2012-08-09

A Confidence-Prioritization Approach to Data Processing in Noisy Data Sets and Resulting Estimation Models for Predicting Streamflow Diel Signals in the Pacific Northwest

Nathaniel Lee Gustafson
Brigham Young University - Provo

A Confidence-Prioritization Approach to Data Processing in Noisy Data Sets and Resulting Estimation Models for Predicting Streamflow Diel Signals in the Pacific Northwest

Nathaniel Lee Gustafson

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Christophe Giraud-Carrier, Chair
Mark Clement
Jay McCarthy

Department of Computer Science

Brigham Young University

December 2012

ABSTRACT

A Confidence-Prioritization Approach to Data Processing in Noisy Data Sets and Resulting Estimation Models for Predicting Streamflow Diel Signals in the Pacific Northwest

Nathaniel Gustafson
Department of Computer Science, BYU
Master of Science

Streams in small watersheds are often known to exhibit diel fluctuations, in which streamflow oscillates on a 24-hour cycle. Streamflow diel fluctuations, which we investigate in this study, are an informative indicator of environmental processes. However, in Environmental Data sets, as well as many others, there is a range of noise associated with individual data points. Some points are extracted under relatively clear and defined conditions, while others may include a range of known or unknown confounding factors, which may decrease those points' validity. These points may or may not remain useful for training, depending on how much uncertainty they contain. We submit that in situations where some variability exists in the clarity or 'Confidence' associated with individual data points – Notably environmental data – an approach that factors this confidence into account during the training phase is beneficial. We propose a methodological framework for assigning confidence to individual data records and augmenting training with that information. We then exercise this methodology on two separate datasets: A simulated data set, and a real-world, Environmental Science data set with a focus on streamflow diel signals. The simulated data set provides integral understanding of the nature of the data involved, and the Environmental Science data set provides a real-world case study of an application of this methodology against noisy data. Both studies' results indicate that applying and utilizing confidence in training increases performance and assists in the Data Mining Process.

Keywords: machine learning, data mining, data, data processing, pre-processing, confidence, prioritization, environmental science, hydrology, diel, diel fluctuation, diel signal, streamflow, hydrogeology, watershed

ACKNOWLEDGEMENTS

I offer my sincere gratitude to Dr. Christophe in his unending patience, as well as his supportive and understanding guidance throughout this entire process, including both periods of progress and periods of drought. I'd also like to thank Josh and Melissa Gustafson for providing a helpful environment. I'd also especially like to thank Heather Ambler Williams for her tireless encouragement and mentoring, which have enabled this thesis to reach completion. Without each of these people this may not have been possible.

TABLE OF CONTENTS

CHAPTER I: INTRODUCTION

Data Mining has developed significantly over the past few decades, with ever increasing levels of automation in the process, along with the improvements to accuracy and robustness of models. There is strong support to say that as much automation as is available, human input is still essential for optimizing the process [1-3].

Data processing, in the data mining sense, is a separate but related process to training and testing. We submit that there are gains that can be made by transferring information regarding **variability in confidence** between the two. We speak of confidence here, not in the statistical sense, but in the sense of how *trustworthy* any given data point is. This can also be described as, 'how accurately does our available *estimated* value match the *actual, real-world* value?' A more precise discussion of how we use the term 'Confidence' throughout this paper is provided in Chapter III.

In this Chapter we will outline how using Confidence as a measure of any given data points' accuracy can assist in the data selection process, providing optimal data sets for training among data sets with ambiguously accurate data.

Confidence Variability

Classic data collection generally dictates that data points are considered equally valued, equally correct and equally 'representative' of the system in question. Data

points are treated equally in training and testing. Outlier detection [4, 5] is a related approach, wherein points that are quantitatively or qualitatively unusual are marked as 'outliers', and these points are generally completely excluded.

In many cases, however, data points follow a gradient of confidence; Some are exceptionally clear and accurate, some are accurate enough to be useful, others *may* be accurate enough to be useful but with some uncertainty, and others still could contain more noise than they are worth. In the classic data mining sense the data collector will either (a) use all of this data, or (b) use some ad-hoc approach to choose some data to keep and some to toss, resulting in a constant set. The shortcomings of such an approach are three-fold: First, understanding the confidence of *test* sets is useful for gauging just what your results are (i.e. You may prefer your model to be moderately accurate against a test set of accurate, confident data, rather than very accurate against a poor, unconfident test set). Second, the data collection task is inherently noisy and imperfect, and the human data collector may very often select a suboptimal set for training. Third, a **single** optimal set *may not exist!* Due to the differing strengths of different training algorithms, some may function better with a greater amount of data (even if it is not confident), or less, clean data. A standard fixed-set data collection model ***does not account for this***.

Using Confidence in Training

In our methodology, the data collector approaches the data collection task with a heuristic, automated or semi-supervised process to label their data points individually with a confidence 'score', which is used in the training and testing phase to select a range of data for training and testing, from the entire original data set to a small subset of the most confident points.

This Confidence-Prioritization can also be used to select test sets which more accurately represent the pertinent variables (or patterns) in the data: By testing generated models against a test set which contains data that is more confident (if perhaps less abundant), one can infer that testing may provide a more accurate measure of those models' ability to predict the salient patterns in that data. Therefore, by testing one's models against multiple test sets which are themselves categorized by differing confidence thresholds, more information can be derived from training and more optimal data sets for training can be determined.

Experimental Data

We apply this methodology to two studies: A simulated data set, which allows us to verify the methodology's effectiveness under known conditions, and an Environmental Data set on streamflow diel fluctuations, which both allows us to address potential gains of understanding in Environmental science from using new

data mining methods, as well as to evaluate the performance of our specific methodology's effectiveness in a real-world, environmental data case study.

Environmental Science Application

Forest management practices are essential in sustaining and regulating the use of numerous natural resources, including lumber, water, bio-diversity, and land. Proper forest management helps reduce deforestation, ensure the availability of fresh water, and protect endangered species. Environmental systems are inherently complex, involving everything from geology to hydrology to biological interactions and meteorological conditions, thus, current research findings in different aspects of ecology – hydrology, biology, etc. – are advancing the field of forest management.

Observing the rate of flow recorded over time (known as a *hydrograph*) for a given watershed can yield useful information about that given stream or river. Many watersheds exhibit a *diel fluctuation* in streamflow, wherein streamflow oscillates on a 24-hour cycle. This signal is especially strong in summer months, when as much as 30% of streamflow can be lost by the afternoon and restored the following morning. Figure 1 shows an example of a watershed exhibiting a diel signal in its streamflow output in late summer months. Most studies on diel signals [6-9] have suggested that the primary source of these signals is daily *evapotranspiration* (*ET*), the combined effect of evaporation and transpiration through plantlife. However, there remain questions as to the mechanics of groundwater transfer which produce these diel signals, and which

watershed characteristics are most critical in producing them. Indeed, it is not fully understood why some watersheds exhibit very strong signals while others of comparable size may have no detectable signal at all.

Machine Learning Methods

Machine Learning methods are not used as frequently or effectively in Environmental Science as they are in Computer Science, perhaps due to the academic and cultural gap between environmental scientists and computer scientists. Environmental scientists often meet some difficulty implementing Data Mining solutions without some support from Computer Science, and Computer Scientists have difficulty applying Data mining Principles to Environmental Science problems without pertinent domain information and guidance. One example of data-intensive Environmental Science which opens bridges between Data Mining and Environmental Science is in the HJ Andrews Experimental Forest.

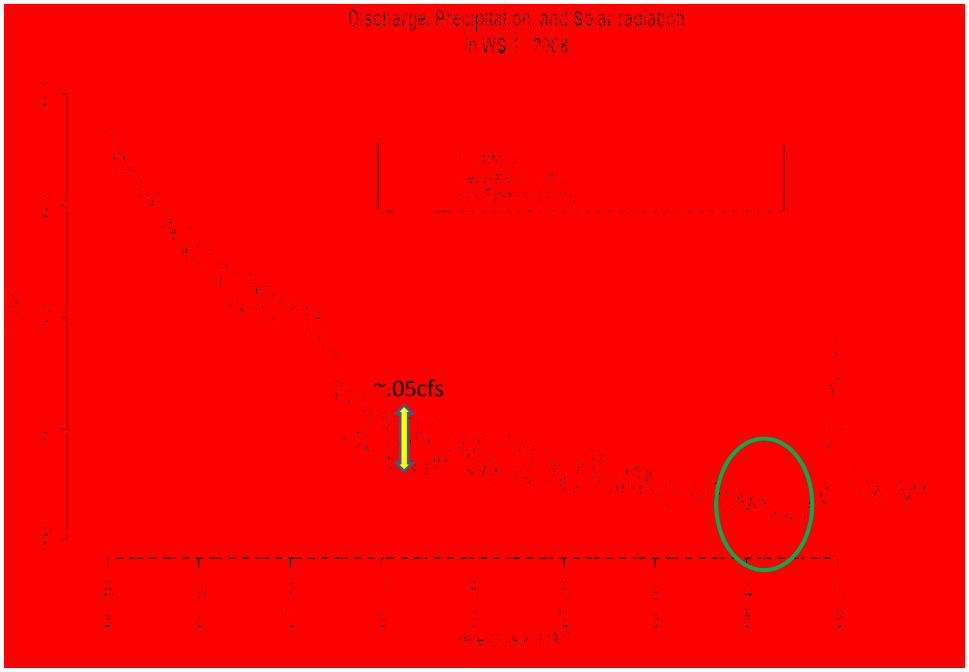

Figure 1: An example of a diel signal. A few interesting features are demonstrated here: **red** circles indicate occurrences of cloudy days (with no significant precipitation) wherein a reduction in solar radiation is very directly correlated to a reduction in signal strength for the associated days. The **green** circle highlights the diminishing of the summer diel signal, as the watershed dries out to the point of not being replenished at night.

The HJ Andrews Experimental Forest

The HJ Andrews Experimental Forest, along with dozens of other research sites throughout Northern America, has collected data continuously over the past several decades as part of its LTER (Long-Term Environmental Research) mission. This data includes streamflow from several watersheds, soil, air, and water temperature, climatic history, and dozens of other variables. The studies which have been done on diel signals typically involve one or two watersheds and seasons, leaving much of this long-

term and larger-scale data unexplored [6, 7, 9-11]. Conducting a data-mining investigation of environmental data across multiple years and watersheds has the potential of yielding useful hydrological insight. In particular, the hydrologically interesting phenomenon of why some watersheds exhibit clear signals lends itself to examination. This question was originally a motivation that led to this study, one we are not able to address entirely but do touch upon in Chapter VI.

Interactions with the EISI

The author developed the initial Environmental Science framework for this project while attending the EcoInformatics Summer Institute (http://eco-informatics.engr.oregonstate.edu). The EISI recruits undergraduate and graduate students to work on problems bridging Environmental Science to Computer Science, Mathematics, and other disciplines during a summer program at Oregon State University and The HJ Andrews Experimental Forest. Many of these problems involve large temporal or spatial data sets, for which Data Mining is used among other methods as a means of analysis. One team investigated streamflow diel fluctuations, with the author exploring the past decade of data gathered for the 10 watersheds monitored in the HJ Andrews forest. R was used to visualize data, overlaying stream discharge with solar radiation, air and stream temperature, and precipitation in an effort to identify seasonal and temporal conditions which correlated with the presence of diel signals.

Emergence of Confidence Problem

On top of other work that has evaluated diel signals on the basis of single-year and single-watershed studies [6, 7, 9-11], Our goal was to create watershed-specific models which incorporate temporal environmental conditions (i.e. Precipitation, solar radiation, temperature, and average daily streamflow) over a variety of conditions, in order to predict diel signal strength. Originally this was done with a couple interests, one being the desire to explore Data Mining in the field of Environmental Science, another being to supplement current research on diel signal generation for hydrologists interested in them.

With this intent we mined the long-term data available from the HJ Andrews Forest. As it became clear that there was a **variable degree of confidence** held in the estimated water loss as calculated for any given day – such that some records were clearly useful, others were clearly erroneous, and a great number fell on a spectrum in the middle – a need to address this confidence variability emerged.

