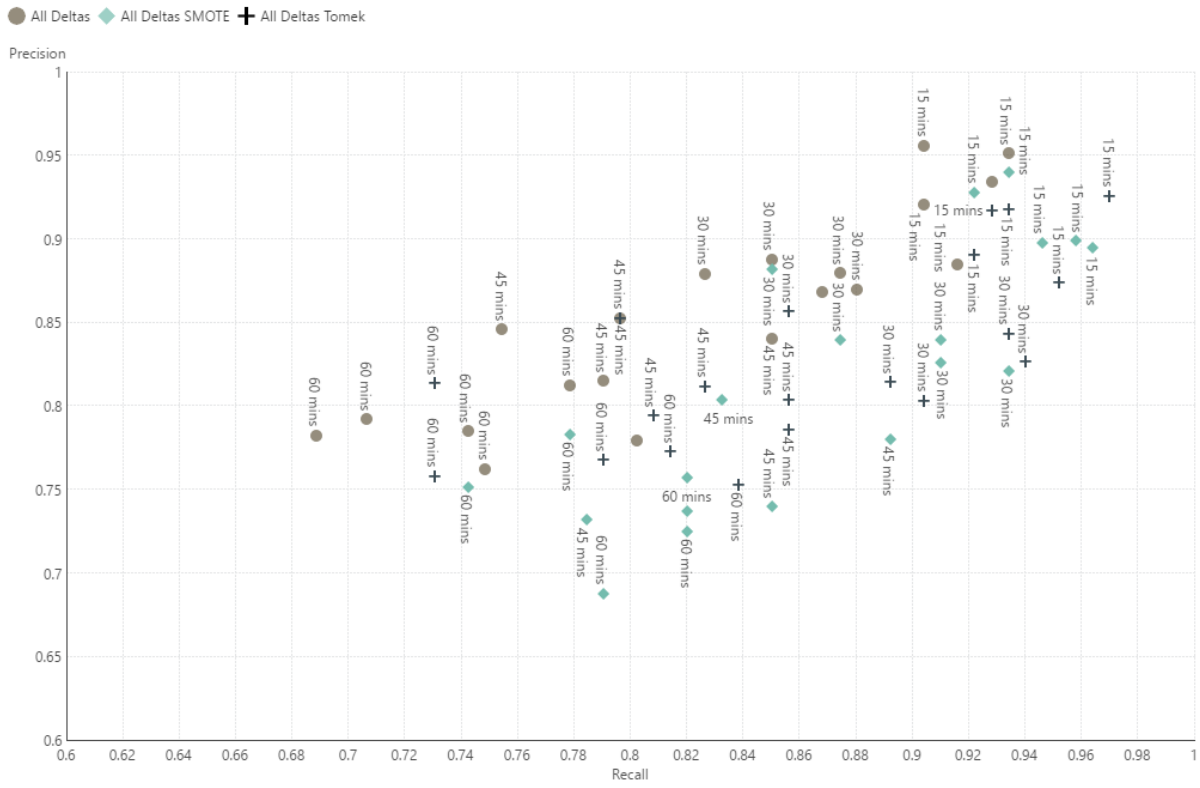

Figure 39: *SimpleCart results for all deltas and after applying SMOTE and Tomek links on all data sets*

For Random Forest, Precision declines when using Tomek links or SMOTE. However, the Precision results stay above 0.8 for all lead-times. Using Tomek links, SMOTE or both combined shows higher Recall results than the baseline or the all delta results across all data sets and lead-times. With a few exceptions, SMOTE shows higher Recall than Tomek links. For data set 3 and using all deltas, SMOTE allows Recall results to stay above the 0.8 threshold for all lead-times while at the same time keeping Precision above 0.9. As a result, Random Forest shows the same F-Measure with and without SMOTE for lead-times of 15 and 30 minutes on the all deltas data sets. For the lead-times of 45 and 60 minutes, however, using SMOTE increases the F-Measure especially for the lead-time of 60 minutes from 0.82 to 0.86 by 4.9%.

For most lead-times and data sets, J48 shows similar behavior to the classification algorithms mentioned previously. However, in some cases, using SMOTE decreases Recall and increases Precision instead. For data set 3 and all deltas, J48 shows lower Recall with SMOTE than without for all lead-times while it shows the highest Precision results. While J48 shows a behavior that is not consistent with the other classification algorithms for only some of the data sets and lead-times, Ridor shows completely dissimilar results. In the case of Ridor, results vary highly from one lead-time to another. For data set 3 and all deltas, using no imbalanced data technique allows for the best Precision results for all lead-times except 30 minutes. For the lead-time of 30 minutes, SMOTE and Tomek links suddenly show Precision results that are 10.3% to

15.4% higher than when no technique is used. The Recall results show the reversed pattern so that for a lead-time of 30 minutes using no technique returns Recall results that are 6.8% to 19.2% higher than when using Tomek links or SMOTE. Other data sets show similar inconsistent behavior.

7 CONCLUSIONS

While the results of the experiments were obtained on data collected at Spring Creek watershed, given that, hydrological processes exhibit common dynamics in highly urbanized watersheds, the findings allowed to propose a framework useful for short-term flash flood predictions at small urbanized watersheds. The framework relies on data collected by a flood monitoring network providing water level and rainfall measurements from multiple observation sites. The framework is an extension of the framework proposed by Erechtchoukova et al. (2016). Figure 40 outlines the steps of the framework which are then explained in more detail.

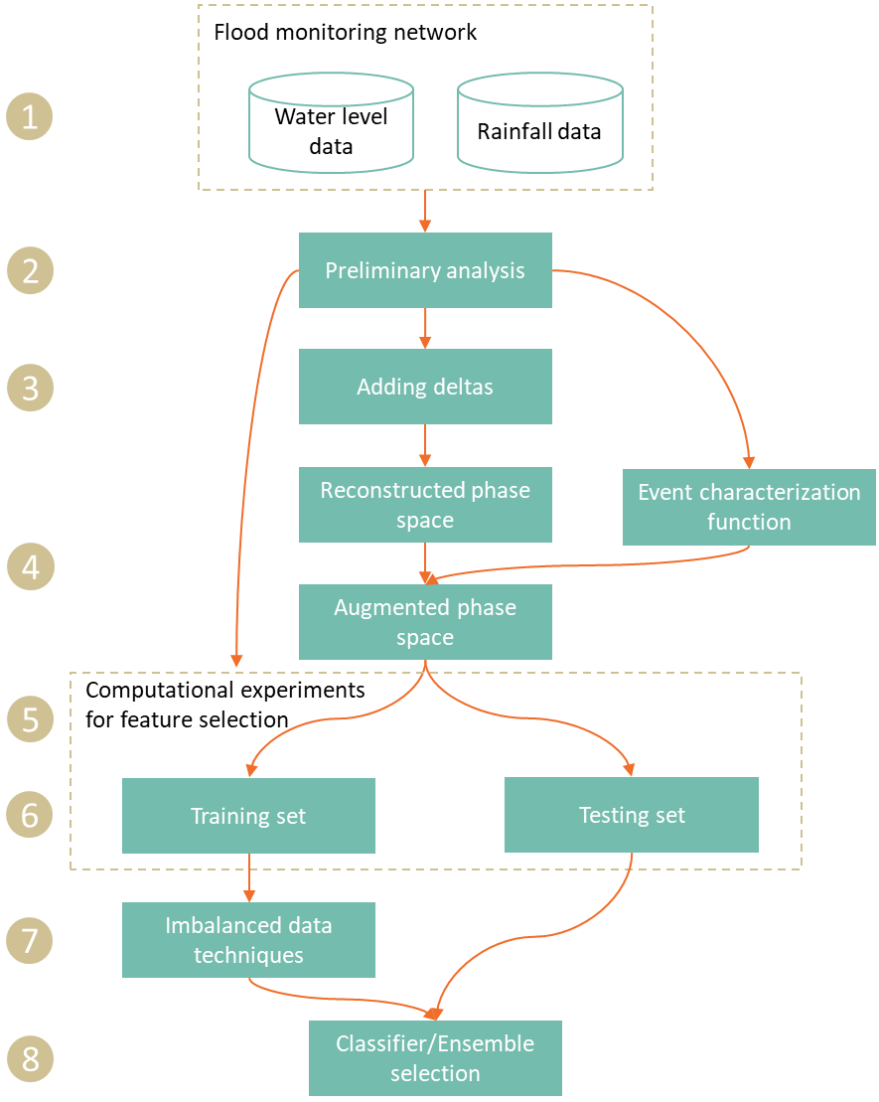

Figure 40: *Framework for short-term flash flood prediction at small urbanized watersheds*

Step 1: Data input

The framework issues data collected by a flood monitoring network on water level and rainfall. Data are obtained using stream and rain gauges installed on multiple observation sites.

Step 2: Preliminary analysis

Before the data can be transformed into a phase space, the hydrological processes of the watershed must be analyzed. This can be done using a visual or computational approach. The goal of this preliminary analysis is to understand the relationship between different observation sites and between observed rainfall and water level. The data analysis should answer the following questions:

- Which period before the flood contains relevant information?
- How long does it take for a change at an upstream location to affect the cross-section of interest?
- What is the temporal scope of a flood event?

The answers to those questions will determine the data granularity for the phase space, the attempted lead-times, and the overall window size.

Step 3: Adding deltas

Deltas must be added to the consolidated time series containing data for all observation sites and over the full window size.

Step 4: Creating the augmented phase space

First, the phase space is reconstructed with the added deltas using the time-delay embedding approach for multiple observation sites. After the phase space has been reconstructed, the event characterization function is applied based on a threshold for the cross-section of interest that is based on historic events retrieved during the preliminary analysis. This results in an augmented phase space.

Step 5: Division into training set and testing set

The augmented phase space must be separated into a training and testing set. Both training and testing set should contain tuples from hydrological wet and dry years. To get a better estimate of the generalization error, multiple training and testing sets should be created by splitting the tuples differently into the two sets. The distribution of high to low flow events should remain constant for each set and should reflect the overall distribution of the collected data.

Step 6: Computational experiments for feature selection

This step involves computational experiments to determine which observation sites and deltas should be added and over what extent of the window size. The computational experiments can follow the examples of this study where different combinations of variables are added to the phase space and the resulting model performance is observed using Precision and Recall. Which variables are added and removed from the phase space should be guided using knowledge of the hydrological domain to ensure that the final set of variables chosen for the prediction task reflects the underlying hydrological processes. Additionally, the experiments should be conducted using multiple classification algorithms or ensembles because, according to the 'no free lunch theorem', the same classifier performs differently on different data sets and different classifiers perform differently on the same data set. The goal of this step is to obtain a data set that can be used to train the inducers while optimizing model performance.

Step 7: Applying imbalanced data techniques (optional)

If it is necessary to further boost Recall results for the flood prediction, imbalanced data techniques such as SMOTE and Tomek links can be applied. As this generally comes with a decrease of Precision, the decision to apply an imbalanced data technique should be considered carefully.

Step 8: Classifier/ensemble selection

Based on the results of the computational experiments, one or multiple classifiers combined into an ensemble must be chosen for the prediction task. This and previous studies have shown that classifier ensembles allow for more robust results. Although for some data sets and lead-times used during this study, single classifiers were able to achieve better prediction results than the ensemble, using ensembles showed

overall higher and more stable results across all lead-times and data sets. Consequently, the recommendation for this framework is to use a classifier ensemble for the prediction task.

The results of the experiments conducted show that data preprocessing techniques which transform the original data sets can improve performance of the models developed by training classification algorithms on the corresponding phase spaces. When adding derivatives, it is essential to choose the most informative ones and to avoid the others that introduce noise. Another reason to select the most important variables is that additional variables increase the size of the data sets and, hence, increase training and prediction time.

The results have shown that in the context of hydrological modeling, the importance of a variable depends highly on the type of information that is measured, as well as the location of the sensors in respect to the cross-section of interest, the lead-time of the prediction, and the position within the prediction window. For the Spring Creek, Ontario, Canada, the results indicate that the changes in water level provide more information, resulting in an increase in model performance, compared to changes in rainfall. Based on the outcome of the applied filter methods for variable selection, water level changes carry more information at the cross-section of interest than at upstream locations. This is supported by the knowledge from the hydrology domain. The experiments showed that adding variables over only part of the prediction window size has the potential to increase model performance as well. The reasoning behind this lies in the characteristic of the underlying processes leading to the hydrological event. Changes at an upstream location such as heavy rainfall or an increase in water level will affect downstream locations with a certain delay. As a result, data collected from upstream locations will likely not contain important information for time stamps too close to the actual event. Another factor to consider is the time frame prior the actual hydrological event since the corresponding variable may carry important information. In the case of the Spring Creek, flash floods occur within three hours a heavy rainfall event, with the greatest changes in rainfall and water level usually occurring even closer to the flood. This was also reflected in the experiment results that showed that adding changes in water level only over the first or first two hours of the data set shows better results than adding the deltas over the full window size. This is especially the case for longer prediction lead-times of 45 and 60 minutes and allows mitigating the drop in performance that is visible for these lead-times when deltas are added over the full window size.

Experiments were conducted both using an ensemble of classifiers and single classifiers. As expected, the ensemble showed more robust results while the results of single classifiers showed a higher spread between the results for the different lead-times and data sets. How the single classifiers reacted to the added deltas also varied from one to the other. SimpleCart, JRip, and J48 showed the highest improvements after adding

the delta variables for both Precision and Recall while Random Forest showed the most consistent results across all experiments. Precision mostly stayed at the same high level but Recall results improved with added deltas. RepTree and NBTree initially showed a drop in Precision and Recall when deltas were added for longer lead-times. This drop was mitigated by adding the deltas only over the first and first two hours. Ridor showed the most inconsistent results across all experiments with the highest spread. There was no obvious pattern visible and results varied highly from one lead-time to another or across the different data sets.

Overall, the addition of water level deltas or in combination with rainfall deltas showed an improvement of about 5% to 9% in the prediction of both floods and low-flow events. Applying Tomek links SMOTE overall increased the model performance of floods at the cost of an increase of misclassification of low-flow events.

REFERENCES

Abarbanel, H. (2012). *Analysis of observed chaotic data*. Springer Science & Business Media.

Almuallim, H. and Dietterich, T.G. (1991). Learning with many irrelevant features, In: *Ninth National Conference on Artificial Intelligence*, MIT Press, pp. 547-552.

Amaldi, E. and Kann, V. (1998) On the approximation of minimizing non zero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209, pp. 237–260.

AMS (2000). *Flash Flood*. [online] Available at: http://glossary.ametsoc.org/wiki/Flash_flood [Accessed 11 November 2017].

Aqil, M., Kita, I., Yano, A., and Nishiyama, S. (2007). Neural networks for real time catchment flow modeling and prediction. *Water Resources Management*, 21(10), pp.1781-1796.

ASCE Task Committee on Application of ANNs in Hydrology (2000 a). Artificial neural networks in hydrology. I: Preliminary concepts. *Journal of Hydrologic Engineering*, 5(2), pp. 115-123.

ASCE Task Committee on Application of ANNs in Hydrology (2000 b). Artificial neural networks in hydrology. II: Hydrologic Applications. *Journal of Hydrologic Engineering*, 5(2), pp. 124-137.

Batista, G.E., Prati, R.C. and Monard, M.C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, 6(1), pp.20-29.

Bergström, S. (1976). Development and application of a conceptual runoff model for Scandinavian catchments. University of Lund.

Bi, J., Bennett, K.P., Embrechts, M., Breneman, C.M., Song, M. (2003). Dimensionality Reduction via Sparse Support Vector Machines. *Journal of Machine Learning Research*, 3, pp. 1229-1243.

Bischl, B. (2018). *Mlr v2.12.1. Machine Learning in R.* [online] Available at: https://www.rdocumentation.org/packages/mlr/versions/2.12.1 [Accessed 21 May 2018].

Blöschl, G. and Sivapalan, M. (1995). Scale issues in hydrological modelling: a review. *Hydrological Processes*, 9(3-4), pp. 251-290.

Blum, A.L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97, pp. 245-271.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), pp. 5–32.

Breiman, L., Friedman J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and regression trees*. Belmont: Wadsworth International Group.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, pp. 321-357.

Chen, J. and Adams, B.J. (2006). Integration of artificial neural networks with conceptual models in rainfall-runoff modeling. *Journal of Hydrology*, 318(1), pp. 232-249.

Ciarapica, L. and Todini, E. (2002). TOPKAPI: A model for the representation of the rainfall-runoff process at different scales. *Hydrological Processes*, 16(2), pp. 207-229.

Corzo Perez, G.A. (2009). Hybrid models for hydrological forecasting: Integration of data-driven and conceptual modelling techniques. UNESCO-IHE, Institute for Water Education.

Crawford, N. H. and Linsley, R. S. (1966). *Digital Simulation in Hydrology: The Stanford Watershed Model IV*. Technical Report No, 39, Paolo Alto: Department of Civil Engineering, Stanford University.

CRED, (2016). *CRED Crunch. Disaster Data: A Balanced Perspective*. [online] Available at: cred.be/sites/default/files/CredCrunch41.pdf [Accessed 15 October 2017].

Damle, C. and Yalcin, A. (2007). Flood prediction using Time Series Data Mining, *Journal of Hydrology*, 333(2-4), pp. 305-316.

Dietterich T.G. (2000). Ensemble Methods in Machine Learning. In: *Multiple Classifier Systems*. MCS 2000. Lecture Notes in Computer Science, vol 1857. Berlin, Heidelberg: Springer, pp. 1-15.

Eagleson, P.S. (1972). Dynamics of flood frequency. *Water Resources Research*, 8(4), pp. 878-898.

Erechtchoukova, M.G. and Khaiter, P.A. (2017). The effect of data granularity on prediction of extreme hydrological events in highly urbanized watersheds: A supervised classification approach. *Environmental Modelling & Software*, 96, pp. 232-238.

Erechtchoukova, M.G., Khaiter, P.A. and Saffarpour, S. (2016). Short-Term Predictions of Hydrological Events on an Urbanized Watershed Using Supervised Classification, *Water Resource Management*, 30, pp. 4329-4343.

FICO, (2018). *FICO Falcon Fraud Manager*. [online] http://www.fico.com/en/products/fico-falcon-fraud-manager [Accessed 21 January 2018].

Furquim, G., Neto, F., Pessin, G., Ueyama, J., De Albuquerque, J. P., Clara, M., Mendiondo, E. M., De Souza, V. C. B., De Souza, P., Dimitrova, D. and Braun, T. (2014). Combining Wireless Sensor Networks and Machine Learning for Flash Flood Nowcasting. In: *28th International Conference on Advanced Information Networking and Applications Workshops*, Victoria, BC, 13-16 May 2014, IEEE, pp. 67-72.

Gaines, B.R. and Compton, P. (1995). Induction of ripple-down rules applied to modeling large databases. *Journal of Intelligent Information Systems*, 5(3), pp. 211-228.

Galka, A. (2000). Topics in nonlinear time series analysis: with implications for EEG analysis. Singapore: World Scientific.

Genuer, R., Poggi, J. and Tuleau-Malot, C. (2010). Variable selection using random forests, *Pattern Recognition Letters*, 31(14), pp. 2225-2236.

German, S., Bienenstock, E. and Doursat, R. (1992). Neural networks and the bias/variance dilemma, *Neural Computation*, 4, pp. 1-48.

Gmail, (2015). *The mail you want, not the spam you don't. Official Gmail Blog*. [online] Available at: https://gmail.googleblog.com/2015/07/the-mail-you-want-not-spam-you-dont.html [Accessed 21 January 2018]

Gore, S. and Govindaraju, V. (2013). Feature Selection using Cooperative Game Theory and Relief Algorithm. *Proceedings of 8th International Conference on Knowledge, Information and Creativity Support Systems*, Krakow, Poland, pp. 114-125

Guyon, I. and Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3, pp. 1157-1182

Habib, E., Krajewski, W.F. and Kruger, A. (2001). Sampling errors of tipping-bucket rain gauge measurements. *Journal of Hydrologic Engineering*, 6(2), pp.159-166.

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P. and Witten I.H. (2009) The Weka data mining software: an update. *SIGKDD Explor* 11(1)

Han, D., Cluckie, I.D., Karbassioun, D. Lawry, J. and Krauskopf, B. (2002). River flow modelling using fuzzy decision trees. *Water Resource Management*. 16. pp. 431-445.

Han, J., Kamber, M. and Pei, J. (2011) *Data mining: Concepts and techniques*. 3rd ed. Amsterdam: Morgan Kaufmann Publishers In.

Hansen, L.K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), pp. 993-1000.

Hapuarachchi, H.A.P., Wang, Q.J. and Pagano, T.C. (2011). A review of advances in flash flood forecasting. *Hydrological Processes*, 25, pp. 2771–2784.

Hart, P. (1968). The condensed nearest neighbor rule (Corresp.). *IEEE Transactions on Information Theory*, 14(3), pp. 515-516.

Hu, T.S., Lam, K.C., and Ng, S.T. (2001). River flow time series prediction with a range-dependent neural network. *Hydrological Sciences Journal*, 46(5), pp. 729-745.

Humphrey, G.B., Gibbs, M.S., Dandy, G.C. and Maier, H.R. (2016). A hybrid approach to monthly streamflow forecasting: Integrating hydrological model outputs into a Bayesian artificial neural network. *Journal of Hydrology*, 540, pp. 623-640.

John, G.H., Kohavi, R. and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In: *Proceedings of the 11th International Conference on Machine Learning (ICML-94)*, New Brunswick, NJ, USA, pp. 121–129.

Jonkman, S.N. (2005). Global perspectives on loss of human life caused by floods. *Nat. Hazards*, 34, pp. 151–175.

Kelsch, M. (2001). Hydrometeorological characteristics of flash floods. In: *Gruntfest E., Handmer J. (eds) Coping with Flash Floods. NATO Science Series (Series 2. Environmental Security)*, 77, Dordrecht: Springer, pp. 181–193.

Kira, K. and L.A. Rendell (1992). The Feature Selection Problem: Traditional Methods and a New Algorithm. *Proceedings of the Tenth National Conference on Artificial Intelligence*, Menlo Park: AAAI Press/ The MIT Press, pp. 129-134.

Kohavi, R. (1996). Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. *KDD*, 96, pp. 202-207.

Kohavi, R. and John, G.H. (1997). Wrappers for feature selection. *Artificial Intelligence*, 97(1-2), pp. 273-324.

Koutanaei, F.N., Sajedi, H. and Khanbabaei, M. (2015). A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring. *Journal of Retailing and Consumer Services*, 27, pp. 11-23.

Kubat, M. and Matwin, S. (1997). Addressing the curse of imbalanced training sets: one-sided selection. *ICML*, 97, pp. 179-186.

Kuhn, M., (2017). *Data Splitting*. [online] Available at: http://topepo.github.io/caret/data-splitting.html [Accessed 18 March 2018].

Langley, P., Iba, W. and Thompson, K. (1992). An Analysis of Bayesian Classifier. In: *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 223-228.

Li, C., Bai, Y. and Zeng, B. (2016). Deep feature learning architectures for daily reservoir inflow forecasting. *Water Resource Management*, 30(14), pp. 5145-5161.

Lin, T., Kaminski, N., and Bar-Joseph, Z. (2008). Alignment and classification of time series gene expression in clinical studies. *Bioinformatics*. 24(13), pp. 147–155.

Liu, Z, Martina, M. L.V. and Todini, E. (2005). Flood forecasting using a fully distributed model: application of the TOPKAPI model to the Upper Xixian Catchment. *Hydrology and Earth System Sciences Discussions, European Geosciences Union*, 9 (4), pp. 347-364.

Maier, H.R. and Dandy, G.C. (2000). Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling & Software* 15, pp. 101–124.

Maier, H.R., Jain, A., Dandy, G.C. and Sudheer, K.P. (2010). Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. *Environmental Modelling & Software*, 25, pp. 891–909.

Martin, B. (1995). Instance-based learning: nearest neighbour with generalisation. *Computer Science Working Papers*, 95(18), New Zealand: University of Waikato, Department of Computer Science.

Martinez, J. and Fuentes, O. (2005). Using C4.5 as Variable Selection Criterion in Classification Tasks. In: *Proceedings of the Ninth International Conference Artificial Intelligence and Soft Computing*, Benidorm, Spain, pp. 171-176

McCulloch, D. R., Lawry, J. and Cluckie, I.D. (2008). Real-time flood forecasting using updateable linguistic decision trees, In: *IEEE International Conference on Fuzzy Systems 2008*, Hong Kong: IEEE: pp. 1935-1942

MMM Group, (2013) *Etobicoke Creek Hydrology Update*. [online] Available at: https://trca.ca/wp-content/uploads/2016/07/Etobicoke-Creek-Hydrology-_-March-2013_FINAL.pdf [Accessed 29 October 2017].

National Weather Service, (2016). *2016 Flash Flood / River Flood Fatalities, NWS Report. Office of Climate, Water and Weather Services, NWS/NOAA*. [online] Available at: http://www.nws.noaa.gov/om/hazstats/flood16.pdf [Accessed 11 November 2017].