Confidence Variability in Streamflow Diel signals

As we encountered in this data – and as is encountered in *many* Environmental data sets, which are often inherently noisy – there was a range of variability in how assuredly we could extract the diel signal strength for any given day based on the measurements available. The hydrograph of some days was consistent and regular, not unlike a sine wave, while other days had more jagged hydrographs, less regular curves

and less certain daily maxima (i.e. having multiple similar local maxima throughout the day). Other days greatly confounded a measure of water loss because of a surge in streamflow due to precipitation. We noticed that ***much of this data was likely still useful***, though it did raise questions as to how accurate our available estimates of water loss were. There was certainly also data that was significantly skewed and would mostly likely be ***harmful*** in training. If our goal was to accurately model **actual water loss** when all we had was **measured water loss**, and a measured water loss that could be significantly misrepresentative at that, **how could we best use the available mix of confident and unconfident data to do so**?

The obvious answer would be to eliminate the days that are inaccurate and misrepresentative, and maintain the days that are accurate and representative. However, **with a gradient of accuracy, how could we best determine the cut-off point?** How could we say that this mildly misrepresentative data point is an 'accurate' measure of water loss while the one that is *slightly* less representative is not?

Conception of Confidence-Prioritization in Diel Signals

With this question in mind we figured that we could take an **empirical approach**: test along a range of data sets, using varying cut-off points in terms of how much we can trust *estimated water loss* to be representative of *actual water loss*. Using a cut-off point somewhere in the range of fuzzy data points should then yield an optimal model, with both higher and lower cut-off points yielding progressively worse models –

progressively worse on one side for including more and more misrepresentative data, and on the other side for not including *enough* data.

Confidence Variability among Environmental Data sets

This approach seemed very applicable to this data set, but we soon realized that such an approach could be used on **any data set where there is a variable range of certainty regarding how measured or calculated values represent actual values**. The simulation study we conduct (as discussed in Chapter V) and this environmental data study (as discussed in Chapter VI) both utilize this Confidence-Prioritization approach, yielding effective results in examples of both simulated and real-world data. We also show that using Environmental data provided through Long-Term Environmental Research study sites, empirically generated models can model at least some aspects of complex environmental phenomena, in particular water loss from streamflow diel signals.

Thesis Statement

Using a Confidence-Prioritization approach to data collection and data mining can assist in selecting optimal data sets for training over standard, ad-hoc data collection, particularly for data sets with both accurate and inaccurate data and an unclear distinction between them.

CHAPTER II: RELATED WORK

This study touches upon aspects of both Computer Science and Environmental Science. It utilizes Confidence-Prioritization as a helpful tool to bridge the two and the unique needs that arise from such an interaction, and appropriate approaches to address those needs.

Confidence as it Relates to Computer Science

Our confidence-based approach is characteristically distinct from any studies we could find in the literature, however, there are many techniques and studies that are somewhat similar. While our confidence depends on a human data collector to assign data *confidence* values to data records and utilizes that confidence to **select** data for training, there *are* some studies which utilize either human or automated measures to **select** or **weight** data records differently in training.

Active Learning [12] is a data selection approach wherein a human is actively queried to select data for training, and for correct responses (i.e. output values) for those records. Active Learning is often effective for training, functioning well with small amounts of data records selected, because the user's expertise can be utilized to iteratively select the next record(s) which will best increase the ability of the resulting model to capture the patterns of the system in question. Curriculum Learning [13]

deals with the *order* of data records used in training, opting to start with simple patterns and work up towards more complex ones. Boosting [14] is a notable example of automated weighting, wherein misclassified data records are reweighted on successive iterations of training to better classify those missed records, while maintaining as much accuracy as possible on correctly-classified records.

In each of these instances, there is some way to derive which records are more 'simple' or more 'difficult.' However, working with confidence, we address a different question: Rather than which records are 'simple' or 'difficult,' which data records are more *correct*? And how should they be handled differently in training?

Ad-Hoc Treatment

There is often a natural process involved in data collection, where a human data collector uses their judgment in a progressive process to help extract the desired data from a system. The human data collector will generally account for any errors, starting at large ones and working towards progressively smaller ones. However, this process is almost always ***implicit and ad hoc***. What we propose in this paper is related in that it can be seen as a sort of extension to this implicit ad-hoc process, meant particularly for data sets in which there is not a clear distinction between clear, useful data, and that which is erroneous and inaccurate. Additionally, as opposed to the standard ad-hoc method, *we propose that identifying confidence* ***beforehand*** *lends additional legitimacy to training*, because it is not based on post-training, 'difficult-to-classify' adjustments.

What we propose in the following chapters – to utilize that same judgment of the human data collector to apply a more explicit confidence measure to assist in data selection – has not been found in the literature.

Outlier Detection

Outlier Detection is, again, similar but distinct from our confidence-based approach. Outlier Detection [4, 5] revolves around finding individual records which are highly dissimilar to the rest of the data, and *implicitly*, often difficult to classify. Generally, when using outlier detection, these records are eliminated from training to avoid having them skew the resulting model (and decrease accuracy).

However, eliminating data that skews results still doesn't answer the question of, 'which records are more *correct*'? Which records, be they outliers or common, difficult or simple, are data records which *accurately represent the phenomenon at hand?* More explicitly, as defined in Chapter III, which records hold high *confidence*?

In fields such as Environmental Science there can be a broad range noise and a broad range of confidence – what one might refer to as *clarity*, *accuracy*, or *representativeness* of the underlying system. For example, if a third of the data points in a data set came from highly accurate, scientific equipment, a third were made with more coarse equipment, and a third were historical estimates, should they be treated identically during training? Outlier Detection, and other approaches such as Active

Learning, Boosting, and Curriculum Learning do not address this question, as we do in this study.

Data Mining In Environmental Science and Hydrology

While most work on streamflow diel signals in particular has involved little computer science, there have been many Machine-Learning-based approaches to environmental science questions. As data-intensive environmental research has grown over the years, computer science methods are increasingly being applied to match [15]. For example, Hierarchical Bayesian Networks are often used as a systematic network for simulation and understanding, particularly in species distribution modeling [16], with some frequent application in climate science as well [17, 18].

Within Streamflow Diel Signals

In terms of streamflow diel signals, a number of studies have investigated general principles of hydrology, but relatively few have targeted questions about diel signals in particular. Gribovszki [8] conducted a review of diel signals, identifying a variety of mechanisms by which signals tend to appear: precipitation, snowmelt, freeze-thaw cycles, anthropogenic sources, and Evapotranspiration. The bulk of his review concerns ET-generated signals, especially about different methods that can be used to estimate daily ET in a given watershed using the hydrograph. He also suggested a couple key conditions for strong ET-induced diel signals: Typically shallow groundwater which is influenced by ET, and a replenishable groundwater supply.

Bond [7] suggested that direct evaporation from a stream can only account for a very small portion of ET-induced signals. She also found a shorter lag between maximum transpiration (as estimated from sapflow measurements) and minimum streamflow earlier in the summer, suggesting that in late summer months, groundwater levels have fallen low enough to either occupy slower flowpaths (inducing a greater time lag) or avoid the root zone altogether, leaving the remaining signal to be accounted for by other evaporative effects. Cadol [19] suggested likewise, that in mid-summer days signals are stronger due to a dependence on root-zone ET, and in later summer hydraulic conductivity becomes a limiting factor. They developed a model describing hydrogeological properties of a small watershed in which the riparian zone (consisting of 1-2% of the total basin area) accounted for the majority of the present streamflow diel signal. A review by Wondzell [11] suggested that the riparian zone (i.e. region immediately beside a stream) of a stream network might be contributing most of a given diel signal, with the potential of hillslope processes contributing as well.

Most studies are observation-based and a few are simulation- or computation-based, however, Barnard [6] did conduct an *experimental* study, wherein a hillslope area beside a small stream (without a riparian zone) was irrigated during dry summer months in an attempt to see if a diel signal could be produced from a hillslope area. He found that the hillslope did indeed provide a reasonable signal, and that the lag and more especially magnitude of that signal were affected by soil moisture.

More Computational Approaches to Streamflow Diel Signals

Wondzell also conducted a simulation study [11], one of the more computational papers on the subject, which suggested that both network topology and streamflow velocity throughout the watershed may attenuate the diel signal for the whole watershed. For smaller watersheds such as those in our study, or for faster streamflows, this may not be a contributing factor, but in large watersheds, or when discharge decreases enough to slow down flow velocity significantly, this can be a more critical factor to consider. Koch, et al. [20] also conducted a simulation study, using Antarctic watersheds and snowmelt-derived streamflow diel signals as their subject. Using subsurface flow models (MODFLOW, SFR2, etc.) they argued that subsurface water storage mediated [snow-melt] flooding discharges, and that unsteady streamflows tend to increase hyporheic exchange.

Significance of Streamflow Diel Signals

Móricz [21] and Zhu [22] both argue for the importance of understanding streamflow diel signals for better estimation methodology in conducting water budgets. Burke [23] mentioned that deeper and more moist soil profiles, as those found in the Pacific Northwest, correlate to heightened hyporheic exchange – Providing further evidence that this is an important phenomenon to understand for that region. Lowry [24] also pointed out how an understanding of groundwater fluctuations is "of particular importance to resource managers when considering the effects of restoration

practices or potential future impact to meadows from roads and trails on vegetation and base flow." They conducted a study on snowmelt, arguing that through various subsurface flowpaths, snowmelt may take up to 20 days or more to reach the main stream channel – accentuating the difficulty of accurate numerical estimation in dealing with groundwater flow. With the difficulty of this problem, and relatively unexplored use of computational tools within hydrology, Machine Learning and Data Mining have significant potential for being welcome new tools to the field.

CHAPTER III: PROBLEM DEFINITION

In this section we briefly discuss the Environmental Science study[1] in the HJ Andrews forest that led to this project as a whole, and the emergence of the issue of varying-confidence within our environmental data. We also discuss some of the emergence of our solution to that problem, and discuss how it can apply to similar problems in other data sets, which we provide a methodology for in the following Chapter.

A more precise definition of 'confidence' is also discussed here, in order to set a framework for defining how Confidence-Prioritization can help select optimal data sets for training.

A Need in Hydrology, a Need in Environmental Science

Our original motive for this project was to investigate the factors contributing to a pattern of diel streamflow fluctuations in small watersheds in the Pacific Northwest by utilizing Machine Learning techniques. As covered in the Related Works section,

[1] We cover our Environmental Science project only briefly here; For a more complete treatment please refer to Chapter VI.

previous literature hadn't investigated the application of ML as applied to streamflow diel signals, and we saw the benefit to be gained by investigating this application.

The data-gathering portion of this project required multiple steps of acquiring, sorting, and processing instantaneous, 15-minute samples of environmental factors (i.e. precipitation, average daily streamflow, air temperature, and solar radiation) into daily summary values. These daily summary values would then be used for training a model to help elucidate what factors affect diel fluctuations.

Of great importance was the measure of diel streamflow fluctuation: The dependent variable this model was made to predict. In hydrology, typically the diel streamflow fluctuation for a particular day is calculated by measuring the area between the streamflow curve and a line drawn across the daily peaks (see Figure 2).

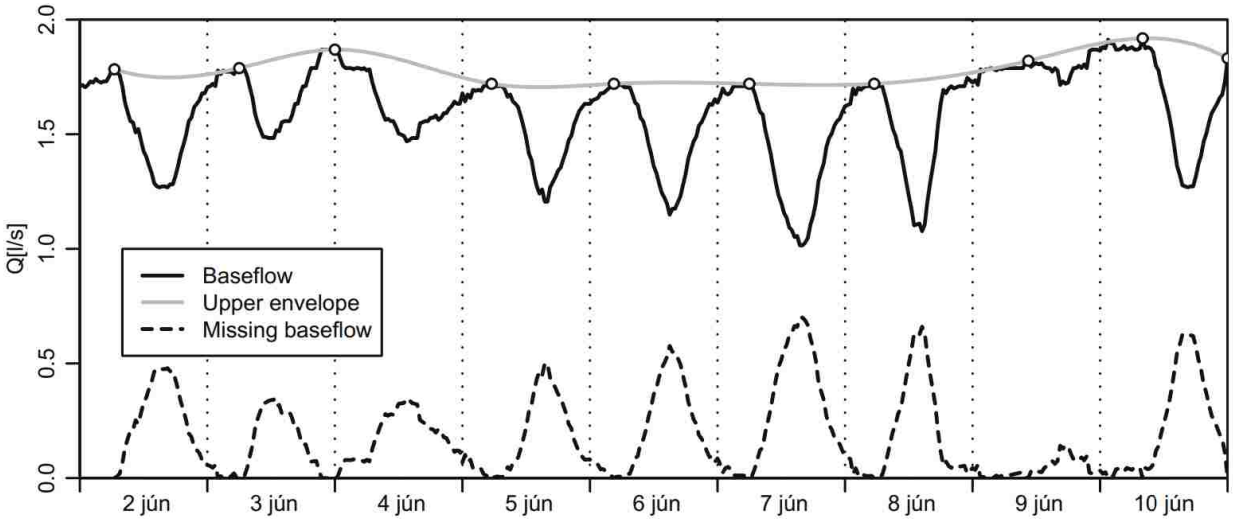

Figure 2: Diel Streamflow Fluctuation Estimation Method from [8]. Baseflow is estimated to run along the maximum flow points for each day. Water loss is estimated as the area between the two curves.