Nayak, P.C., Sudheer, K.P., Rangan, D.P. and Ramasastri, K.S. (2005). Short-term flood forecasting with a neurofuzzy model. *Water Resource Research*, 41(4), pp. 1-16

Plate, E.J. (2002). Flood risk and flood management. *Journal of Hydrology*, 267, pp. 2-11.

Povinelli, R.J. (2001). Identifying temporal patterns for characterization and prediction of financial time series events. In: *Temporal, Spatial, and Spatio-Temporal Data Mining*, Lecture Notes in Computer Science, 2007, Berlin, Heidelberg: Springer, pp. 46-61.

Povinelli, R.J. and Feng, X. (2003). A new temporal pattern identification method for characterization and prediction of complex time series events. *IEEE Transactions on Knowledge and Data Engineering*, 15(2), pp. 339-352.

Pozzolo, A.D., Caelen, O. and Bontempi, G. (2015). *Package 'unbalanced'*. [online] Available at: https://cran.r-project.org/web/packages/unbalanced/unbalanced.pdf [Accessed 6 April 2018]

Quinlan, J.R. (1987). Simplifying decision trees. *International Journal of Human-Computer Studies*, 51(2), pp. 497-510.

Quinlan, J.R. (1993). *C4.5: programs for machine learning*. USA: Morgan Kaufmann Publishers

Reunanen, J. (2003). Overfitting in Making Comparisons Between Variable Selection Methods. *Journal of Machine Learning Research*, 3, pp. 1371-1382.

Rokach, L., and Maimon, O. (2014). *Data Mining with Decision Trees: Theory and Applications*, World Scientific Publishing Co Pte Ltd, Singapore.

Saffarpour, S., Erechtchoukova, M.G., Khaiter, P.A., Chen, S.Y. and Heralall, M. (2015). Short-term prediction of flood events in a small urbanized watershed using multi-year hydrological records. In: *21st International Congress on Modelling and Simulation*, Gold Coast, 2015, pp. 2234-2240.

Savic, D.A., Walters G.A. and Davidson J.W. (1999). A genetic programming approach to rainfall-runoff modelling. *Water Resource Management*, 13, pp. 219-231.

Sebastiani, F. (2002). Machine learning in automated text categorization, *ACM Computing Surveys*, 34(1), pp. 1–47.

Segretier, W., Clergue, M., Collard, M. and Izquierdo, L. (2012). An evolutionary data mining approach on hydrological data with classifier juries. In: *IEEE World Congress on Computational Intelligence*, Brisbane, 2012. IEEE. pp. 844-851.

Segretier, W., Collard, M. and Clergue, M. (2013). Evolutionary predictive modelling for flash floods. In: *Congress on Evolutionary Computation*. Cancun, 2013. IEEE. pp. 844-851.

Solomatine, D.P., and Dulal, K.N. (2003). Model trees as an alternative to neural networks in rainfall-runoff modelling. *Hydrological Sciences Journal*, 48(3), pp. 399-411.

Swets, J.A. (1988). Measuring the accuracy of diagnostic systems. *Science*, 240(4857), pp.1285-1293.

Takens, F. (1980). Detecting Strange Attractors in Turbulence, *Proceedings Dynamical Systems and Turbulence*, pp. 366-381.

Taormina, R. and Chau, K. (2015). Data-driven input variable selection for rainfall–runoff modeling using binary-coded particle swarm optimization and Extreme Learning Machines. *Journal of Hydrology*, 529, pp. 1617-1632

The R Foundation, (2018). *The R Project for Statistical Computing*. [online] Available at: https://www.r-project.org/ [Accessed 6 April 2018]

Todini, E. (1995). New trends in modelling soil processes from hillslope to GCM scales. In: *The Role of Water and the Hydrological Cycle in gGlobal Change*, NATO ASI Series (Series I: Global Environmental Change), 31, Berlin, Heidelberg: Springer, pp. 317-347.

Tokar, A.S. and Johnson, P.A. (1999). Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, 4(3), pp. 232-239.

Tomek, I. (1976). Two Modifications of CNN. In: *IEEE Transactions on Systems Man and Communications SMC-6*, pp. 769-772.

Toth, E., Brath, A. and Montanari, A. (2000). Comparison of short-term rainfall prediction models for real-time flood forecasting. *Journal of Hydrology*, 239(1), pp. 132-147.

TRCA (2006) *Etobicoke-Mimico watersheds coalition briefing book*. [online] Available at: http://www.trca.on.ca/dotAsset/159240.pdf [Accessed 14 April 2018].

TRCA (2018 a) *TRCA Monitoring Locations*. [online] Available at: http://www.trcagauging.ca/xcreports/SystemData/livemap.asp?net=Precipitation [Accessed 29 October 2017].

TRCA, (2018 b). *About TRCA*. [online] Available at: https://trca.ca/about/ [Accessed 26 March 2018].

United Nations (2004). *Living with Risk - A global review of disaster reduction initiatives*. [online] Available at: https://www.unisdr.org/files/657_lwr1.pdf [Accessed 16 October 2017].

Vieux, B.E., Cui, Z. and Gaur, A. (2004). Evaluation of a physics-based distributed hydrologic model for flood forecasting, *Journal of Hydrology*, 298 (1-4), pp. 155-177.

Weka (2017 a). *Class SimpleCart*. [online] Available at: http://weka.sourceforge.net/doc.packages/simpleCART/weka/classifiers/trees/SimpleCart.html [Accessed 13 December 2017].

Weka (2017 b). *Class Ridor*. [online] Available at: http://weka.sourceforge.net/doc.packages/ridor/weka/classifiers/rules/Ridor.html [Accessed 12 January 2018].

Weka (2017 c). *Class JRip*. [online] Available at: http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/JRip.html [Accessed 12 January 2018]

Weka, (2016 a). *Class J48*. [online] Available at: http://weka.sourceforge.net/doc.stable-3-8/weka/classifiers/trees/J48.html [Accessed 13 December 2017].

Weka, (2016 b). *Class Random Forest*. [online] Available at: http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/Random Forest.html [Accessed 13 December 2017].

Whigham, P.A. and Crapper, P.F. (1999). Time series modelling using genetic programming: An application to rainfall-runoff models. *Advances in Genetic Programming*, 3(5), pp.89-104.

Witten, I.H., Frank, E. (2000). *Data mining: practical machine learning tools and techniques with java implementations*. Morgan Kaufmann, San Mateo

Wolpert, D.H. (1996). The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7), pp. 1341-1390.

Young, P. (2003). Top-down and data-based mechanistic modelling of rainfall–flow dynamics at the catchment scale. *Hydrological Processes*, 17(11), pp. 2195-2217.

APPENDICES

Appendix A: Baseline Ensemble Experiments Result Table

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Baseline	Ensemble	15 mins	0.96	0.95	0.95	0.93	0.97	0.94	0.90	0.95	0.94	0.89
		30 mins	0.90	0.85	0.90	0.90	0.90	0.87	0.86	0.89	0.88	0.81
		45 mins	0.90	0.77	0.85	0.89	0.86	0.78	0.85	0.79	0.84	0.79
		60 mins	0.87	0.75	0.81	0.78	0.80	0.79	0.80	0.80	0.89	0.67

Appendix B: Baseline Classifier Experiments Results Table

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Baseline	J48	15 mins	0.90	0.95	0.94	0.93	0.95	0.91	0.90	0.89	0.90	0.89
		30 mins	0.86	0.80	0.84	0.89	0.85	0.82	0.84	0.85	0.89	0.79
		45 mins	0.85	0.78	0.78	0.81	0.85	0.80	0.81	0.84	0.81	0.76
		60 mins	0.76	0.69	0.80	0.77	0.73	0.69	0.74	0.76	0.76	0.63
	JRip	15 mins	0.92	0.95	0.93	0.92	0.92	0.93	0.89	0.97	0.89	0.93
		30 mins	0.83	0.89	0.82	0.92	0.84	0.88	0.84	0.89	0.82	0.87
		45 mins	0.81	0.84	0.82	0.90	0.73	0.84	0.68	0.83	0.76	0.79
		60 mins	0.80	0.78	0.73	0.83	0.69	0.86	0.68	0.81	0.83	0.74
	NBTree	15 mins	0.90	0.88	0.95	0.90	0.92	0.92	0.91	0.93	0.90	0.92
		30 mins	0.85	0.86	0.86	0.85	0.84	0.81	0.90	0.87	0.84	0.78
		45 mins	0.80	0.74	0.80	0.73	0.80	0.80	0.77	0.72	0.81	0.72
		60 mins	0.82	0.70	0.75	0.78	0.81	0.75	0.80	0.81	0.83	0.71
	RandomForest	15 mins	0.96	0.93	0.96	0.91	0.95	0.94	0.93	0.92	0.92	0.87
		30 mins	0.92	0.81	0.91	0.87	0.92	0.86	0.91	0.84	0.92	0.80
		45 mins	0.92	0.72	0.94	0.84	0.92	0.80	0.94	0.81	0.92	0.72
		60 mins	0.98	0.71	0.94	0.79	0.92	0.77	0.93	0.77	0.92	0.72
	REPTree	15 mins	0.94	0.92	0.92	0.92	0.95	0.90	0.90	0.93	0.91	0.89
		30 mins	0.86	0.83	0.81	0.81	0.88	0.87	0.85	0.81	0.86	0.80
		45 mins	0.84	0.74	0.86	0.85	0.81	0.65	0.80	0.77	0.82	0.78
		60 mins	0.79	0.75	0.75	0.72	0.83	0.66	0.87	0.69	0.81	0.68
	Ridor	15 mins	0.96	0.92	0.89	0.92	0.92	0.88	0.89	0.95	0.96	0.78
		30 mins	0.89	0.69	0.90	0.87	0.80	0.88	0.80	0.93	0.89	0.75
		45 mins	0.88	0.71	0.76	0.90	0.79	0.81	0.86	0.59	0.88	0.73
		60 mins	0.79	0.69	0.80	0.69	0.69	0.74	0.67	0.83	0.88	0.59
	SimpleCart	15 mins	0.93	0.93	0.95	0.93	0.97	0.92	0.89	0.92	0.92	0.90
		30 mins	0.91	0.80	0.88	0.83	0.89	0.84	0.86	0.86	0.89	0.80
		45 mins	0.85	0.75	0.79	0.83	0.80	0.75	0.77	0.78	0.83	0.77
		60 mins	0.81	0.69	0.77	0.74	0.74	0.80	0.78	0.78	0.80	0.66

Appendix C: Full Window Size Delta Ensemble Experiments Result Table

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
All Deltas	Ensemble	15 mins	0.97	0.92	0.94	0.93	0.95	0.95	0.92	0.94	0.94	0.92
		30 mins	0.89	0.86	0.90	0.89	0.89	0.90	0.90	0.90	0.93	0.86
		45 mins	0.89	0.78	0.90	0.87	0.89	0.78	0.88	0.80	0.86	0.80
		60 mins	0.92	0.72	0.85	0.76	0.80	0.75	0.89	0.78	0.82	0.71
Rain Deltas	Ensemble	15 mins	0.94	0.95	0.96	0.93	0.96	0.94	0.91	0.93	0.93	0.92
		30 mins	0.87	0.86	0.88	0.87	0.92	0.87	0.87	0.84	0.89	0.80
		45 mins	0.88	0.75	0.83	0.86	0.84	0.77	0.87	0.77	0.84	0.79
		60 mins	0.83	0.75	0.81	0.78	0.82	0.75	0.81	0.78	0.84	0.76
Water Level Deltas	Ensemble	15 mins	0.93	0.93	0.94	0.93	0.96	0.94	0.93	0.93	0.93	0.92
		30 mins	0.92	0.86	0.89	0.90	0.89	0.90	0.87	0.86	0.93	0.84
		45 mins	0.90	0.78	0.90	0.87	0.87	0.81	0.84	0.84	0.86	0.80
		60 mins	0.86	0.74	0.86	0.81	0.82	0.79	0.88	0.80	0.87	0.75

Appendix D: Full Window Size Delta Classifier Experiments Result Table

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
All Deltas	J48	15 mins	0.96	0.94	0.88	0.93	0.95	0.93	0.91	0.91	0.92	0.93
		30 mins	0.89	0.84	0.86	0.86	0.89	0.89	0.84	0.87	0.90	0.80
		45 mins	0.83	0.75	0.79	0.82	0.83	0.83	0.82	0.73	0.80	0.79
		60 mins	0.76	0.70	0.74	0.78	0.71	0.74	0.74	0.75	0.80	0.66
	JRip	15 mins	0.95	0.97	0.92	0.93	0.89	0.96	0.89	0.94	0.92	0.92
		30 mins	0.86	0.90	0.86	0.88	0.84	0.91	0.82	0.92	0.80	0.91
		45 mins	0.77	0.81	0.84	0.86	0.73	0.81	0.78	0.83	0.78	0.78
		60 mins	0.80	0.76	0.77	0.80	0.70	0.77	0.71	0.82	0.74	0.77
	NBTree	15 mins	0.94	0.92	0.93	0.91	0.92	0.91	0.88	0.91	0.91	0.91
		30 mins	0.88	0.82	0.88	0.86	0.85	0.83	0.86	0.75	0.88	0.80
		45 mins	0.82	0.83	0.84	0.78	0.82	0.81	0.81	0.72	0.80	0.72
		60 mins	0.83	0.72	0.82	0.71	0.72	0.75	0.78	0.72	0.80	0.78
	RandomForest	15 mins	0.98	0.92	0.96	0.89	0.96	0.93	0.94	0.90	0.95	0.84
		30 mins	0.96	0.81	0.92	0.88	0.94	0.86	0.95	0.84	0.92	0.81
		45 mins	0.97	0.75	0.93	0.83	0.93	0.79	0.94	0.80	0.94	0.78
		60 mins	0.98	0.70	0.96	0.80	0.95	0.72	0.95	0.74	0.95	0.73
	REPTree	15 mins	0.90	0.93	0.94	0.90	0.91	0.95	0.92	0.95	0.90	0.92
		30 mins	0.86	0.83	0.88	0.89	0.90	0.83	0.86	0.84	0.85	0.83
		45 mins	0.79	0.74	0.85	0.82	0.85	0.80	0.83	0.77	0.82	0.71
		60 mins	0.84	0.74	0.84	0.72	0.79	0.64	0.83	0.69	0.77	0.66
	Ridor	15 mins	0.99	0.89	0.92	0.95	0.94	0.87	0.94	0.93	0.97	0.90
		30 mins	0.87	0.87	0.90	0.87	0.78	0.93	0.90	0.86	0.94	0.81
		45 mins	0.93	0.63	0.89	0.83	0.92	0.68	0.88	0.70	0.86	0.71
		60 mins	0.90	0.59	0.88	0.70	0.80	0.68	0.85	0.69	0.80	0.63
	SimpleCart	15 mins	0.96	0.90	0.93	0.93	0.95	0.93	0.88	0.92	0.92	0.90
		30 mins	0.89	0.85	0.88	0.87	0.87	0.88	0.87	0.87	0.88	0.83
		45 mins	0.85	0.80	0.84	0.85	0.78	0.80	0.85	0.75	0.81	0.79
		60 mins	0.79	0.71	0.78	0.74	0.76	0.75	0.81	0.78	0.78	0.69

Rain Deltas	J48	15 mins	0.92	0.90	0.89	0.89	0.90	0.91	0.90	0.89	0.88	0.93
		30 mins	0.87	0.80	0.86	0.84	0.83	0.82	0.84	0.80	0.88	0.80
		45 mins	0.79	0.73	0.78	0.76	0.77	0.80	0.81	0.74	0.79	0.75
		60 mins	0.76	0.67	0.78	0.77	0.75	0.76	0.71	0.76	0.77	0.65
	JRip	15 mins	0.92	0.96	0.94	0.93	0.91	0.95	0.90	0.96	0.92	0.92
		30 mins	0.81	0.87	0.84	0.92	0.80	0.89	0.80	0.91	0.85	0.80
		45 mins	0.77	0.85	0.79	0.87	0.78	0.84	0.73	0.83	0.78	0.82
		60 mins	0.79	0.80	0.77	0.83	0.76	0.79	0.68	0.81	0.77	0.76
	NBTree	15 mins	0.93	0.89	0.95	0.92	0.91	0.92	0.89	0.93	0.91	0.88
		30 mins	0.84	0.83	0.83	0.79	0.85	0.81	0.86	0.82	0.80	0.79
		45 mins	0.85	0.75	0.79	0.82	0.78	0.75	0.79	0.79	0.78	0.75
		60 mins	0.81	0.70	0.78	0.71	0.74	0.69	0.72	0.71	0.78	0.74
	RandomForest	15 mins	0.95	0.93	0.95	0.91	0.94	0.92	0.92	0.92	0.93	0.85
		30 mins	0.94	0.80	0.90	0.87	0.93	0.84	0.91	0.84	0.90	0.80
		45 mins	0.98	0.74	0.93	0.83	0.92	0.78	0.93	0.80	0.93	0.72
		60 mins	1.00	0.71	0.96	0.77	0.91	0.75	0.95	0.76	0.92	0.70
	REPTree	15 mins	0.93	0.93	0.93	0.84	0.95	0.90	0.90	0.93	0.90	0.88
		30 mins	0.86	0.83	0.82	0.81	0.88	0.87	0.85	0.81	0.84	0.81
		45 mins	0.84	0.74	0.84	0.76	0.81	0.65	0.80	0.78	0.81	0.73
		60 mins	0.75	0.71	0.73	0.76	0.82	0.67	0.86	0.69	0.78	0.68
	Ridor	15 mins	0.90	0.93	0.95	0.91	0.96	0.90	0.92	0.92	0.94	0.86
		30 mins	0.83	0.83	0.88	0.78	0.96	0.76	0.91	0.75	0.81	0.84
		45 mins	0.91	0.64	0.70	0.89	0.76	0.72	0.81	0.75	0.91	0.57
		60 mins	0.78	0.74	0.80	0.69	0.78	0.77	0.76	0.75	0.79	0.72
	SimpleCart	15 mins	0.93	0.93	0.95	0.93	0.94	0.93	0.91	0.92	0.91	0.92
		30 mins	0.87	0.87	0.87	0.84	0.88	0.82	0.83	0.86	0.89	0.80
		45 mins	0.85	0.74	0.81	0.82	0.78	0.75	0.78	0.78	0.82	0.75
		60 mins	0.83	0.66	0.78	0.72	0.76	0.78	0.78	0.74	0.79	0.71

Water Level Deltas	J48	15 mins	0.92	0.94	0.90	0.93	0.91	0.92	0.91	0.92	0.91	0.92
		30 mins	0.88	0.83	0.86	0.87	0.85	0.89	0.85	0.83	0.88	0.83
		45 mins	0.85	0.78	0.79	0.81	0.81	0.84	0.80	0.81	0.86	0.76
		60 mins	0.76	0.70	0.78	0.79	0.74	0.73	0.75	0.74	0.83	0.66
	JRip	15 mins	0.90	0.96	0.95	0.95	0.93	0.93	0.92	0.93	0.94	0.93
		30 mins	0.84	0.86	0.84	0.90	0.83	0.90	0.85	0.92	0.85	0.86
		45 mins	0.82	0.82	0.84	0.86	0.77	0.86	0.79	0.86	0.74	0.84
		60 mins	0.73	0.80	0.70	0.87	0.78	0.84	0.75	0.84	0.76	0.81
	NBTree	15 mins	0.95	0.91	0.92	0.91	0.92	0.93	0.88	0.90	0.91	0.90
		30 mins	0.82	0.78	0.79	0.81	0.85	0.86	0.86	0.84	0.88	0.84
		45 mins	0.86	0.84	0.79	0.80	0.79	0.81	0.89	0.76	0.84	0.74
		60 mins	0.82	0.71	0.87	0.74	0.73	0.74	0.81	0.74	0.82	0.76
	RandomForest	15 mins	0.96	0.92	0.96	0.92	0.95	0.93	0.93	0.92	0.94	0.87
		30 mins	0.97	0.84	0.94	0.88	0.96	0.89	0.93	0.85	0.93	0.81
		45 mins	0.95	0.75	0.93	0.84	0.95	0.81	0.93	0.80	0.94	0.77
		60 mins	0.94	0.70	0.94	0.81	0.94	0.78	0.93	0.77	0.92	0.72
	REPTree	15 mins	0.91	0.92	0.94	0.90	0.92	0.93	0.91	0.94	0.89	0.93
		30 mins	0.86	0.89	0.88	0.89	0.89	0.83	0.86	0.84	0.86	0.82
		45 mins	0.77	0.75	0.84	0.82	0.85	0.80	0.81	0.79	0.79	0.77
		60 mins	0.84	0.74	0.81	0.74	0.76	0.68	0.83	0.71	0.77	0.67
	Ridor	15 mins	0.95	0.93	0.92	0.95	0.96	0.92	0.92	0.94	0.93	0.94
		30 mins	0.94	0.81	0.90	0.87	0.78	0.93	0.90	0.86	0.94	0.81
		45 mins	0.93	0.63	0.89	0.83	0.90	0.72	0.84	0.80	0.87	0.65
		60 mins	0.85	0.62	0.88	0.75	0.80	0.68	0.82	0.71	0.84	0.60
	SimpleCart	15 mins	0.96	0.90	0.93	0.93	0.95	0.93	0.89	0.92	0.92	0.90
		30 mins	0.88	0.87	0.85	0.89	0.87	0.85	0.84	0.90	0.88	0.84
		45 mins	0.85	0.80	0.83	0.86	0.79	0.78	0.84	0.84	0.82	0.78
		60 mins	0.79	0.71	0.79	0.78	0.76	0.75	0.82	0.81	0.80	0.75

Appendix E: Partial Window Size Delta Ensemble Experiments Result Table

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
All Deltas First 2 Hours	Ensemble	15 mins	0.96	0.93	0.94	0.94	0.95	0.95	0.92	0.94	0.92	0.93
		30 mins	0.89	0.89	0.92	0.90	0.92	0.90	0.91	0.88	0.92	0.84
		45 mins	0.90	0.77	0.87	0.86	0.86	0.84	0.87	0.79	0.88	0.81
		60 mins	0.90	0.79	0.85	0.79	0.81	0.83	0.86	0.78	0.86	0.74
All Deltas First Hour	Ensemble	15 mins	0.96	0.93	0.95	0.94	0.97	0.95	0.93	0.94	0.93	0.94
		30 mins	0.92	0.88	0.90	0.90	0.89	0.90	0.89	0.91	0.93	0.83
		45 mins	0.88	0.83	0.90	0.84	0.84	0.83	0.85	0.84	0.86	0.79
		60 mins	0.90	0.75	0.85	0.77	0.85	0.79	0.88	0.78	0.83	0.72
Rain Deltas First 2 Hours	Ensemble	15 mins	0.93	0.94	0.96	0.93	0.97	0.95	0.92	0.94	0.94	0.91
		30 mins	0.90	0.84	0.89	0.89	0.91	0.88	0.89	0.84	0.88	0.81
		45 mins	0.88	0.73	0.81	0.87	0.85	0.78	0.85	0.80	0.86	0.80
		60 mins	0.88	0.75	0.78	0.78	0.86	0.77	0.85	0.77	0.86	0.75
Rain Deltas First Hour	Ensemble	15 mins	0.94	0.94	0.94	0.93	0.97	0.94	0.91	0.94	0.92	0.92
		30 mins	0.88	0.87	0.89	0.89	0.90	0.89	0.89	0.85	0.88	0.81
		45 mins	0.86	0.80	0.88	0.87	0.86	0.75	0.83	0.78	0.85	0.80
		60 mins	0.85	0.77	0.80	0.77	0.85	0.76	0.84	0.76	0.85	0.67
Water Level Deltas First 2 Hours	Ensemble	15 mins	0.95	0.93	0.95	0.93	0.97	0.93	0.91	0.95	0.93	0.92
		30 mins	0.90	0.89	0.91	0.89	0.89	0.90	0.87	0.87	0.92	0.86
		45 mins	0.90	0.82	0.89	0.85	0.86	0.84	0.86	0.81	0.86	0.82
		60 mins	0.88	0.80	0.86	0.79	0.80	0.81	0.88	0.77	0.88	0.72
Water Level Deltas First Hour	Ensemble	15 mins	0.96	0.93	0.96	0.92	0.96	0.95	0.94	0.95	0.92	0.91
		30 mins	0.93	0.88	0.90	0.90	0.92	0.87	0.91	0.88	0.92	0.85
		45 mins	0.90	0.83	0.85	0.87	0.86	0.79	0.87	0.84	0.81	0.81
		60 mins	0.88	0.74	0.85	0.80	0.83	0.77	0.88	0.77	0.84	0.69