However, upon closer examination of the data, we recognized that while there were times that this method yielded a very accurate estimate, there were other conditions – such as precipitation-modified streamflow, equipment that was less sensitive, or simply a lower signal-to-noise ratio – that ***did not totally discount the relevancy of a given data point, but did decrease our confidence in our measurement of it***. Figure 3 provides some salient examples of this. Notice that signals are quite clear towards the right, as the season enters mid- to late- summer. As you move left into earlier summer, signals are still present, though less visibly clear and with some more jaggedness and irregularity. Towards the left, in winter and spring, signals may be present but are much weaker and noisier – Though not necessarily distinctly outliers, in

the sense that their daily streamflow, precipitation, solar radiation, and temperature may not be appreciably different from in the summer. Notably, a data point's confidence does not necessarily correlate with it being an outlier in any sense; **Data points can be confident outliers, as well as unconfident non-outliers**.

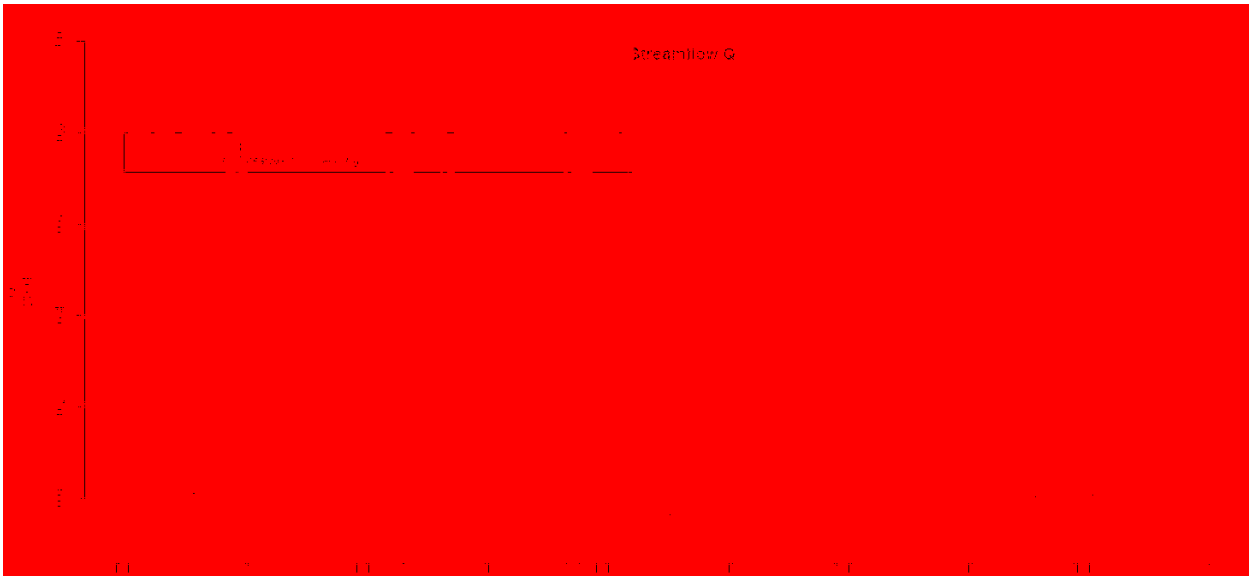

Figure 3: Hydrograph for a portion of 2001 in Watershed 1. Light red is original streamflow, dark red is the residual from the daily streamflow average window, along with a line connecting daily maxes, like in Figure 2. This chart shows examples of differing levels of signal clarity for measuring diel streamflow fluctuations. Throughout all seasons, precipitation spikes interfere with diel signals considerably, which also decreases confidence.

From this point, we began wondering which data to include in training and which to exclude – And we created rules and measures to estimate the relative confidence any given day's measure of diel signal fluctuation. (These methods are described in greater detail in Chapter VI.) Through this approach, we also began

addressing the link between data processing and training/testing. We surmised that useful information that the human annotator gathers about the confidence of one data point as compared to another during the data collection phase can be very useful during the training and testing phase, to increase the accuracy of generated models.

General open problem

The standard for the data mining model is that data is provided from some source, organized, and run through some combination of learning algorithms to expose inherent patterns of interest. In this process, information about the data collection, organization, and annotation process is typically unused in the learning process.

Often it is up to the data collector to clean data as well, deciding on an all-or-nothing basis whether or not any given data point is included in training or not. This often comes down to a question of whether a given data point contains enough noise that it will only help to confound the model in training, or whether it has enough useful information that can be utilized to help the model to be more accurate. Unsurprisingly, there is a wide spectrum of data quality, but the only decision the data collector has is whether or not to *include* that data record. At that point, all data records are essentially equal.

Variable Confidence, Determined Beforehand

Additionally, since a human data collector often implicitly uses their judgment anyways, to discern viable data in the data collection process, we propose that this

implicit, ad-hoc process can be extended and strengthened by explicit confidence determination. Particularly, we propose that gains can be made for data sets with a variety of confidence and ambiguously erroneous/accurate data, as we discuss further in this section. We also suggest that applying confidence **before training** holds advantages over the iterative process of training and readjusting data selection **afterwards**, based upon which data points are misclassified.

Current Methods for Handling Difficult Data

There have been some approaches that can 'handle' or eliminate points which may not be useful, some examples would include Outlier Detection, Active Learning, and Boosting. While there is much work that has been done in outlier detection [4, 5], outlier detection almost always handles outlying points by simply eliminating them. Not as much work has been done to address those points that may be outliers ***but still valid points***. Active Learning [12] has either the user or some automated system iteratively and individually select points for training, which proves to be a fairly effective technique for training, generating models which can account for both 'simple' and 'difficult-to-classify' data points.. Boosting [14] also does a fair job of handling difficult-to-classify data points; it is a well-known weighting/voting technique where points that are misclassified are re-weighted on successive iterations to improve performance over multiple successive trainings. Boosting effectively addresses *outliers*

and other difficult-to-classify data points, however, it still does not address the question of **training with data that is variably misrepresentative of the system at hand**.

Complexity of Data

Whether a point is or is not an outlier, if it is valid it may be useful to training – and whether or not another point is *not* an outlier, if it is inaccurate or misrepresentative it may harm training. If it is somewhere in between, it may do either. Outlier detection, like many data mining approaches, often assumes a 'ground truth' model, where outliers are excluded in order to score better on the rest of the data, *which is assumed to be a correct 'ground truth'*. In Environmental data [15], oftentimes **there is no ground truth available**, all data available will include varying levels of inherent noise.

In most cases with Environmental Data there are not only known complicating factors – Such as how a cold front may affect weather patterns over the next several days – but many *unknown* factors, theoretically even things which are unnamed, because they are not yet studied or understood. In the extreme case this can be viewed as the analogy of how a butterfly flapping its wings can cause a hurricane on the other side of the world, but in practice it can often hinge on more present, but still unknown factors. The shape of bedrock underneath a watershed will affect the patterns of water flow out of it, as will volumetric soil density. These can still be difficult or impossible to quantify – However, as with all Data Mining, a balance can be found between complexity and accuracy.

Human Intuition

There is an argument for humans being able to judge factors which may be difficult for computers to determine [25]. How environmental data is collected and collated may affect its inherent accuracy in *representing the underlying system*, but that will not be manifest in the raw data. For example, in Environmental data collection, a data collector might know that a given pattern on the solar radiation data is from a bird nest on the equipment for those dates, rather than the weather being cloudy. Or they may know that population measures for one area of the forest are less accurate because predators send many animals into hiding, not necessarily reducing the population.

A human data collector will be much better able to determine the accuracy of a given data point than, for example, an automated outlier detection algorithm that clumps all outliers as 'invalid' and others as 'valid'; While Outlier Detection and similar methods are well studied, in this paper we investigate a different principle: that of Confidence.

Defining Confidence

'Confidence,' as used throughout this paper, can be defined as *the degree to which a data point is quantitatively accurate in regards to the system which it is meant to represent*. This can be thought of as 'accuracy', 'reliability', 'consistency', or 'precision'. This is NOT, by contrast, simply selecting for the data points which represent the model the human data collector may desire, but it does require intelligent selection. There are a

few considerations in dealing with confidence that we discuss here: **imprecision**, **context**, and **variance**.

Imprecision

Inasmuch as a measure of confidence **is inherently not precisely quantitatively definable**, a human data collector's knowledge of what may contribute to additional variance in individual data points is required. The human data collector is *not correcting these inaccuracies*, as the nature of such inaccuracies is that they are providing an uncertain effect (positive? Negative? How much?), but are recognizable. The human data collector therefore is only estimating which points are prone to being more or less accurate. This may involve differences in measurement equipment, compiling composite data from separate sources, or confounding environmental factors, as we discuss further in this paper.

Context

Confidence is context-specific; that is, depending on how data is collected or otherwise how it came to be, confidence will differ. The goal is to neither take all data points as equal, nor to select data based on personal bias, it is to apply confidence for accuracy in representing **the particular variable in question**. This is an important distinction to make. Factors which confound a given data point's accuracy may not always be determinable from the data point itself, it may be associated with the collection process. This makes more sense when looking at data points, not as raw

collections of associated numbers, but as abstractions, representations **estimating** another factor.

For example, in this project we use 15-minute-interval stream discharge data to estimate daily water loss, which is an abstract estimate of how much additional water *would be assumed* to flow out of a watershed on a given day were it not for a cyclic, daily 'decrease' in flow. In truth the raw data as provided only show measurements at 15 minute intervals, but by inferring water 'lost' we extract an abstract measure, calculated numerically, but abstract in nature. As such, a human data collector may judge the numerical methods used to provide a more accurate estimate of the abstract value desired on some days than others. If the base streamflow fluctuates unexpectedly in the middle of the day, estimations become more difficult; minor rainfall may affect estimated water loss in either direction (positive or negative) and major rainfall will likely cause an over-estimate in water loss, but it is difficult to say how much of an overestimate. If, for one year, only hourly data was available (instead of 15-minute intervals), that would also decrease confidence. Associations such as these provide a context which can be used by the human data collector to calculate confidence.

Other Potential Contexts for Confidence

We do not attempt to comprehensively cover potential situations in which confidence can be applied, but varied contexts will provide useful information for confidence determination. For example, data that is collated from multiple data bases

of differing quality or reliability may apply higher confidence for records from more reliable sources. For a collection of personal survey data, lower confidence might be given to entries which are incomplete or somehow profile with people who provide uninformative results (such as always selecting 'a', selecting only extremes, etc.). Any time composite summary values are calculated for environmental data (eg. As with estimated water loss), due to the inherent noise of complex environmental systems, similar contexts for confidence determination will be present.

In essence, this process bears some resemblance to what many human data collectors intuitively do during data extraction – that is, identify suitable records for collection and eschew unsuitable ones. However, as we discuss further below and in Chapter IV, there are benefits to making this process more explicit. Using confidence as a measure enables a more sensitive range of data selection to occur, picking up on subtle distinctions between data points which lay on the (often wide) line between those data points which are distinctly accurate and those which are distinctly erroneous.

Variance

Some data sets are explicitly exact in all of their measures, others are known to be inherently noisy and include 'general' inaccurate measures across all entries, either indiscriminately or *without an understanding of how a variance of confidence might exist between data points*. Other data sets, as we handle here, include a **variance of confidence**

which the human data collector identifies. That variance of confidence between data points is critical for our Confidence-Prioritization methodology to assist in training.

Proposed Solution

We propose that one inherent question of data mining – That is, where does one draw the line between data that is clear and representative and therefore included for model training and testing, and data that is noisy or inaccurate enough to be considered more harmful than not, and therefore excluded from training and testing – can be effectively addressed by combining the knowledge of a human data collector with an automated process in training which utilizes that knowledge.

Of course, such a task requires some extra human attention over fully automated methods, but with some streamlining of process it does not require much, and can pay off such an investment with better, more accurate models. Active Learning [12] provides evidence that a computer-directed, human-augmented approach can aid training and increase accuracy while decreasing the number of training instances required. Similarly, we seek to use an efficiently balanced, human-augmented approach to improve the data mining process.

Also, in comparison to the Environmental Science problem, it should be noted that this is more than simply a signal processing problem; While some error can be reduced by clever signal extraction, Environmental systems inherently contain a significant amount of noise [15], often from factors that are unidentifiable within the

scope of feasible research. However, whether these sources of noise are naturally inherent in the system, artifacts of imperfect equipment, or errors of technique along the way, **the human annotator often has a good sense of what data is more accurately representative of the phenomena at hand, and which are more heavily influenced by noise and other factors**.

So two questions come from this issue: (1) How can a human data collector best go about efficiently and accurately annotating their data with a measure of confidence (as opposed to simply identifying outliers)? And (2) Once that confidence data is determined for the data set, how can it be used to best augment training? We address both of these questions in the following Chapter.

CHAPTER IV: METHODS

In this Chapter we focus on the additional information available during the data collection phase regarding confidence – such as the environmental data's variability of confidence in diel signal strength as discussed in Chapter III – and discuss how it can be used during the data mining phase. We propose a general methodology that can be used as a guideline for incorporating data point-level confidence, provided by a data collector, into training.

The two main issues we address in this methodology are (1) that of how a human data collector can approach their particular data collection task in such a way that they can efficiently and effectively assign confidence values to individual data records, and (2) How the training process can be modified to best make use of the data point-level confidence values.

We also provide an outline of our experimental approach, which highlights how the Simulation Study in Chapter V and the Environmental Science Study in Chapter VI provide evidence that Confidence-Prioritization assists in the data selection process to provide more optimal data sets for training.

Overview

Data collection is done with various methods and means, and therefore our approach retains some flexibility and necessarily requires some adaptation to particular data sets.

However, since there are many common approaches in data collection, our approach is also specific enough to be useful. The basic outline is provided in Figure 4.

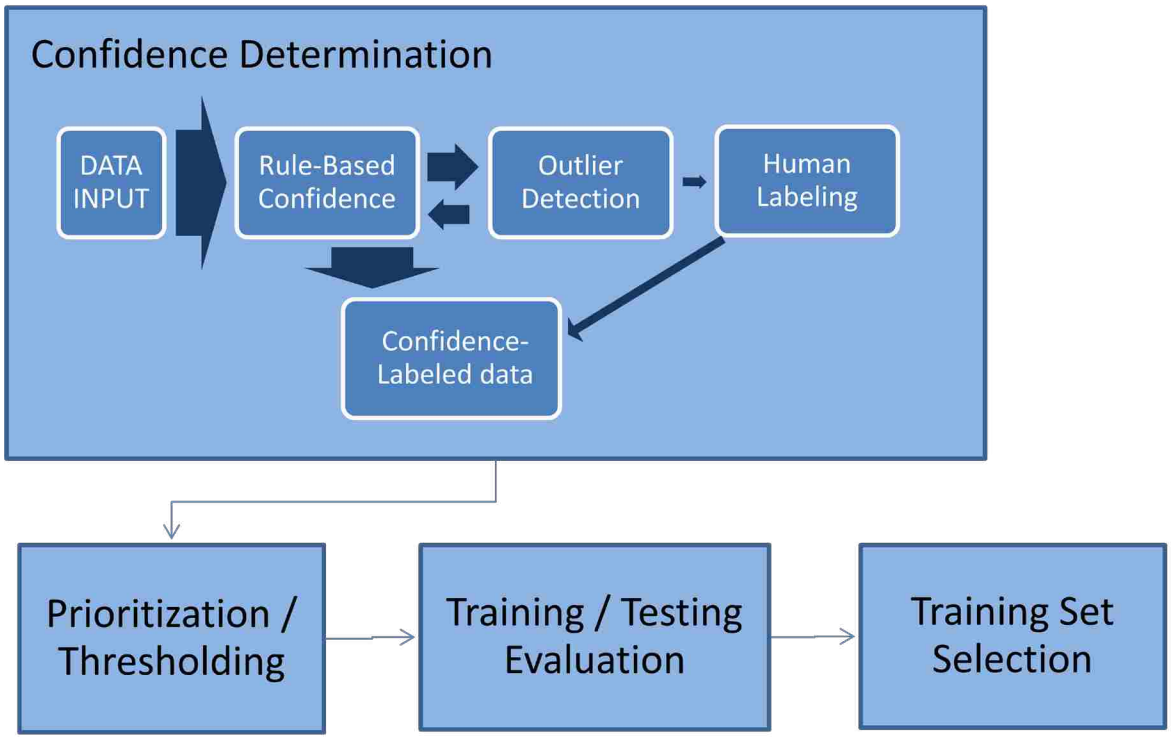

Figure 4: Overview for our Confidence-Prioritization methodology. Widths of arrows inside the Confidence-Determination step are meant to infer typical amounts of data flow. They are not to scale, but wider arrows do infer greater amounts of data flow.

A data collector, during the data collection phase, undergoes a process to annotate individual data records with confidence values. Using appropriate Rule-

Based confidence value assignments (as we discuss further below), they assign confidence to data records – up to the majority or even entirety of them. For some exceptional records, rules that otherwise assign reasonable confidence estimates to other data records may fall short, and a Human Labeling process will be more accurate. Of course, it would be unwieldy to rely on Human Labeling for any more than a handful of records. Outlier Detection algorithms help to identify particular records that may require special attention.

The purpose of all of these steps is to automate the process that a data collector would, if given infinite time and energy, use to assign confidence to all individual data records. Human-Labeling ALL records would most likely provide the best results, but that is almost always unreasonably costly. By using a heuristic of Rule-Based Confidence assignments to approximate what the data collector *would* assign manually given infinite time and energy, this becomes a much more feasible task. Augmented appropriately with Outlier Detection, it can become even more approachable.

After this Confidence Determination step is complete, the data set has been provided with per-data-record confidence values. These values are treated on a relative rather than raw numerical basis, i.e. the **order** of records sorted by confidence is of more significant interest than the raw confidence values. The now confidence-prioritized data set is used to create multiple training sets, which are evaluated for comparative effectiveness and selected from to generate an optimum model.

All of these components are discussed further in this Chapter, below.

Confidence Determination

The goal of this component is to label individual data records with a relative confidence value through the use of heuristic tools and the judgment of a human data collector. There are multiple different tools available which we discuss here: **Rule-Based Confidence**, **Outlier Detection**, and **Human-Labeling**. Rules can determine confidence for many data points (even most or all of them), while the various available Outlier Detection methods [4, 5] are subsequently used to identify outliers which are not already well accounted for by rules. The human expert takes these outliers, as well as any other points they determine are not being assigned appropriate (we discuss what is 'appropriate' in the next paragraph) confidence values, and either (a) pipes them back into the Rule-Based Confidence step to generate another round of rules to account for them, or (b) manually tags these data points with confidence.

Necessarily, this process relies on, to an extent, human intuition. Approaches such as Active Learning [12] and Mechanical Turk [25] provide salient examples of human intuition augmenting the data extraction/mining process. As with Active Learning, it is not to say that confidence could never be determined by automated methods, however, since it relies on complex factors *per data set*, a human familiar with the data set and extraction thereof will, in practice, often be the quickest guide to determining confidence with consistency. In this paper, we take an approach that

requires a human data collector, leaving full automation of confidence determination to potential future work.

The human expert's knowledge of the data here is helpful for selecting rules for defining relative confidence; this can be based on a given data point's source, clarity of signal, surrounding conditions, or any other factors that may affect or confound data accuracy. **At the very least**, the resulting confidence values should have some positive correlation to each point's actual accuracy. It can be helpful to remember that because this approach does not use the *scale* of confidence (as discussed later in this chapter), exact confidence values are not so important as the **order**. It is more critical that greater confidence values are assigned to more accurate, representative data points. Also, *relative order is much simpler to calculate than raw numerical values*, which simplifies the confidence-determination process. Using all of this as a guideline, in the end, the human expert is the one whose judgment determines what 'appropriate' confidence values for any given points should be, and thus how adequately their rules are assigning confidences to data points, and whether more rules should be added or more points manually tagged.

These three factors, **Outlier Detection**, **Rule-Based Confidence**, and **Human-Labeling**, affect one another. Any or all of them may be used. For very small data sets, the data collector may simply manually label all points. They may also opt to forego

manual labeling, choosing instead to generate enough rules to effectively account for all (or nearly all) data points.

Outlier Detection, as well, may be foregone. In our study in Chapter VI, we chose to take a more manually evaluative process, observing the confidence values resulting from successive rules, and adding more rules until the resulting confidence values were 'adequately' (as discussed above) consistent with human judgment of confidence across a wide spectrum of data points.

Rule-Based Confidence

Rules are the first line of attack, so to speak, of generating the confidence variable. They are, generally speaking, of the form:

```
If (some condition) applies to (said data record),
  Then (modify data record's confidence in some way),
    AND/OR (tag this record for Human-Labeling).
```

For example, if a feature in some data record is extracted from a composite of other data, and that composite varies in clarity of signal or certainty, then applying such a rule(s) to increase confidence by signal clarity would be appropriate. If there were some question of accuracy from data of different sources – such as can be the case with conglomerate data bases [26] – then assigning some additional confidence to more trustworthy sources could be appropriate as well.

The key to rule generation is that rules are meant to be a **heuristic**. They're meant to make the process of Confidence Determination simpler for the data collector. With most data sets, there will be large tight clusters of points that are easy to account for with simple rules, followed by progressively smaller clusters that can be accounted for by other rules, followed by outliers that may warrant additional rules or Human-Labeling.

For example, for our environmental study in Chapter VI, the simplest, most common pattern among diel signals was their tendency to have maximum flow at the same time of day during successive days. After making a rule for this, we saw that there were many cases where there were days with valid signals that were not accounted for because streamflow would remain near the maximum level for several hours during the day and the *absolute* maximum from one day to the next may change by several hours, even if the wave-form shape of the curve between the two days does not. So, we made a rule that measured the variance of timing for the top 20% highest flow points throughout the day and associated lower variance with more confident signals. Then we saw that some individual days for which there was NOT an actual signal would exhibit high confidence by these rules, so we added rules that granted additional confidence to given days for being adjacent to other days with high-confidence signals, and reduced confidence for singletons – and so on.

This process for Rule-Based Confidence is not unlike Active Learning [12] or in some ways Curriculum Learning [13] in that you handle the simplest or most general cases first, and work your way up to the more complicated data points. Active Learning would indeed be a good candidate for the Confidence Determination step, the user assigning confidence values to data points which are used to progressively build a model to estimate confidence for all data points – Which confidence values would then in turn be passed forward to the rest of the Confidence-Prioritization process for training and testing, as we discuss later in this chapter. However, we do not pursue this application of Active Learning here, but recommend it in future work.

Either way, by using rules, a human data collector greatly streamlines this process and reduces the amount of time, effort, and money required for what could otherwise be an expensive human-labeling task.

Outlier Detection

While we will not go very in-depth here about Outlier Detection, [4] and [5] provide greater treatment of it. Outlier detection is primarily useful here to assist in determining which points are unusual and thus may not be accounted for by rules that handle a general observed sample from the Rule-Based-Confidence step. Such points can then have rules generated to handle them, or can be human-labeled for confidence.

Outlier detection is not explicitly necessary in the presence of other rules for confidence generation, indeed in Chapter VI we opt to not use it, instead digging

manually into the data to make observations about outlying situations. However, it does provide a fairly accessible well of algorithms and approaches to draw from in situations where the complexity of the data set is out of the scope of the data collector to observe manually. Many algorithms for performing actual Outlier Detection are available [5]. We suggest that Outlier Detection is used to identify data points which fall outside the realm of rules that account for the more general body of data points. If a data set is not prohibitively large, it may be simpler to observe such points manually. Either way, these 'outlying' points can then be accounted for with further rules or human-labeling.