Appendix F: Partial Window Size Delta Classifiers Experiments Result Table

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
All Deltas First 2 Hours	J48	15 mins	0.94	0.94	0.88	0.92	0.96	0.91	0.91	0.91	0.92	0.94
		30 mins	0.86	0.83	0.86	0.88	0.88	0.89	0.85	0.87	0.89	0.80
		45 mins	0.82	0.77	0.80	0.83	0.81	0.83	0.80	0.78	0.83	0.80
		60 mins	0.77	0.68	0.73	0.77	0.72	0.75	0.73	0.78	0.82	0.65
	JRip	15 mins	0.95	0.98	0.93	0.95	0.90	0.96	0.91	0.95	0.89	0.96
		30 mins	0.86	0.87	0.89	0.89	0.85	0.92	0.88	0.90	0.79	0.87
		45 mins	0.81	0.80	0.80	0.87	0.78	0.90	0.77	0.84	0.80	0.80
		60 mins	0.81	0.80	0.82	0.84	0.77	0.82	0.68	0.86	0.78	0.83
	NBTree	15 mins	0.94	0.92	0.91	0.89	0.91	0.91	0.94	0.89	0.89	0.91
		30 mins	0.91	0.87	0.93	0.84	0.87	0.81	0.82	0.82	0.86	0.80
		45 mins	0.84	0.78	0.83	0.81	0.81	0.80	0.81	0.77	0.80	0.72
		60 mins	0.82	0.70	0.82	0.77	0.79	0.74	0.72	0.68	0.80	0.76
	RandomForest	15 mins	0.97	0.93	0.95	0.92	0.96	0.93	0.94	0.92	0.93	0.88
		30 mins	0.95	0.84	0.92	0.88	0.95	0.86	0.95	0.84	0.94	0.81
		45 mins	0.96	0.77	0.93	0.85	0.95	0.77	0.94	0.78	0.93	0.77
		60 mins	0.98	0.71	0.94	0.80	0.93	0.78	0.91	0.74	0.92	0.72
	REPTree	15 mins	0.92	0.93	0.93	0.90	0.92	0.95	0.92	0.95	0.89	0.92
		30 mins	0.86	0.89	0.89	0.89	0.93	0.83	0.87	0.84	0.86	0.82
		45 mins	0.80	0.77	0.83	0.81	0.84	0.80	0.84	0.77	0.82	0.71
		60 mins	0.82	0.77	0.82	0.73	0.75	0.66	0.83	0.69	0.77	0.66
	Ridor	15 mins	0.97	0.93	0.96	0.93	0.94	0.87	0.94	0.93	0.97	0.90
		30 mins	0.86	0.87	0.90	0.87	0.86	0.87	0.90	0.86	0.94	0.80
		45 mins	0.93	0.63	0.89	0.72	0.82	0.80	0.88	0.70	0.86	0.78
		60 mins	0.80	0.75	0.90	0.68	0.76	0.87	0.85	0.69	0.80	0.63
	SimpleCart	15 mins	0.96	0.90	0.97	0.93	0.95	0.93	0.88	0.92	0.92	0.90
		30 mins	0.89	0.85	0.88	0.87	0.87	0.88	0.88	0.86	0.88	0.83
		45 mins	0.85	0.80	0.83	0.85	0.79	0.78	0.85	0.83	0.81	0.79
		60 mins	0.78	0.75	0.80	0.78	0.76	0.75	0.82	0.80	0.83	0.74

All Deltas First Hour	J48	15 mins	0.92	0.95	0.93	0.91	0.96	0.91	0.92	0.90	0.91	0.94
		30 mins	0.87	0.84	0.86	0.90	0.88	0.89	0.85	0.89	0.90	0.81
		45 mins	0.84	0.77	0.78	0.78	0.80	0.84	0.80	0.80	0.80	0.78
		60 mins	0.75	0.68	0.73	0.78	0.72	0.75	0.72	0.77	0.80	0.63
	JRip	15 mins	0.94	0.95	0.96	0.94	0.94	0.93	0.93	0.96	0.93	0.90
		30 mins	0.92	0.90	0.88	0.88	0.81	0.91	0.83	0.90	0.84	0.82
		45 mins	0.80	0.85	0.82	0.89	0.80	0.88	0.74	0.86	0.78	0.77
		60 mins	0.82	0.75	0.78	0.80	0.75	0.84	0.74	0.86	0.72	0.81
	NBTree	15 mins	0.98	0.95	0.95	0.93	0.93	0.92	0.95	0.94	0.89	0.86
		30 mins	0.85	0.80	0.88	0.89	0.87	0.81	0.84	0.89	0.90	0.79
		45 mins	0.86	0.75	0.85	0.79	0.81	0.82	0.78	0.79	0.80	0.75
		60 mins	0.80	0.68	0.84	0.75	0.79	0.76	0.75	0.74	0.81	0.78
	RandomForest	15 mins	0.97	0.93	0.95	0.92	0.95	0.95	0.94	0.92	0.94	0.89
		30 mins	0.94	0.85	0.93	0.89	0.96	0.90	0.92	0.86	0.93	0.81
		45 mins	0.95	0.76	0.92	0.84	0.93	0.82	0.94	0.80	0.91	0.76
		60 mins	0.98	0.71	0.94	0.78	0.94	0.78	0.93	0.77	0.95	0.74
	REPTree	15 mins	0.92	0.93	0.93	0.92	0.92	0.95	0.92	0.95	0.89	0.92
		30 mins	0.85	0.89	0.89	0.89	0.89	0.84	0.88	0.87	0.85	0.82
		45 mins	0.80	0.77	0.87	0.81	0.84	0.80	0.86	0.76	0.83	0.71
		60 mins	0.82	0.78	0.82	0.74	0.80	0.69	0.83	0.69	0.77	0.68
	Ridor	15 mins	0.97	0.93	0.96	0.95	0.96	0.92	0.94	0.93	0.97	0.90
		30 mins	0.89	0.86	0.86	0.88	0.80	0.91	0.90	0.86	0.93	0.81
		45 mins	0.88	0.77	0.85	0.81	0.86	0.78	0.84	0.80	0.86	0.75
		60 mins	0.68	0.77	0.90	0.68	0.70	0.85	0.85	0.73	0.84	0.60
	SimpleCart	15 mins	0.96	0.90	0.96	0.93	0.94	0.95	0.89	0.93	0.91	0.92
		30 mins	0.88	0.87	0.88	0.89	0.87	0.85	0.84	0.90	0.87	0.84
		45 mins	0.82	0.81	0.83	0.86	0.78	0.81	0.85	0.83	0.82	0.80
		60 mins	0.80	0.69	0.79	0.81	0.75	0.79	0.85	0.78	0.78	0.68

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Rain Deltas First 2 Hours	J48	15 mins	0.92	0.93	0.89	0.92	0.91	0.90	0.90	0.89	0.90	0.92
		30 mins	0.89	0.80	0.87	0.89	0.82	0.82	0.83	0.87	0.88	0.80
		45 mins	0.82	0.74	0.80	0.78	0.77	0.81	0.80	0.77	0.78	0.71
		60 mins	0.76	0.67	0.76	0.78	0.74	0.72	0.72	0.77	0.79	0.69
	JRip	15 mins	0.90	0.95	0.94	0.92	0.87	0.95	0.92	0.95	0.92	0.91
		30 mins	0.79	0.90	0.82	0.89	0.80	0.92	0.85	0.92	0.83	0.84
		45 mins	0.80	0.83	0.72	0.89	0.76	0.85	0.70	0.84	0.79	0.80
		60 mins	0.75	0.78	0.75	0.83	0.78	0.80	0.71	0.83	0.75	0.76
	NBTree	15 mins	0.94	0.92	0.96	0.90	0.90	0.92	0.92	0.93	0.90	0.88
		30 mins	0.85	0.75	0.81	0.80	0.82	0.83	0.88	0.80	0.79	0.81
		45 mins	0.80	0.68	0.78	0.74	0.79	0.79	0.77	0.81	0.79	0.73
		60 mins	0.79	0.66	0.72	0.72	0.72	0.66	0.76	0.72	0.76	0.73
	RandomForest	15 mins	0.95	0.93	0.95	0.91	0.93	0.92	0.93	0.91	0.93	0.86
		30 mins	0.96	0.81	0.92	0.86	0.93	0.86	0.93	0.83	0.91	0.81
		45 mins	0.97	0.73	0.95	0.82	0.91	0.80	0.92	0.78	0.89	0.72
		60 mins	0.98	0.70	0.95	0.75	0.91	0.74	0.93	0.75	0.94	0.71
	REPTree	15 mins	0.93	0.93	0.93	0.84	0.95	0.90	0.90	0.93	0.90	0.88
		30 mins	0.86	0.83	0.82	0.81	0.88	0.87	0.85	0.81	0.84	0.81
		45 mins	0.86	0.74	0.82	0.83	0.81	0.65	0.80	0.78	0.82	0.78
		60 mins	0.78	0.71	0.73	0.75	0.82	0.67	0.86	0.68	0.78	0.68
	Ridor	15 mins	0.89	0.96	0.95	0.91	0.95	0.90	0.88	0.96	0.94	0.86
		30 mins	0.89	0.70	0.88	0.78	0.96	0.76	0.91	0.75	0.87	0.78
		45 mins	0.91	0.68	0.70	0.89	0.76	0.72	0.84	0.72	0.85	0.76
		60 mins	0.78	0.74	0.80	0.69	0.79	0.65	0.80	0.72	0.79	0.72
	SimpleCart	15 mins	0.93	0.93	0.95	0.93	0.96	0.93	0.91	0.92	0.91	0.92
		30 mins	0.87	0.87	0.89	0.83	0.88	0.82	0.85	0.85	0.89	0.80
		45 mins	0.85	0.74	0.81	0.82	0.81	0.72	0.78	0.78	0.82	0.75
		60 mins	0.83	0.66	0.78	0.72	0.76	0.78	0.78	0.74	0.79	0.71

Rain Deltas First Hour	J48	15 mins	0.88	0.90	0.92	0.91	0.92	0.90	0.89	0.89	0.90	0.94
		30 mins	0.88	0.80	0.87	0.89	0.83	0.83	0.80	0.86	0.89	0.79
		45 mins	0.82	0.77	0.80	0.81	0.79	0.81	0.78	0.79	0.78	0.76
		60 mins	0.74	0.66	0.77	0.76	0.76	0.71	0.73	0.77	0.78	0.65
	JRip	15 mins	0.93	0.94	0.92	0.95	0.90	0.95	0.88	0.93	0.87	0.92
		30 mins	0.85	0.88	0.82	0.92	0.79	0.89	0.85	0.88	0.81	0.86
		45 mins	0.79	0.81	0.80	0.86	0.74	0.84	0.81	0.81	0.74	0.77
		60 mins	0.74	0.81	0.73	0.84	0.75	0.80	0.73	0.80	0.75	0.75
	NBTree	15 mins	0.92	0.92	0.94	0.90	0.92	0.92	0.89	0.93	0.90	0.91
		30 mins	0.87	0.72	0.81	0.87	0.83	0.80	0.79	0.79	0.83	0.83
		45 mins	0.81	0.77	0.79	0.77	0.81	0.79	0.78	0.79	0.77	0.75
		60 mins	0.79	0.69	0.84	0.74	0.75	0.71	0.75	0.75	0.77	0.74
	RandomForest	15 mins	0.96	0.93	0.94	0.92	0.93	0.92	0.93	0.92	0.94	0.86
		30 mins	0.94	0.81	0.91	0.86	0.91	0.86	0.91	0.83	0.90	0.79
		45 mins	0.95	0.75	0.95	0.83	0.94	0.78	0.95	0.78	0.93	0.73
		60 mins	1.00	0.71	0.96	0.79	0.91	0.76	0.93	0.75	0.92	0.71
	REPTree	15 mins	0.94	0.92	0.92	0.92	0.95	0.90	0.90	0.93	0.90	0.88
		30 mins	0.86	0.83	0.82	0.81	0.88	0.87	0.85	0.81	0.84	0.81
		45 mins	0.84	0.74	0.84	0.81	0.81	0.65	0.79	0.78	0.82	0.78
		60 mins	0.77	0.72	0.75	0.72	0.80	0.67	0.85	0.72	0.79	0.65
	Ridor	15 mins	0.89	0.96	0.95	0.93	0.92	0.88	0.88	0.96	0.93	0.84
		30 mins	0.76	0.93	0.88	0.78	0.84	0.87	0.88	0.82	0.85	0.82
		45 mins	0.77	0.80	0.80	0.85	0.89	0.65	0.75	0.73	0.85	0.76
		60 mins	0.78	0.74	0.80	0.69	0.79	0.65	0.80	0.72	0.88	0.59
	SimpleCart	15 mins	0.93	0.93	0.95	0.93	0.96	0.93	0.90	0.91	0.91	0.92
		30 mins	0.91	0.80	0.89	0.83	0.89	0.84	0.85	0.86	0.89	0.80
		45 mins	0.85	0.75	0.79	0.83	0.80	0.74	0.78	0.78	0.83	0.77
		60 mins	0.83	0.66	0.78	0.72	0.74	0.78	0.81	0.78	0.81	0.66