Human-labeling

Obviously, due to cost of time and effort, this is a secondary option for data records where confidence is not well-determined by more set-specific rules made by the data collector. However, for sets of just a few records it may be easier to Human-Label than create new rules. With tractability in mind, and reserving Human-Labeling for only those few records that rules have the most difficulty determining confidence for, it is not infeasible that a data set of thousands of records can be annotated fairly quickly.

Prioritization / Thresholding

In this approach, we do not assume that a data collector will necessarily be able to tag their data with a to-scale measure of noise or variation, and indeed with this approach scale is not necessary to. What is critical is that, for the most part, order is

fairly well maintained from least-confident to most-confident data records. Confidence will be used to create a prioritized list of data records, which is then specially used in the training step. While future work could certainly be done to implement raw confidence values – such as by weighting – we wished to avoid the constraint of having a data collector make confidence values specifically numerically relevant beyond accurate order. Being able to say "This record is more accurate or reliable than that one" is a much more accessible and feasible task for these data sets than "This record is precisely twice as accurate or reliable as that one." At the same time, Prioritized lists of data records provide great functionality already, and allow us to use multiple different confidence-thresholds to select data sets for testing, as we discuss further below.

Training on Confidence-Prioritized points of data

We provide here a framework of steps for utilizing confidence-tagged data records. This does not imply that other methods are not also useful, but our purpose is to show that confidence-tagging in data sets is useful, and we provide this as a readily available means to demonstrate that.

In traditional data mining methods, the data collector decides by binary inclusion/exclusion which data to include and which to not include for all of training. In many cases this consists, simply enough, of accepting all data points. In many cases this may be prudent, particularly when dealing with data that doesn't involve much noise or uncertainty. However, in such data sets (as, for example, in Environmental

Systems) including those points which are low-confidence (refer to Chapter III for further discussion on the definition of confidence) may be harmful to training, so data collectors may opt for more of an *ad hoc* process to include or exclude data points in a binary fashion, to pass forward into the training algorithms. This binary inclusion/exclusion often cuts short valuable information that the data collector had: A knowledge of which records were more confident than which others. Given that a data collector already needs to gather knowledge of the data set at hand to make reasonable decisions regarding data inclusion, utilizing that knowledge in our methodology provides a more thorough and accurate approach for training.

The basic approach is to train in steps, including progressively smaller subsets of the data, filtering out by a progressively increasing threshold of confidence. The underlying theory being that: The most inaccurate and misrepresentative, lowest-confidence data records will harm training by misrepresenting the system in question (in practice, the variable they are meant to estimate or define), and thus by using too many records and including those inaccurate/misrepresentative records, you may diminish the resulting model's accuracy – However, by using only the most confident records, you also risk training with too *little* data, thus also diminishing the resulting model's accuracy. Somewhere in the middle is an optimum confidence threshold to train with, which will avoid the sparseness problem of using too little data, and also the problem of using too much (inaccurate) data.

Training Set Selection

By training along a continuum of multiple confidence-threshold steps, for many (not all) data sets, a peak accuracy will be evident somewhere along the curve. Often this peak will occur somewhere in the middle, between a low threshold that uses too much low-confidence data, and a high threshold that uses too little data overall. At times this peak may occur at quite high or low points, or even at either extreme end – for example, if *all* data points are actually fairly confident, it may be best to train using all data points. Selecting this confidence threshold, and the associated data set, constitutes the final step of our process.

Some additional work does need to be done by the data collector to annotate records with confidence values, but with our methodology's dynamic data selection approach, the data collector has a greatly increased chance of finding optimal data sets for training by using our approach. To demonstrate this improvement, consider a human driven binary *ad hoc* inclusion/exclusion approach – it can be viewed as a simplified version of our methodology, where points have only '0' and '1' confidence. Allowing a finer gradient of confidence values, as tuned by the same human data collector, allows for training on this set, as well as ranges of data that may be useful but not included in the single data set extracted in the binary inclusion/exclusion approach. Particularly in that range of middle-confidence data where the accuracy is enough in question that they may be considered harmful to training, but confident enough that it

could be useful, the human collector may make an arbitrary decision to include or exclude such points. Our approach ensures that more accurately representative points are included, and misrepresentative points are not.

In Chapter V we provide an example implementation of this approach.

Testing Sets

It is noted, of course, that training progressively more confident data sets on themselves will generally result in building progressively more 'accurate' models. We wish to test against a 'ground truth' set, **however, in uncertain or complex conditions such as in environmental science, a ground truth set often does not exist**. Often, then one will use as close to a ground truth set as can be found.

This brings up the question of using different confidence thresholds in testing set determination. The purpose of the testing set is to be as representative as possible of the system being evaluated (for example, streamflow diel signals, as we discuss in Chapter VI). Using a test set that includes a high level of uncertainty or lack of confidence may fail to represent the system properly. Extremely narrow or sparse test sets could also fail to represent the system properly and encourage overtraining to certain conditions.

Confidence-Prioritization can therefore assist in the test set selection process as well. Correlation, RMSE, and other performance metrics describe how a given model performs against a given test set, but what that test set represents alters interpretation of

the results. We discuss further just how Confidence-Prioritization can be helpful by using examples in the discussion section of Chapter V.

Note on Dependent and Independent Variables

Throughout this document we exercise an approach that applies confidence to the *estimated* variable in our models. This could be done as well on any other fields in a very similar fashion; Training of the model is adjusted by exclusion of data points containing relatively less confident *estimated variables*, for finding a confidence threshold which optimizes model efficacy. Future work may warrant further investigation into the specifics of this, but for the scope of this study we limit the application of Confidence-Prioritization to the estimated variable.

Note on Learning Algorithms

Choice of training algorithm, as with any data mining task, depends on the data. For our Simulation Study, we use M5Rules and MultiLayerPerceptron for our regression model training algorithms. M5Rules is fairly effective for extracting regression patterns in systems that contain multiple inherent patterns beyond simple linear regression, while MultiLayerPerceptron is likewise known to be fairly effective for extracting hidden non-linear patterns, and is a common enough algorithm that it provides a good baseline. For our Environmental Case Study we use M5Rules and MultiLayerPerceptron, in addition to KStar. KStar, which uses instance-based

classification and has a modification to handle regression, is another algorithm that could be effective for environmental data with complex patterns that simpler regressions would have difficulty capturing. These are all well-known, effective algorithms, chosen for their ability to handle diverse and somewhat complicated regressions.

Additional effective algorithms such as Boosting – or more particularly for regression, AdaBoosting [27] – exist, but we do not pursue them here. Boosting in particular relies upon the premise that you are training and testing with a **ground-truth data set,** which we explicitly do not assume in this study. It is therefore uncertain how Boosting should apply in our methodology, and we leave the evaluation of that application for future work, along with the evaluation of other training algorithms using this same methodology against more data sets.

As with any data mining task, results vary according to choice of Learning Algorithm. Correspondingly, we would expect that by using different algorithms, different optimal confidence thresholds may be found.

This, of course, only emphasizes the need for an approach such as the one we have here; using a data collector's single selection of which data to include or exclude limits various algorithms by not finding that optimum confidence-threshold point *per algorithm*.

Outline of Experiment

First, we will conduct a Simulation Study, allowing us to answer the question, '*Provided that confidence DOES correspond to more accurate data, and using data with a broad range between accurate and inaccurate data points, does using Confidence-Prioritization help identify optimal data sets for training?*' Then, after addressing that question, we present our Environmental Science Study, wherein we apply Confidence-Prioritization to predict water loss for each given watershed based upon average daily Solar Radiation, Precipitation, Streamflow, and Temperature. This study addresses a different problem than the Simulation study, the question of, '*In a data set with human-estimated confidence values (i.e. where the association of high confidence with more accurate data is uncertain, as opposed to the simulation study), does using Confidence-Prioritization help identify optimal data sets for training?*'

Following that, we discuss the results of both studies and their implications for Confidence-Prioritization and its capacity to help select optimal data sets for training, addressing specifically our thesis statement, that "*using a Confidence-Prioritization approach to data collection and data mining can assist in selecting optimal data sets for training over standard, ad-hoc data collection, particularly for data sets with both accurate and inaccurate data and an unclear distinction between them.*"

CHAPTER V: SIMULATION STUDY

We've discussed both specific and general reasons why our Confidence-Prioritization methodology can be useful, but up until now the discussion of the methodology itself has been theoretical in approach. In this and the following chapters, we apply this methodology to, first, a simulation study, and then a real world study of diel streamflow fluctuations in small world watersheds.

As mentioned in the Outline of Experiment in Chapter IV, this Simulation Study addresses the question of '*Provided that confidence DOES correspond to more accurate data, and using data with a broad range between accurate and inaccurate data points, does using Confidence-Prioritization help identify optimal data sets for training?*' This opens the way for addressing the additional uncertainty of how human-estimated confidence relates to actual data accuracy, which we do with the Environmental Science Study in Chapter VI – And in turn our thesis statement that Confidence-Prioritization helps identify optimal data sets for training.

By applying this methodology to a simulation study, we provide a proof of concept for the effectiveness of Confidence-Prioritization, as well as explore the conditions under which it does or does not improve performance, and discuss further implications of use. One critical advantage of using a Simulation Study is the knowledge regarding data sources; Using real world data (particularly data with a large

amount of inherent noise, such as environmental data) one is limited in the assumptions that can be drawn regarding that data or ground truth, and therefore any estimations of confidence are (by nature) inexact. With a simulated data set it is much more reasonable to derive conclusively whether or not a result is accurate, and therefore assign a 'known' confidence. This is not meant to be a standard for Confidence-Prioritization in the real world, but a step in the process of our study to provide evidence that *if there* ***is*** a correlation between confidence values and data points' actual accuracy, that our Confidence-Prioritization approach will help identify optimal data sets for training.

Individuation

As with all methods, some discretion is needed when applying our methodology to different circumstances. In exercising our methodology in this simulation study we provide conditions that we intend to be general for the sake of application and comparison to other situations, however, as a specific application it cannot encompass the full range of potential data sets to which this methodology might be applied.

Throughout this section we will address this where appropriate, providing some discussion of potentially different conditions and how such needs for individuation could be met. In particular, we will address some differences in:

(a) How, and how quickly, less confident data deviates from accurate values,

(b) How much (reliable) data is needed to build a good model, and

(c) How correlation, RMSE, or other measures may best address users' goals.

(d) Using Confidence-Prioritization in input or output variables.

Data Generation

The data for our simulation study were generated using R. Four input variables and one confidence variable were each drawn from five separate distributions. Two separate noise factors were also drawn from other distributions, the first representing noise inherent in the system, and the second representing noise which can be attributed to varying levels of confidence. The first, 'natural' noise factor was independently drawn for each data point, and the second, 'confidence' noise factor was drawn independently and then weighted according to the level of confidence for each data point, as defined by the confidence variable. The one output variable was then calculated from the four independent variables, and summed with both noise factors.

In order to represent a variety of data distributions, the input variables were drawn from, respectively, a beta distribution, a log-normal distribution multiplied by a binomial mask (so that, in effect, a portion of values are zero and the rest follow the log-normal distribution), a beta distribution summed with the first input variable and a log-normal distribution. The confidence variable was created as a sum of two beta distributions, to provide somewhat of a 'peanut'-shaped, bimodal distribution, where there was a cluster of roughly non-confident points at 0.0 to 0.4, and a smaller cluster of confident points at about 0.8 to 1.0.

```
generate_data <- function (n=1000)
{
  #first input variable
  var_1 <- rbeta(n, 6, 2, 0) * 7

  #second input variable (combination of two distributions)
  var_2 <- rlnorm(floor(n/2), 3, .4)
  var_2_mod <- rgamma(ceiling(n/2), .8, 2)*30
  var_2_mask <- rbinom(n/2, 1, .45)
  var_2 <- var_2*var_2_mask
  var_2 <- sample(c(var_2, var_2_mod))

  #third input variable
  var_3 <- rbeta(n, 11.0, 5.0, 0) * 9
  var_3 <- var_3 + var_1

  #fourth input variable
  var_4 <- rlnorm(n, 1.6, .70)

  #confidence variable, determining confidence
  var_confidence <- sample(c(rbeta(floor(n*.82), 1.7, 4.4, 0),
                             rbeta(n-floor(n*.82), 6.9, 1.7, 0)))
  var_confidence[var_confidence<0] <- 0 #Truncate negative values

  #output variable - base value, before noise
  out_var_base <- 4*sqrt(.07/(.35+abs(2.8-var_1)))
  out_var_base <- out_var_base +
      mapply(function(x,y){if(x>11 && x<13){1-2*((x-12)^2)}
                             else{sqrt(y/10)}},
             var_3, var_2)
  out_var_base <- out_var_base +
      mapply(function(x,y){if(x<20){x/30}
                             else{(y-10)/4}},
             var_4, var_3)

  #natural noise
  natural_noise <- (rnorm(n, 0, .15))

  #Confidence noise. The more confident, the less influence this has.
  offset_noise <- rlnorm(n, meanlog=2.5, sdlog=0.6) * 0.2 - 8

  #final output variable
  out_var_w_noise <- out_var_base + natural_noise +
                offset_noise*((1-var_confidence)^4)*1.5

  #[... output resulting variables ...]
}
```

Figure 5: Data generation script in R for Simulation Study. Constants were empirically determined to provide, for each input variable, a reasonable distribution of values (see Figure 6) and comparable contribution to the output variable.