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Water Level Deltas First 2 Hours	J48	15 mins	0.92	0.95	0.91	0.92	0.93	0.90	0.91	0.92	0.90	0.91
		30 mins	0.86	0.84	0.86	0.87	0.85	0.89	0.86	0.84	0.89	0.83
		45 mins	0.85	0.77	0.77	0.80	0.80	0.83	0.80	0.83	0.86	0.78
		60 mins	0.75	0.71	0.78	0.79	0.75	0.75	0.73	0.76	0.81	0.65
	JRip	15 mins	0.95	0.96	0.96	0.96	0.91	0.94	0.87	0.97	0.93	0.90
		30 mins	0.88	0.87	0.85	0.89	0.83	0.93	0.82	0.87	0.82	0.90
		45 mins	0.82	0.87	0.82	0.88	0.82	0.87	0.78	0.79	0.83	0.83
		60 mins	0.80	0.79	0.77	0.87	0.77	0.80	0.70	0.80	0.80	0.84
	NBTree	15 mins	0.96	0.92	0.90	0.92	0.92	0.91	0.90	0.90	0.92	0.89
		30 mins	0.90	0.81	0.79	0.80	0.81	0.83	0.87	0.80	0.90	0.83
		45 mins	0.86	0.77	0.80	0.82	0.74	0.79	0.83	0.78	0.76	0.79
		60 mins	0.77	0.75	0.83	0.83	0.76	0.76	0.76	0.73	0.79	0.72
	RandomForest	15 mins	0.96	0.95	0.94	0.92	0.96	0.94	0.92	0.92	0.94	0.87
		30 mins	0.94	0.83	0.93	0.89	0.96	0.90	0.91	0.84	0.93	0.82
		45 mins	0.96	0.77	0.94	0.83	0.94	0.82	0.93	0.81	0.92	0.78
		60 mins	0.96	0.73	0.93	0.80	0.93	0.77	0.94	0.77	0.95	0.72
	REPTree	15 mins	0.93	0.92	0.93	0.90	0.92	0.93	0.91	0.94	0.89	0.93
		30 mins	0.86	0.89	0.89	0.89	0.91	0.83	0.87	0.84	0.86	0.82
		45 mins	0.79	0.77	0.85	0.78	0.84	0.80	0.81	0.79	0.79	0.77
		60 mins	0.82	0.77	0.82	0.74	0.76	0.68	0.83	0.71	0.77	0.67
	Ridor	15 mins	0.96	0.93	0.93	0.96	0.96	0.92	0.92	0.94	0.93	0.94
		30 mins	0.86	0.87	0.90	0.87	0.83	0.91	0.90	0.86	0.94	0.80
		45 mins	0.93	0.63	0.90	0.70	0.85	0.78	0.84	0.80	0.84	0.79
		60 mins	0.70	0.77	0.90	0.68	0.74	0.90	0.82	0.71	0.84	0.60
	SimpleCart	15 mins	0.96	0.90	0.96	0.93	0.95	0.93	0.89	0.92	0.92	0.90
		30 mins	0.88	0.87	0.85	0.89	0.87	0.86	0.87	0.86	0.87	0.85
		45 mins	0.85	0.80	0.84	0.86	0.79	0.78	0.84	0.84	0.82	0.78
		60 mins	0.78	0.75	0.79	0.80	0.76	0.75	0.82	0.80	0.81	0.73

Water Level Deltas First Hour	J48	15 mins	0.92	0.95	0.94	0.92	0.93	0.90	0.92	0.90	0.90	0.91
		30 mins	0.87	0.84	0.86	0.87	0.85	0.89	0.84	0.84	0.89	0.84
		45 mins	0.85	0.77	0.79	0.81	0.82	0.83	0.81	0.83	0.82	0.77
		60 mins	0.74	0.69	0.79	0.78	0.75	0.75	0.74	0.75	0.81	0.62
	JRip	15 mins	0.93	0.96	0.96	0.92	0.93	0.93	0.93	0.96	0.92	0.89
		30 mins	0.88	0.87	0.84	0.90	0.82	0.93	0.85	0.89	0.83	0.90
		45 mins	0.84	0.88	0.81	0.89	0.77	0.83	0.80	0.84	0.76	0.81
		60 mins	0.77	0.80	0.78	0.86	0.74	0.83	0.73	0.83	0.76	0.74
	NBTree	15 mins	0.94	0.92	0.94	0.93	0.93	0.93	0.92	0.91	0.91	0.91
		30 mins	0.88	0.82	0.82	0.81	0.85	0.84	0.82	0.83	0.86	0.78
		45 mins	0.82	0.77	0.82	0.86	0.81	0.77	0.80	0.81	0.81	0.69
		60 mins	0.83	0.72	0.78	0.75	0.74	0.79	0.77	0.75	0.78	0.74
	RandomForest	15 mins	0.98	0.94	0.95	0.93	0.95	0.96	0.93	0.92	0.94	0.86
		30 mins	0.95	0.86	0.92	0.90	0.95	0.86	0.92	0.83	0.94	0.83
		45 mins	0.96	0.79	0.95	0.83	0.94	0.82	0.93	0.80	0.92	0.77
		60 mins	0.95	0.72	0.96	0.80	0.93	0.78	0.94	0.78	0.93	0.74
	REPTree	15 mins	0.93	0.92	0.93	0.92	0.92	0.93	0.91	0.94	0.89	0.93
		30 mins	0.85	0.89	0.89	0.89	0.91	0.83	0.89	0.84	0.85	0.82
		45 mins	0.79	0.77	0.84	0.81	0.84	0.80	0.83	0.78	0.80	0.77
		60 mins	0.82	0.78	0.82	0.74	0.83	0.68	0.83	0.71	0.77	0.69
	Ridor	15 mins	0.96	0.93	0.93	0.96	0.96	0.92	0.92	0.94	0.93	0.94
		30 mins	0.89	0.86	0.87	0.87	0.90	0.75	0.90	0.86	0.93	0.81
		45 mins	0.88	0.74	0.82	0.89	0.86	0.78	0.83	0.80	0.78	0.76
		60 mins	0.76	0.78	0.90	0.68	0.79	0.63	0.85	0.73	0.84	0.60
	SimpleCart	15 mins	0.96	0.90	0.96	0.93	0.95	0.95	0.89	0.92	0.92	0.90
		30 mins	0.91	0.85	0.88	0.89	0.87	0.86	0.88	0.87	0.87	0.85
		45 mins	0.85	0.80	0.83	0.86	0.79	0.78	0.84	0.84	0.81	0.78
		60 mins	0.80	0.69	0.79	0.80	0.75	0.79	0.82	0.80	0.78	0.68

Appendix G: Imbalanced Data Experiments Ensemble Results Table

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
All Deltas SMOTE	Ensemble	15 mins	0.98	0.97	0.95	0.95	0.91	0.96	0.90	0.96	0.93	0.94
		30 mins	0.89	0.91	0.87	0.92	0.87	0.91	0.87	0.91	0.86	0.87
		45 mins	0.84	0.84	0.85	0.91	0.81	0.84	0.78	0.83	0.83	0.83
		60 mins	0.84	0.81	0.84	0.82	0.80	0.81	0.79	0.83	0.86	0.75
All Deltas Tomek	Ensemble	15 mins	0.95	0.96	0.93	0.93	0.91	0.95	0.92	0.94	0.92	0.93
		30 mins	0.86	0.92	0.83	0.92	0.88	0.90	0.86	0.91	0.88	0.86
		45 mins	0.84	0.85	0.83	0.87	0.81	0.89	0.82	0.84	0.81	0.85
		60 mins	0.86	0.80	0.79	0.84	0.76	0.81	0.86	0.81	0.83	0.75
All Deltas Tomek and SMOTE	Ensemble	15 mins	0.92	0.98	0.92	0.93	0.91	0.96	0.90	0.96	0.91	0.93
		30 mins	0.85	0.95	0.82	0.92	0.81	0.93	0.83	0.94	0.84	0.87
		45 mins	0.83	0.89	0.79	0.90	0.79	0.93	0.79	0.86	0.78	0.84
		60 mins	0.83	0.79	0.80	0.89	0.69	0.88	0.78	0.83	0.76	0.85
Baseline SMOTE	Ensemble	15 mins	0.94	0.96	0.92	0.94	1.00	1.00	0.88	0.95	0.94	0.92
		30 mins	0.87	0.88	0.81	0.92	0.83	0.88	0.87	0.89	0.84	0.88
		45 mins	0.83	0.83	0.85	0.89	0.85	0.82	0.81	0.83	0.85	0.81
		60 mins	0.86	0.78	0.81	0.84	0.75	0.79	0.79	0.81	0.80	0.79
Baseline Tomek	Ensemble	15 mins	0.94	0.95	0.93	0.92	0.92	0.95	0.90	0.95	0.91	0.87
		30 mins	0.91	0.87	0.86	0.90	0.85	0.90	0.82	0.91	0.87	0.85
		45 mins	0.85	0.77	0.84	0.87	0.78	0.83	0.80	0.80	0.82	0.81
		60 mins	0.87	0.77	0.78	0.85	0.76	0.80	0.77	0.77	0.80	0.71
Baseline Tomek and SMOTE	Ensemble	15 mins	0.92	0.96	0.94	0.93	0.86	0.96	0.90	0.95	0.92	0.91
		30 mins	0.87	0.93	0.83	0.94	0.86	0.93	0.79	0.90	0.81	0.89
		45 mins	0.81	0.84	0.79	0.88	0.77	0.85	0.75	0.86	0.82	0.83
		60 mins	0.78	0.80	0.79	0.86	0.76	0.85	0.79	0.82	0.78	0.73

Appendix H: Imbalanced Data Experiments No Delta Single Classifier Results Table

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Baseline SMOTE	J48	15 mins	0.91	0.95	0.93	0.90	1.00	1.00	0.86	0.92	0.93	0.90
		30 mins	0.86	0.88	0.81	0.88	0.81	0.86	0.84	0.86	0.85	0.83
		45 mins	0.78	0.79	0.77	0.82	0.78	0.80	0.74	0.81	0.78	0.82
		60 mins	0.75	0.74	0.76	0.78	0.68	0.77	0.78	0.81	0.76	0.75
	JRip	15 mins	0.89	0.97	0.91	0.92	1.00	1.00	0.80	0.95	0.92	0.92
		30 mins	0.75	0.89	0.81	0.89	0.85	0.87	0.83	0.92	0.79	0.91
		45 mins	0.79	0.80	0.79	0.86	0.74	0.84	0.73	0.87	0.79	0.82
		60 mins	0.74	0.85	0.72	0.86	0.69	0.85	0.70	0.83	0.74	0.81
	NBTree	15 mins	0.91	0.90	0.95	0.90	1.00	1.00	0.89	0.92	0.91	0.90
		30 mins	0.84	0.84	0.86	0.85	0.80	0.84	0.81	0.83	0.81	0.80
		45 mins	0.78	0.74	0.77	0.81	0.72	0.76	0.80	0.80	0.79	0.74
		60 mins	0.81	0.75	0.73	0.75	0.74	0.72	0.80	0.75	0.72	0.71
	RandomForest	15 mins	0.93	0.92	0.95	0.93	1.00	1.00	0.91	0.93	0.92	0.90
		30 mins	0.90	0.86	0.90	0.89	0.91	0.89	0.87	0.87	0.88	0.82
		45 mins	0.89	0.78	0.91	0.87	0.90	0.83	0.90	0.83	0.88	0.78
		60 mins	0.90	0.75	0.91	0.85	0.88	0.78	0.88	0.83	0.88	0.78
	REPTree	15 mins	0.93	0.96	0.90	0.94	1.00	1.00	0.88	0.96	0.92	0.86
		30 mins	0.83	0.89	0.78	0.92	0.80	0.86	0.84	0.89	0.88	0.82
		45 mins	0.80	0.82	0.73	0.86	0.69	0.80	0.77	0.81	0.82	0.75
		60 mins	0.83	0.69	0.77	0.75	0.70	0.77	0.78	0.75	0.71	0.77
	Ridor	15 mins	0.94	0.93	0.96	0.92	1.00	1.00	0.89	0.95	0.92	0.89
		30 mins	0.86	0.80	0.82	0.92	0.85	0.83	0.91	0.83	0.86	0.78
		45 mins	0.85	0.81	0.86	0.80	0.87	0.74	0.82	0.73	0.83	0.75
		60 mins	0.79	0.74	0.75	0.78	0.77	0.74	0.70	0.83	0.79	0.74
	SimpleCart	15 mins	0.90	0.95	0.92	0.94	1.00	1.00	0.89	0.95	0.90	0.94
		30 mins	0.86	0.88	0.82	0.87	0.77	0.90	0.81	0.87	0.82	0.89
		45 mins	0.77	0.84	0.81	0.90	0.79	0.81	0.74	0.80	0.82	0.74
		60 mins	0.79	0.75	0.76	0.82	0.68	0.77	0.75	0.79	0.71	0.76