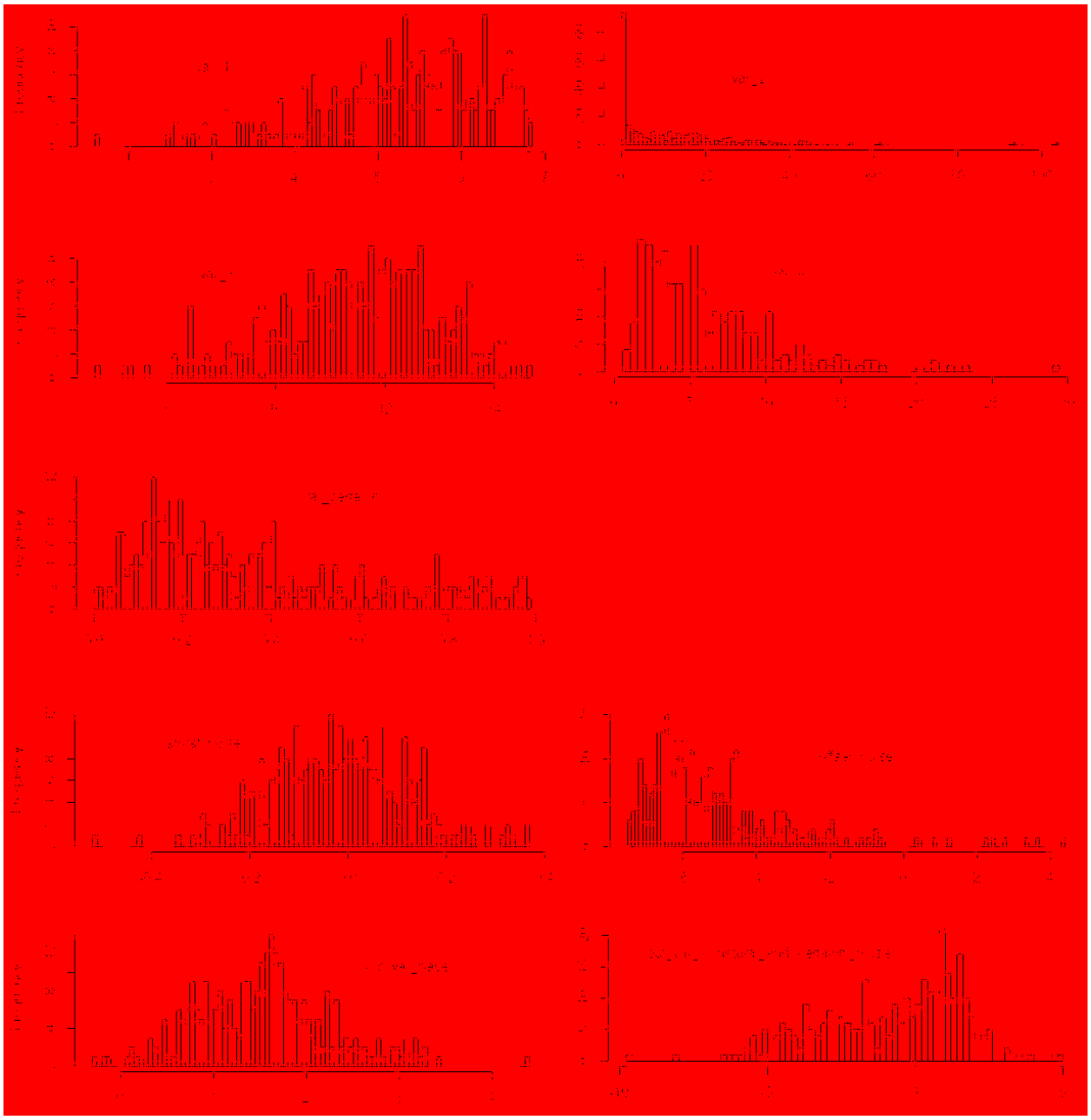

Figure 6: Resulting data distributions from calculations shown in Figure 5. Histograms shown are for a data set of size n=300.

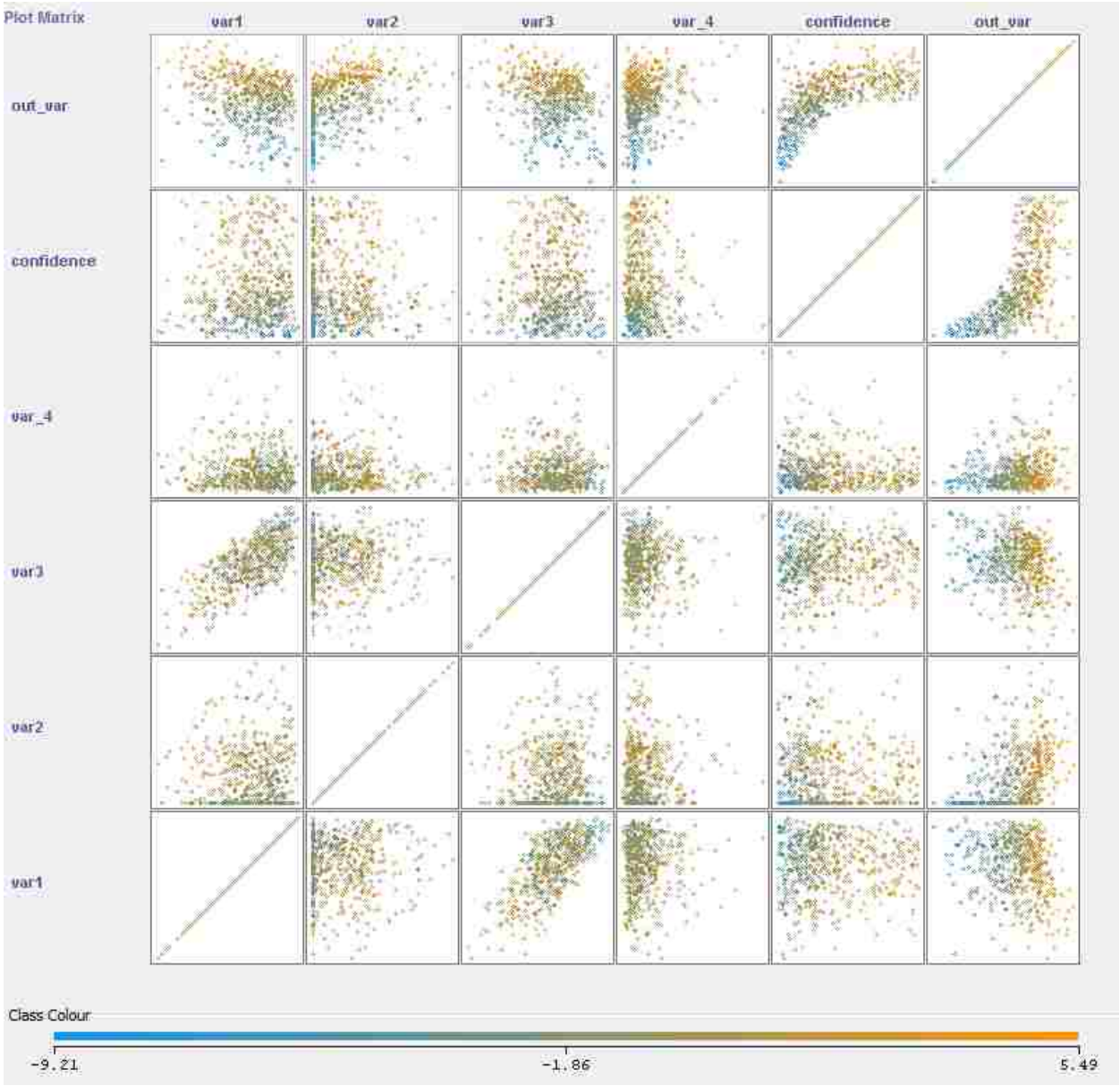

Figure 7: one-to-one relations of input and output variables and confidence. Data set size of n=500.

The calculation of the dependent variable was weighted from each input variable such that no single variable had a significantly greater influence than the others on the estimated variable across data points. It includes a combination of square roots of, divisions by, and linearly scaled additions of the input variables.

The natural noise factor was scaled to provide some noise to the dependent variable, but a relatively smaller influence (approximately half) than any of the independent variables. The confidence noise factor was weighted to increase at a greater rate with greater noise, which can be said to be representative of a de-stabilizing effect of lack of confidence, wherein the less confidence one has in a data point, the progressively greater chance that point will be farther from truth. A generation script for all of these values, written in R, is shown in Figure 5. Figure 6 shows data distributions for the resulting values, and Figure 7 provides one-to-one relations between the variables.

Methods for Simulation Study

Handling Confidence

As stated above, the purpose of this simulation study is to show that confidence-based training can provide a benefit over standard approaches. As such, confidence in this simulation study is necessarily known by the study, associating higher-confidence values with more accurate results. Obviously, in other real-world data sets, the human data collector will be filling this role and not have a 'ground truth' confidence available. However, the same principle – that higher confidence data records are more accurate – should still apply.

In this approach, we used a confidence distribution that was independent and bimodal on a scale of 1 to 100, with a slightly greater portion of points leaning towards

the less-confident side. This is meant to represent somewhat of a 'noisy' data set, one in which some points were collected under conditions that provide clarity, but most have some inherent noise factors confounding them to some degree.

Confidence-Based Noise

A natural-noise element was included in every data point generated, representative of real-world data sets; there is generally some natural noise inherent in both the system and measurements that cannot feasibly be accounted for (particularly in Environmental Science, as we address in Chapter VI). A confidence-based noise factor was also made and applied to data points, to a degree dependent on each data point's confidence. At lower confidence this 'confidence' noise was quantitatively more significant than the 'natural' noise, and at higher levels of confidence the natural noise is more significant.

There is an accelerating rate of noise as confidence decreases; high-confidence records take relatively little impact from incremental confidence value differences, and low-confidence values harbor a greater amount of noise for the same incremental confidence differences. This is meant to represent somewhat of a 'decay' effect, in that small confidence deficiencies in high-confidence data points do affect fine-tuned accuracies, but at lower confidence a point may be confounded by significant factors, even to the point of being a degree of magnitude off. For an example of this, observe the difference that precipitation spikes can make in measured water loss in Figure 3.

Prioritization

As discussed earlier, prioritization of confidence values for training, rather than numerical treatment, is an important constraint because it releases the data collector from the requirement of creating a numerically-significant confidence value, leaving them to only handle the more feasible task of sorting records by confidence. Of course, in this simulation study we use the numerical values of confidence in order to calculate the noise that will apply to any given data record, but for training these raw numerical values are not used, but the order of records, as sorted by confidence, is. In Chapter VI a more complete discussion is provided on this.

Percentiling

In order to utilize the order of confidence values among data records without weighting their raw values, we use percentiles. At the 50^{th} percentile, for example, we use only the top 50% most confident data records, and at the 10^{th} percentile we use only the top 90%, and so on.

Training

Using confidence-percentiles as described above, we generate models for a given data set across nearly the full range of percentile values (i.e. 0 up to and not including 100), in 5-percentile increments. From each of these runs we save the correlation and error in various measures (RMSE, etc., we will refer primarily to RMSE for this study

from this point on, but other measures of error can also certainly be used as appropriate), which we discuss further below in the results section.

Control Condition

In order to compare our Confidence-Prioritization approach to a 'default' data set, we use the 0^{th} percentile data in training as a control. This provides a comparison of numerical results to show evidence for the relative performance of Confidence-Prioritization against a default data set selection.

In most cases, using all data available is also a very realistic data selection method, in that many data collectors will simply use all data available for training. While it is not feasible for us to evaluate an unbounded variety of more exclusive ad-hoc selection methods, we ***do*** compare our numerical results across confidence training percentiles to show what a variety of ad-hoc approaches might derive – More discussion, of course, is given on this in the Results and Discussion sections of this chapter, and the Environmental Science Study in Chapter VI. This provides additional evidence to demonstrate where gains made by using Confidence-Prioritization are *more likely* to be significant.

Training Algorithms

We use M5Rules and MultiLayerPerceptron for model generation. M5Rules is fairly effective for extracting regression patterns in systems that contain multiple inherent patterns beyond simple linear regression. As the data generated in this study

has reasonable regression subpatterns (see Figure 5, Figure 6 and Figure 7), M5Rules should be a reasonable fit. MultiLayerPerceptron is likewise known to be fairly effective for extracting hidden non-linear patterns, and is a common enough algorithm that it provides a good baseline. We show and discuss the results of these later in this chapter, in the results section.

Redundancy

For this simulation study, due to the relatively higher amount of noise in the model we generated, we chose to generate redundant runs in order to clarify the signal in the results. More specifically, we generated m full data sets (m=30) and ran them along each of the 20 percentile steps for training. Using one run can still show the same pattern, but by exercising this redundancy, we provide more robust results below to verify that, on average, this approach is indeed going to help.

Alternative to Redundancy

Generally speaking, not as much redundant data will be available for testing in real-world data sets. While redundancy helps provide a proof of concept in this simulation study, we do design this approach to be applicable to real world data.

When training on data, if results are not already evident using training at incremental 5-percentile marks, training can also be done in smaller increments, removing even 1 or two data records at a time, depending on the size of said data set. While this is likely to vary metrics of interest (i.e. confidence, RMSE, etc.) widely, by

using an averaging window that signal can be smoothed out to determine an optimal training percentile.

Testing

Since cross-validation is not so informative for a training approach that removes less-confident values (i.e. removing them from BOTH the training set and test set, of course, results in higher performance, unsurprisingly), we use fixed test sets across different percentile-runs, as well as across the multiple redundant data sets.

These test sets, like real world test sets might, don't exclude noise entirely, either from natural inherent noise or that from confidence. We run against test sets that are each 300 data records in size, but vary by the range of noise within each of them. Using the same data and confidence-value generation described earlier in this chapter, we generate four test sets separately, each from different percentile-ranges of confidence: 0%, 50%, 90%, and 98%. This variety in testing is meant to represent the variety that occurs in real world testing; Sometimes more accurate or even nearly optimal test sets, such as the 90% or 98%, are available, while other times a limit on data may restrict you to testing on a data set with more or even all noise, such as in the 50% or 0% test sets.

Again, we compare the results of each and discuss them below.

Finding best fit point

Simply enough, plotting correlation or RMSE across the confidence-percentile values tested, we find a peak in correlation (or alternately, minimum in RMSE) along that curve.

In some cases, this peak may occur on either extreme end of the curve; If data is exceptionally noisy and only the very few most confident values are close to accurate, using as little data as possible will be beneficial. If the confidence-based noise is not necessarily significant, then using as *much* data as possible will be beneficial. However, in many more cases, there will be some optimal point where enough clear data is provided to build an accurate model, and not too much noisy data is introduced to befuddle it, and that max/min can be found along this curve.

Without Redundant Data/Clarifying Signal

Again, as discussed above, real world data sets may not have the redundancy used in this simulation study, and smaller percentile steps may be taken. If smaller percentile increments are used, the resulting correlation or RMSE values can be averaged using a window with appropriate sizing to clear out noise and identify an optimum training point.

Note on Correlation vs. Error

In many cases choosing to maximize correlation results in the same confidence-percentile training point as minimizing RMSE would. However, in some cases

correlation and error can both be increased, or both be decreased. For an example that is somewhat of a boundary case: a model may assign a constant value for all predicted dependent variables if trained on, say, a single confident data point. This will have very poor (i.e. 0) correlation, but if that single data point is within the actual (or test) values' range, then a constant-value estimation may not have terribly high RMSE. Including other, less confident, significantly skewed data points that are available may have negative impacts on the resulting model's RMSE, by erroneously widening the range of values assigned to the dependent variable well beyond actual (or test) values for the dependent variable. At the same time, having more data, even poor data, could likely increase correlation above 0.

Results of Simulation Study

Figure 8 and Figure 9 visually summarize results using M5Rules. They include a comparison of results using two result parameters (*Root Mean Square Error* and *Correlation*) across data sets of three original sizes (n=500, n=200, n=40), both not using and using the confidence value as an additional parameter. They include all of these factors among twenty subsequent confidence percentile points (0,5,10, ... 95), *including the default test set of the 0th percentile*, and testing against four different test sets with varying confidences.

Notable patterns to consider here are:

- An optimum performance point (one which minimizes RMSE and/or maximizes Correlation) does indeed appear, from around the 40th to 80th confidence-percentile, depending on which training and testing sets are used. This supports our general hypothesis; that when the variable accuracy of data points can be associated with a confidence value, that confidence value can be used to determine which data points are helpful and which are harmful for training.
- As expected, as the size **n** of the base training set increases, overall RMSE drops and correlation increases. Having proportionately more data available of all confidences *should* improve performance.
- As the confidence (and implicitly, quality, accuracy, representativeness) of testing data increases, optimum confidence-percentiles for RMSE increase, while they remain unchanged for correlation. This makes some sense; when testing against a higher quality test data set, the patterns of the test data should be a little simpler and more similar to higher quality *training* data. Training on higher-quality data can increase *correlation* of estimated to actual output values, even if it increases error by, perhaps, spreading the range of estimation out and under-training. It may be that, as the additional noise from unconfident data *cannot* be successfully accounted for, correlation can at least be increased with less, higher-confidence data – leading to a

model that doesn't as accurately estimate output values (thus an increase in error) but greater correlation.

- When testing against higher-confidence test sets (particularly 90 and 98), as the size of the base training set **n** increases, the optimum confidence-percentile rises for both RMSE and Correlation.
- Only when testing against the poor-quality data test set (0) did using all low-confidence data in training yield the lowest RMSE and highest correlation.
- Using confidence as an attribute in training and testing yields no performance gains at optimal or near-optimal confidence-percentiles, but at sub-optimal training percentiles does increase performance as compared to training without confidence.
- Using confidence as an attribute in training, the effect of performance gains based on specific training percentile are diminished significantly, as compared to training without confidence as an attribute. Essentially this may imply that that including confidence as an attribute does not necessarily make the model stronger, but does allow for more variance in data selection.

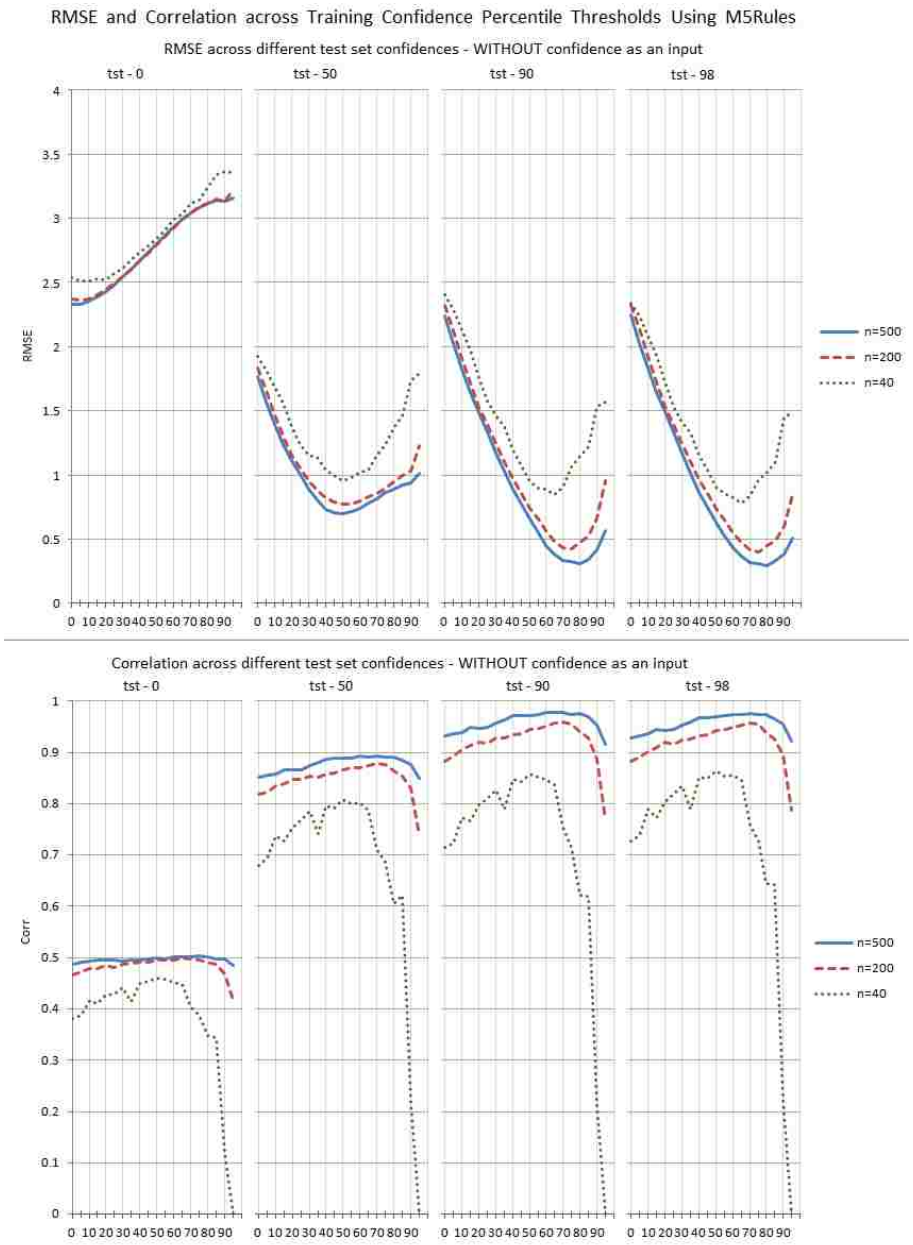

Figure 8: Results of our approach using the M5Rules learning algorithm. Does NOT include confidence as an attribute (Figure 9 does). Top half shows RMSE and bottom half shows Correlation, as tested on four different test sets: (from left to right) 0th, 50th, 90th, and 98th percentile-confident data, each test set having n=300. The three training sets start with n=500, n=200, and n=40 as shown in the legend, and decrease in size as narrowed by percentile-confidence, as indicated on the horizontal axis, in 5-percentile increments.