Baseline Tomek	J48	15 mins	0.90	0.94	0.92	0.90	0.93	0.92	0.89	0.93	0.86	0.89
		30 mins	0.84	0.84	0.80	0.88	0.86	0.87	0.82	0.89	0.86	0.83
		45 mins	0.78	0.77	0.77	0.81	0.77	0.80	0.75	0.78	0.76	0.79
		60 mins	0.77	0.78	0.72	0.78	0.72	0.73	0.74	0.76	0.74	0.74
	JRip	15 mins	0.90	0.95	0.89	0.93	0.87	0.95	0.87	0.98	0.87	0.95
		30 mins	0.80	0.89	0.79	0.93	0.81	0.92	0.84	0.86	0.89	0.85
		45 mins	0.81	0.83	0.77	0.87	0.71	0.88	0.74	0.81	0.76	0.85
		60 mins	0.71	0.79	0.71	0.86	0.64	0.81	0.65	0.81	0.74	0.80
	NBTree	15 mins	0.93	0.95	0.91	0.92	0.90	0.96	0.90	0.93	0.92	0.87
		30 mins	0.86	0.83	0.88	0.88	0.80	0.86	0.84	0.83	0.82	0.83
		45 mins	0.84	0.76	0.80	0.83	0.76	0.81	0.80	0.79	0.74	0.78
		60 mins	0.76	0.75	0.75	0.78	0.74	0.75	0.75	0.72	0.76	0.79
	RandomForest	15 mins	0.94	0.93	0.93	0.93	0.91	0.95	0.91	0.93	0.93	0.90
		30 mins	0.89	0.81	0.90	0.90	0.89	0.90	0.87	0.86	0.87	0.81
		45 mins	0.91	0.77	0.88	0.87	0.85	0.84	0.87	0.81	0.84	0.77
		60 mins	0.93	0.73	0.88	0.84	0.86	0.81	0.87	0.81	0.87	0.77
	REPTree	15 mins	0.96	0.93	0.90	0.94	0.86	0.95	0.87	0.92	0.90	0.87
		30 mins	0.81	0.83	0.78	0.93	0.82	0.87	0.86	0.83	0.88	0.78
		45 mins	0.78	0.81	0.72	0.86	0.79	0.78	0.74	0.79	0.79	0.75
		60 mins	0.73	0.75	0.69	0.80	0.85	0.72	0.76	0.69	0.77	0.68
	Ridor	15 mins	0.93	0.94	0.89	0.94	0.90	0.95	0.88	0.94	0.88	0.91
		30 mins	0.89	0.80	0.81	0.89	0.85	0.80	0.83	0.84	0.92	0.77
		45 mins	0.78	0.74	0.75	0.87	0.74	0.83	0.76	0.83	0.77	0.84
		60 mins	0.84	0.71	0.78	0.82	0.80	0.74	0.77	0.75	0.75	0.80
	SimpleCart	15 mins	0.95	0.93	0.92	0.93	0.86	0.96	0.91	0.94	0.88	0.93
		30 mins	0.84	0.87	0.83	0.85	0.83	0.86	0.83	0.90	0.85	0.83
		45 mins	0.83	0.76	0.82	0.84	0.72	0.83	0.72	0.80	0.78	0.78
		60 mins	0.76	0.72	0.71	0.83	0.72	0.80	0.73	0.78	0.75	0.70

Baseline Tomek and SMOTE	J48	15 mins	0.90	0.93	0.86	0.91	0.88	0.95	0.87	0.96	0.90	0.90
		30 mins	0.84	0.94	0.82	0.90	0.79	0.87	0.81	0.90	0.80	0.78
		45 mins	0.76	0.80	0.75	0.85	0.73	0.86	0.76	0.83	0.75	0.77
		60 mins	0.72	0.73	0.71	0.83	0.73	0.83	0.71	0.78	0.75	0.69
	JRip	15 mins	0.91	0.96	0.89	0.94	0.84	0.96	0.85	0.96	0.89	0.93
		30 mins	0.77	0.92	0.73	0.94	0.75	0.92	0.79	0.93	0.80	0.92
		45 mins	0.78	0.83	0.71	0.93	0.70	0.89	0.73	0.84	0.78	0.85
		60 mins	0.67	0.84	0.70	0.86	0.71	0.83	0.68	0.84	0.70	0.80
	NBTree	15 mins	0.94	0.95	0.88	0.96	0.86	0.96	0.88	0.90	0.90	0.91
		30 mins	0.85	0.87	0.82	0.87	0.77	0.81	0.79	0.86	0.78	0.80
		45 mins	0.76	0.75	0.74	0.81	0.79	0.83	0.75	0.81	0.77	0.80
		60 mins	0.69	0.73	0.74	0.79	0.69	0.81	0.73	0.75	0.71	0.79
	RandomForest	15 mins	0.92	0.95	0.90	0.93	0.89	0.96	0.89	0.94	0.89	0.92
		30 mins	0.88	0.87	0.87	0.92	0.85	0.92	0.84	0.90	0.83	0.84
		45 mins	0.89	0.80	0.85	0.89	0.84	0.84	0.82	0.83	0.84	0.82
		60 mins	0.87	0.77	0.84	0.88	0.86	0.82	0.82	0.84	0.86	0.79
	REPTree	15 mins	0.89	0.96	0.89	0.94	0.82	0.96	0.87	0.96	0.90	0.92
		30 mins	0.80	0.92	0.76	0.92	0.81	0.90	0.80	0.87	0.81	0.85
		45 mins	0.79	0.80	0.76	0.87	0.74	0.86	0.76	0.85	0.76	0.84
		60 mins	0.77	0.77	0.75	0.77	0.65	0.80	0.65	0.80	0.74	0.74
	Ridor	15 mins	0.87	0.96	0.87	0.95	0.90	0.95	0.90	0.94	0.93	0.90
		30 mins	0.88	0.84	0.82	0.90	0.80	0.91	0.80	0.85	0.82	0.84
		45 mins	0.80	0.80	0.80	0.84	0.69	0.87	0.79	0.78	0.80	0.75
		60 mins	0.79	0.75	0.62	0.92	0.76	0.76	0.72	0.83	0.73	0.81
	SimpleCart	15 mins	0.90	0.95	0.93	0.93	0.86	0.96	0.88	0.96	0.91	0.92
		30 mins	0.85	0.89	0.81	0.89	0.75	0.90	0.76	0.89	0.82	0.92
		45 mins	0.74	0.78	0.73	0.91	0.70	0.83	0.77	0.86	0.73	0.77
		60 mins	0.72	0.80	0.71	0.86	0.65	0.80	0.75	0.80	0.70	0.75

Appendix I: Imbalanced Data Experiments All Delta Single Classifier Results Table

			set1		set2		set3		set4		set5	
Category	Classifier	Lead	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
All Deltas SMOTE	J48	15 mins	0.93	0.93	0.94	0.92	0.90	0.94	0.89	0.90	0.93	0.91
		30 mins	0.91	0.85	0.86	0.87	0.85	0.84	0.83	0.81	0.86	0.87
		45 mins	0.78	0.81	0.82	0.85	0.76	0.77	0.74	0.81	0.75	0.77
		60 mins	0.75	0.68	0.79	0.74	0.73	0.73	0.72	0.78	0.74	0.67
	JRip	15 mins	0.94	0.97	0.91	0.95	0.91	0.96	0.90	0.97	0.89	0.93
		30 mins	0.82	0.92	0.78	0.93	0.79	0.91	0.78	0.93	0.77	0.87
		45 mins	0.78	0.80	0.76	0.89	0.76	0.84	0.70	0.90	0.80	0.86
		60 mins	0.72	0.84	0.74	0.80	0.72	0.83	0.70	0.88	0.78	0.81
	NBTree	15 mins	0.87	0.92	0.91	0.90	0.90	0.89	0.92	0.92	0.91	0.94
		30 mins	0.83	0.83	0.86	0.83	0.86	0.83	0.80	0.77	0.84	0.88
		45 mins	0.83	0.81	0.77	0.87	0.75	0.81	0.79	0.81	0.79	0.77
		60 mins	0.77	0.71	0.79	0.83	0.77	0.78	0.77	0.78	0.75	0.70
	RandomForest	15 mins	0.95	0.93	0.93	0.92	0.93	0.95	0.92	0.92	0.93	0.88
		30 mins	0.93	0.83	0.91	0.92	0.91	0.88	0.89	0.86	0.90	0.85
		45 mins	0.94	0.78	0.88	0.87	0.90	0.83	0.90	0.83	0.91	0.81
		60 mins	0.93	0.77	0.91	0.83	0.90	0.81	0.90	0.81	0.89	0.78
	REPTree	15 mins	0.93	0.99	0.89	0.89	0.91	0.94	0.83	0.99	0.91	0.95
		30 mins	0.86	0.93	0.75	0.90	0.81	0.91	0.82	0.91	0.81	0.85
		45 mins	0.79	0.81	0.86	0.88	0.76	0.83	0.76	0.80	0.77	0.78
		60 mins	0.76	0.77	0.72	0.80	0.75	0.81	0.72	0.81	0.70	0.72
	Ridor	15 mins	0.96	0.93	0.92	0.96	0.92	0.95	0.92	0.91	0.92	0.93
		30 mins	0.92	0.87	0.92	0.88	0.86	0.87	0.88	0.86	0.86	0.86
		45 mins	0.75	0.84	0.89	0.81	0.79	0.89	0.75	0.81	0.85	0.80
		60 mins	0.82	0.80	0.74	0.77	0.76	0.71	0.72	0.82	0.89	0.66
	SimpleCart	15 mins	0.89	0.96	0.94	0.93	0.90	0.96	0.90	0.95	0.93	0.92
		30 mins	0.84	0.87	0.84	0.91	0.83	0.91	0.82	0.93	0.88	0.85
		45 mins	0.80	0.83	0.78	0.89	0.74	0.85	0.74	0.82	0.73	0.78
		60 mins	0.78	0.78	0.76	0.82	0.69	0.79	0.72	0.82	0.75	0.74

All Deltas Tomek	J48	15 mins	0.90	0.93	0.93	0.92	0.92	0.93	0.90	0.92	0.88	0.95
		30 mins	0.87	0.87	0.83	0.89	0.82	0.84	0.83	0.92	0.85	0.81
		45 mins	0.77	0.79	0.78	0.85	0.77	0.83	0.78	0.81	0.79	0.75
		60 mins	0.77	0.76	0.73	0.80	0.66	0.76	0.75	0.79	0.72	0.67
	JRip	15 mins	0.89	0.95	0.91	0.93	0.89	0.96	0.90	0.96	0.89	0.95
		30 mins	0.82	0.87	0.81	0.91	0.83	0.89	0.83	0.91	0.79	0.90
		45 mins	0.79	0.84	0.75	0.86	0.72	0.85	0.76	0.87	0.78	0.85
		60 mins	0.73	0.87	0.75	0.88	0.70	0.83	0.73	0.84	0.69	0.79
	NBTree	15 mins	0.92	0.93	0.94	0.92	0.92	0.94	0.90	0.94	0.89	0.91
		30 mins	0.90	0.87	0.82	0.88	0.76	0.85	0.84	0.86	0.85	0.83
		45 mins	0.81	0.81	0.84	0.79	0.76	0.77	0.79	0.80	0.86	0.77
		60 mins	0.78	0.75	0.78	0.80	0.71	0.81	0.77	0.74	0.77	0.73
	RandomForest	15 mins	0.95	0.93	0.93	0.92	0.89	0.93	0.90	0.92	0.92	0.86
		30 mins	0.92	0.85	0.90	0.89	0.87	0.91	0.89	0.85	0.88	0.82
		45 mins	0.93	0.78	0.89	0.85	0.89	0.83	0.88	0.80	0.88	0.80
		60 mins	0.95	0.75	0.90	0.82	0.91	0.78	0.90	0.80	0.91	0.78
	REPTree	15 mins	0.90	0.98	0.91	0.93	0.89	0.95	0.93	0.93	0.91	0.89
		30 mins	0.83	0.90	0.84	0.89	0.86	0.93	0.82	0.92	0.87	0.84
		45 mins	0.80	0.81	0.79	0.83	0.82	0.78	0.82	0.81	0.75	0.81
		60 mins	0.76	0.77	0.79	0.76	0.74	0.78	0.79	0.74	0.79	0.71
	Ridor	15 mins	0.95	0.94	0.92	0.93	0.89	0.96	0.89	0.96	0.94	0.87
		30 mins	0.87	0.86	0.86	0.89	0.90	0.78	0.89	0.81	0.89	0.78
		45 mins	0.70	0.90	0.84	0.82	0.74	0.87	0.81	0.83	0.72	0.83
		60 mins	0.83	0.72	0.65	0.91	0.71	0.79	0.83	0.75	0.87	0.59
	SimpleCart	15 mins	0.93	0.97	0.92	0.93	0.87	0.95	0.89	0.92	0.92	0.93
		30 mins	0.84	0.93	0.80	0.90	0.83	0.94	0.81	0.89	0.86	0.86
		45 mins	0.85	0.80	0.80	0.86	0.79	0.86	0.81	0.83	0.79	0.81
		60 mins	0.81	0.73	0.76	0.73	0.75	0.84	0.77	0.81	0.77	0.79

All Deltas Tomek and SMOTE	J48	15 mins	0.91	0.94	0.88	0.94	0.87	0.95	0.86	0.95	0.86	0.92
		30 mins	0.85	0.93	0.82	0.90	0.82	0.90	0.82	0.90	0.85	0.84
		45 mins	0.77	0.84	0.75	0.87	0.75	0.84	0.76	0.83	0.80	0.78
		60 mins	0.75	0.77	0.69	0.79	0.65	0.84	0.69	0.83	0.70	0.79
	JRip	15 mins	0.92	0.98	0.90	0.93	0.89	0.96	0.91	0.95	0.92	0.88
		30 mins	0.84	0.93	0.79	0.90	0.76	0.93	0.75	0.94	0.80	0.89
		45 mins	0.74	0.87	0.73	0.92	0.76	0.84	0.72	0.89	0.73	0.84
		60 mins	0.73	0.83	0.73	0.91	0.70	0.87	0.64	0.87	0.73	0.80
	NBTree	15 mins	0.90	0.98	0.90	0.95	0.88	0.94	0.90	0.95	0.87	0.90
		30 mins	0.82	0.83	0.85	0.88	0.81	0.84	0.74	0.90	0.86	0.87
		45 mins	0.81	0.79	0.79	0.83	0.68	0.82	0.79	0.87	0.79	0.80
		60 mins	0.74	0.77	0.72	0.84	0.68	0.75	0.74	0.78	0.73	0.72
	RandomForest	15 mins	0.92	0.94	0.88	0.93	0.87	0.96	0.90	0.95	0.91	0.92
		30 mins	0.88	0.87	0.86	0.91	0.86	0.92	0.83	0.87	0.86	0.85
		45 mins	0.88	0.84	0.86	0.91	0.86	0.85	0.88	0.85	0.86	0.83
		60 mins	0.90	0.78	0.85	0.88	0.86	0.83	0.85	0.84	0.87	0.81
	REPTree	15 mins	0.90	0.97	0.80	0.93	0.88	0.99	0.85	0.96	0.88	0.96
		30 mins	0.85	0.94	0.83	0.92	0.81	0.92	0.81	0.92	0.82	0.84
		45 mins	0.79	0.81	0.78	0.88	0.69	0.89	0.73	0.81	0.68	0.83
		60 mins	0.77	0.76	0.75	0.80	0.58	0.89	0.77	0.80	0.70	0.79
	Ridor	15 mins	0.93	0.94	0.93	0.90	0.85	0.96	0.91	0.95	0.87	0.91
		30 mins	0.78	0.92	0.82	0.92	0.77	0.90	0.79	0.92	0.85	0.87
		45 mins	0.65	0.88	0.79	0.83	0.74	0.90	0.77	0.87	0.72	0.84
		60 mins	0.74	0.75	0.76	0.81	0.71	0.78	0.76	0.80	0.66	0.84
	SimpleCart	15 mins	0.91	0.98	0.92	0.93	0.91	0.96	0.86	0.95	0.90	0.90
		30 mins	0.81	0.94	0.80	0.91	0.79	0.97	0.78	0.92	0.80	0.88
		45 mins	0.77	0.82	0.77	0.90	0.73	0.89	0.71	0.88	0.77	0.80
		60 mins	0.76	0.75	0.69	0.86	0.65	0.81	0.73	0.78	0.70	0.81

Appendix J: Information Gain For All Variables

Location	Information Gain
WS t-15	0.061523
WS t-30	0.053865
WS t-45	0.046884
WS t-60	0.040668
WS t-75	0.035853
WS DELTA t30	0.035544
WS DELTA t15	0.035008
WS DELTA t45	0.034182
WS DELTA t60	0.032938
WS t-90	0.031643
WN t-15	0.030974
WS DELTA t-75	0.030058
WN t-30	0.028768
WS t-105	0.028272
WS DELTA t-90	0.02787
WN t-45	0.026628
WS DELTA t-105	0.025993
WS t-120	0.0253
RS t-105	0.025149
RS t-120	0.025137
RN t-120	0.024676
WN t-60	0.02457
WN DELTA t-45	0.024293
RN t-135	0.024292
WN DELTA t-60	0.024227
WS DELTA t-120	0.024
RS t-90	0.023901
RN t-105	0.023887
RS t-135	0.023788
WN DELTA t-30	0.02343
WS t-135	0.02297
WN DELTA t-75	0.022781
WN t-75	0.022753
RN t-90	0.022019
RS t-75	0.021839
RS t-150	0.021568
WN DELTA t-15	0.021499
RS DELTA t-90	0.021485

RN t-150	0.021485
WS DELTA t-135	0.021479
RS DELTA t-105	0.021413
WN DELTA t-90	0.021181
WS t-150	0.020972
WN t-90	0.020944
RN DELTA t-120	0.020805
RS DELTA t-120	0.020768
RN DELTA t-105	0.020452
RS t-165	0.019887
RS DELTA t-75	0.019842
WN t-105	0.01977
RN DELTA t-90	0.019691
WS DELTA t-150	0.019682
RN DELTA t-135	0.01959
RN t-165	0.019516
RN t-75	0.019412
WN DELTA t-105	0.019394
WS t-165	0.019306
RS DELTA t-135	0.019273
RS t-60	0.018974
WS t-180	0.0183
WN t-120	0.018296
RN DELTA t-75	0.018284
RS DELTA t-150	0.018183
WS DELTA t-165	0.018002
RS t-180	0.017999
RN t-180	0.017746
RS DELTA t-60	0.017739
WN DELTA t-120	0.017726
WN t-135	0.01734
RN DELTA t-150	0.017328
WS t-195	0.016962
RS DELTA t-165	0.016877
WS DELTA t-180	0.016559
RS t-195	0.016557
RN t-60	0.016548
WN t-150	0.016437
WS t-210	0.016224
RN t-195	0.016051
WN t-165	0.015913
RN DELTA t-165	0.015741
RN DELTA t-60	0.015653

WS t-225	0.015627
WN DELTA t-135	0.015546
RS t-45	0.015427
RS DELTA t-180	0.015388
WN t-180	0.015351
WS DELTA t-195	0.015296
WS t-240	0.015239
RS DELTA t-45	0.015042
RS t-210	0.014897
WN t-195	0.014664
RN t-210	0.014529
RS DELTA t-195	0.01443
RN DELTA t-180	0.014393
WN t-210	0.014264
WS DELTA t-210	0.014247
WN t-240	0.013903
WN t-225	0.013884
RN t-45	0.013749
RS t-225	0.013524
RN DELTA t-195	0.013516
WN DELTA t-150	0.013473
RN DELTA t-45	0.013404
RN t-225	0.013158
RS DELTA t-30	0.013117
WS DELTA t-225	0.012994
RS DELTA t-210	0.012827
RS t-30	0.012565
RS t-240	0.012321
WS DELTA t-240	0.01207
WN DELTA t-165	0.011993
RN DELTA t-210	0.01184
RN t-30	0.011496
RN t-240	0.011431
RS DELTA t-225	0.011297
WN DELTA t-180	0.011221
RN DELTA t-30	0.011186
RS DELTA t-15	0.011071
RN DELTA t-225	0.01078
RS DELTA t-240	0.010345
WN DELTA t-195	0.010235
RS t-15	0.01022
RN DELTA t-240	0.009911
RN DELTA t-15	0.009776

WN DELTA t-210	0.009645
RN t-15	0.009608
WN DELTA t-225	0.009096
WN DELTA t-240	0.00891

Appendix K: Relief Values For All Variables

Location	Relief
WS t-15	0.191089
WS t-30	0.1416
WS DELTA t-15	0.117419
WS t-45	0.105496
WS DELTA t-45	0.095591
WS DELTA t-30	0.086372
RN t-45	0.073529
WS t-60	0.066004
RS t-60	0.057436
WS t-75	0.053825
RS t-105	0.049744
RN t-60	0.04902
RS t-120	0.045641
RS t-75	0.04359
WS t-90	0.040941
RS t-45	0.040769
WS t-105	0.03775
RN t-75	0.037255
RS t-90	0.037179
RN t-105	0.036667
RN t-30	0.036471
RS t-165	0.035641
WS t-120	0.033799
WS DELTA t-75	0.032758
WS DELTA t-60	0.0324
RN DELTA t-45	0.030658
WS t-135	0.030552
RS DELTA t-60	0.030303
RS DELTA t-150	0.030101
RS DELTA t-165	0.029899
RN t-15	0.02902
RS DELTA t-30	0.028485
WS t-150	0.028116
RN DELTA t-30	0.0275
WS t-195	0.027329
WS t-165	0.02732
WS t-180	0.02723
WS t-210	0.026935

WS t-225	0.026333
WS t-240	0.025905
RS t-135	0.025897
RN t-120	0.025294
RS t-30	0.025128
RN DELTA t-60	0.023684
RN t-90	0.022549
RS DELTA t-45	0.020808
RN t-135	0.020588
RN DELTA t-75	0.0175
RS t-150	0.017436
RS t-15	0.017179
RS DELTA t-120	0.015556
RN DELTA t-15	0.015526
RN t-210	0.014902
RS DELTA t-180	0.013535
RN t-150	0.013333
RS t-225	0.012821
RN t-165	0.011961
RN t-195	0.011373
RS DELTA t-135	0.010707
RS t-195	0.010256
RS t-240	0.010256
RS DELTA t-15	0.009899
RS DELTA t-90	0.009899
WS DELTA t-105	0.00981
RN DELTA t-90	0.009474
RN t-180	0.009216
WS DELTA t-90	0.008839
WS DELTA t-120	0.008453
WS DELTA t-135	0.008347
RS DELTA t-105	0.008081
RS DELTA t-210	0.008081
RN DELTA t-135	0.008026
RN DELTA t-120	0.007895
RS t-210	0.007692
RN DELTA t-105	0.007632
RS t-180	0.006923
RN DELTA t-150	0.006184
RN DELTA t-180	0.006184
RN DELTA t-210	0.006053
RN t-225	0.005882
RN t-240	0.005882

RN DELTA t-225	0.005263
RS DELTA t-75	0.005051
RN DELTA t-195	0.005
WS DELTA t-150	0.004053
RN DELTA t-165	0.003421
WN t-15	0.0024
WN t-30	0.00226
WN t-45	0.002092
RS DELTA t-195	0.00202
RS DELTA t-225	0.00202
RS DELTA t-240	0.00202
WN t-60	0.001896
WN t-75	0.001635
WN t-90	0.001497
WN t-105	0.001465
WN t-120	0.001436
WN t-135	0.00141
WN t-150	0.001391
WN t-240	0.001372
WN t-210	0.001368
WN t-225	0.001364
WN t-195	0.001364
WN t-165	0.001363
WS DELTA t-210	0.001349
WN t-180	0.001343
RN DELTA t-240	0.001316
WS DELTA t-195	0.000939
WS DELTA t-165	0.000938
WS DELTA t-225	0.000874
WS DELTA t-180	0.000611
WS DELTA t-240	0.0004
WN DELTA t-75	0.000131
WN DELTA t-60	0.000126
WN DELTA t-45	7.66E-05
WN DELTA t-30	6.85E-05
WN DELTA t-15	6.58E-05
WN DELTA t-90	6.34E-05
WN DELTA t-105	4.48E-05
WN DELTA t-120	1.86E-05
WN DELTA t-150	1.14E-05
WN DELTA t-240	1.07E-05
WN DELTA t-210	5.39E-06
WN DELTA t-135	3.44E-06

WN DELTA t-180	3.93E-07
WN DELTA t-225	-2.43E-07
WN DELTA t-165	-1.21E-06
WN DELTA t-195	-6.85E-06