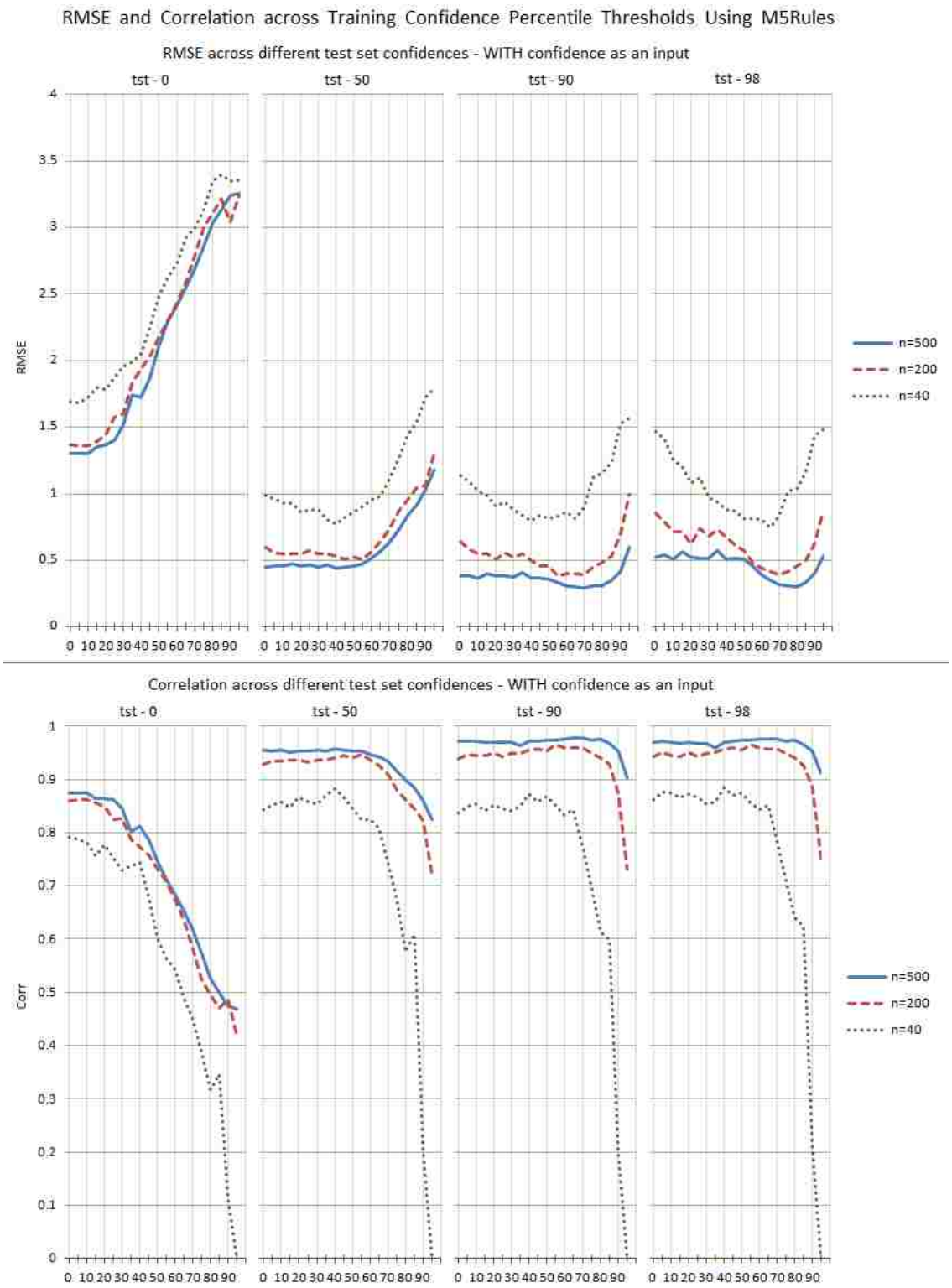

Figure 9: Results of our approach using the M5Rules learning algorithm, USING Confidence as an attribute. Formatting is the same as for Figure 8.

Figure 10 and Figure 11 likewise show results for the same data using MultiLayerPerceptron as a learning algorithm. Besides those patterns listed above for results on M5Rules, notable **differing** patterns in this result set are:

- The optimum confidence-threshold for RMSE does NOT increase with greater training set size **n**. While M5Rules may be able to build better models with less data using simpler rules, that MLP uses neural networks and back-propagation may mean that MLP preferably runs using more data for training, thus opting to keep the extra data and not increase the training confidence threshold.
- The performance of MLP seems to be a more sensitive to selected confidence-threshold than M5Rules. It may be that M5Rules builds rules that are relatively similar regardless of training confidence threshold (and the resulting data), but with MLP, extra, unconfident data can significantly throw off the neural network so as to overtrain on false, noisy patterns that can emerge from unconfident data.

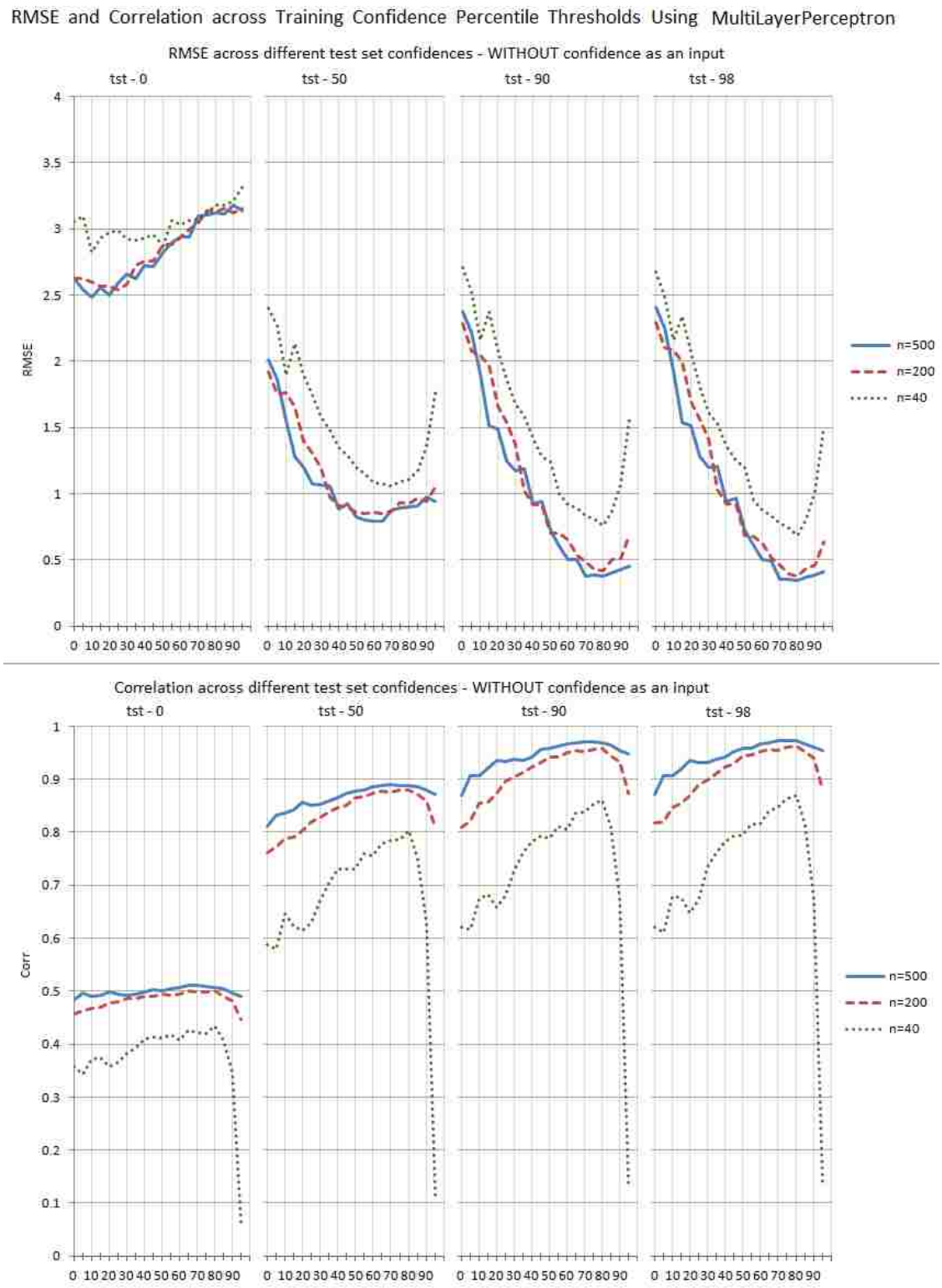

Figure 10: Results of our approach using the MultiLayerPerceptron learning algorithm, NOT using Confidence as an attribute. Layout is as described in Figure 8.

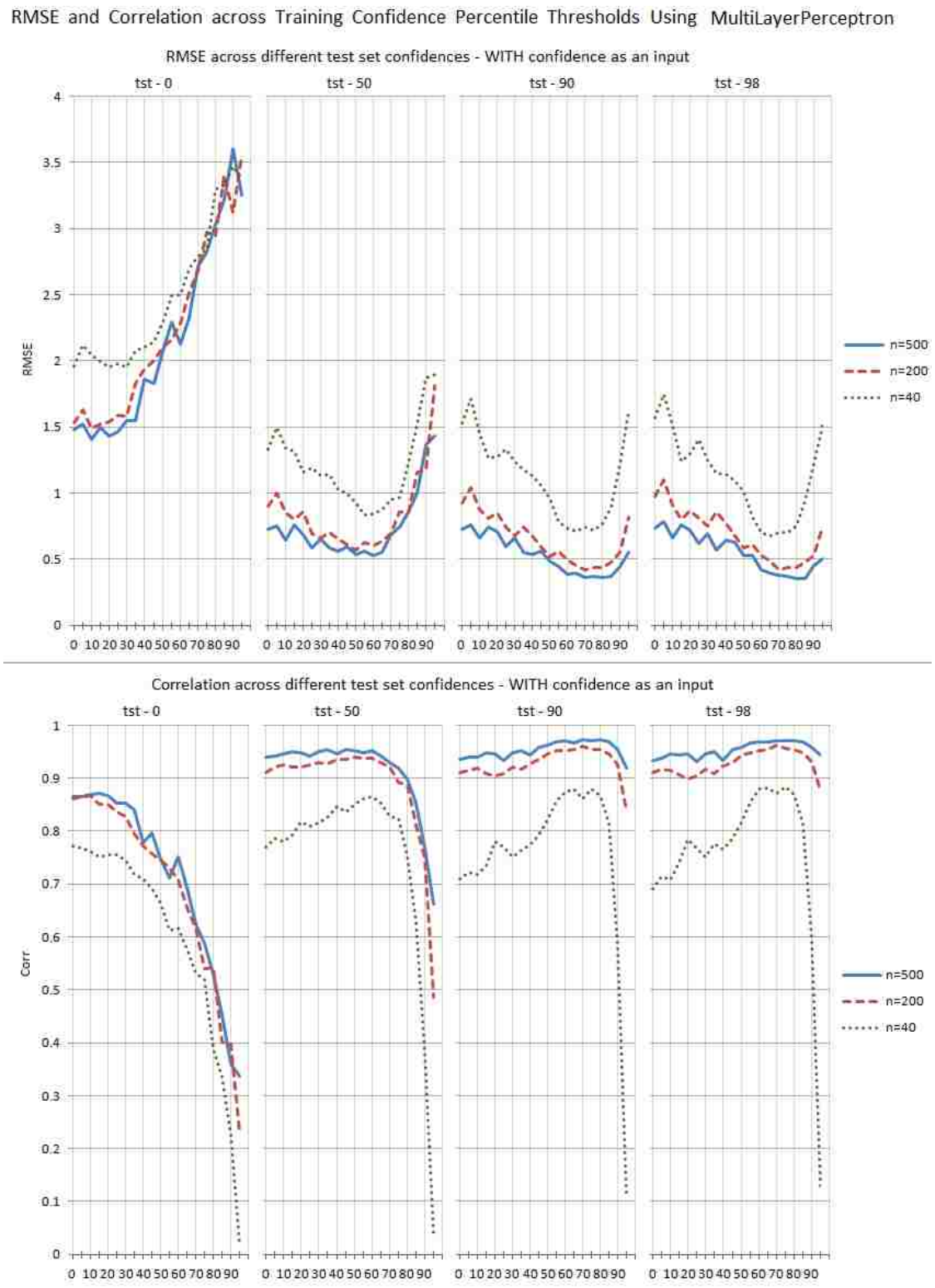

Figure 11: Results of our approach using the MultiLayerPerceptron learning algorithm, USING Confidence as an attribute. Layout is as described in Figure 8.

Figure 12 and Figure 13 include both results curves for M5Rules and MultiLayerPerceptron for comparison, without and with confidence as an input variable, respectively (i.e. combining Figure 8 and Figure 10, and Figure 9 and Figure 11).

Figure 14 and Figure 15 numerically summarize results, both of a default baseline of using the 0^{th} percentile confidence for training (i.e. using ALL data available for training), and of the optimal RMSE and Correlation values and their corresponding training percentiles. Interesting items to note:

- Using Confidence-Prioritization provides an improvement across all testing percentiles as compared to the baseline of 0% confidence.
- This improvement is less accentuated, of course, when using confidence as an input variable. Not surprising, as having confidence available in training can help mediate for the negative effects of unconfident data.
- Observing the relative gains of optimum Confidence-Prioritization performance over the default of using the 0^{th} training percentile, as shown in Figure 14 and Figure 15, shows that relative gains of Confidence-Prioritization increase when (A) the test set is more confidence selective, or (B) the initial test set is smaller.

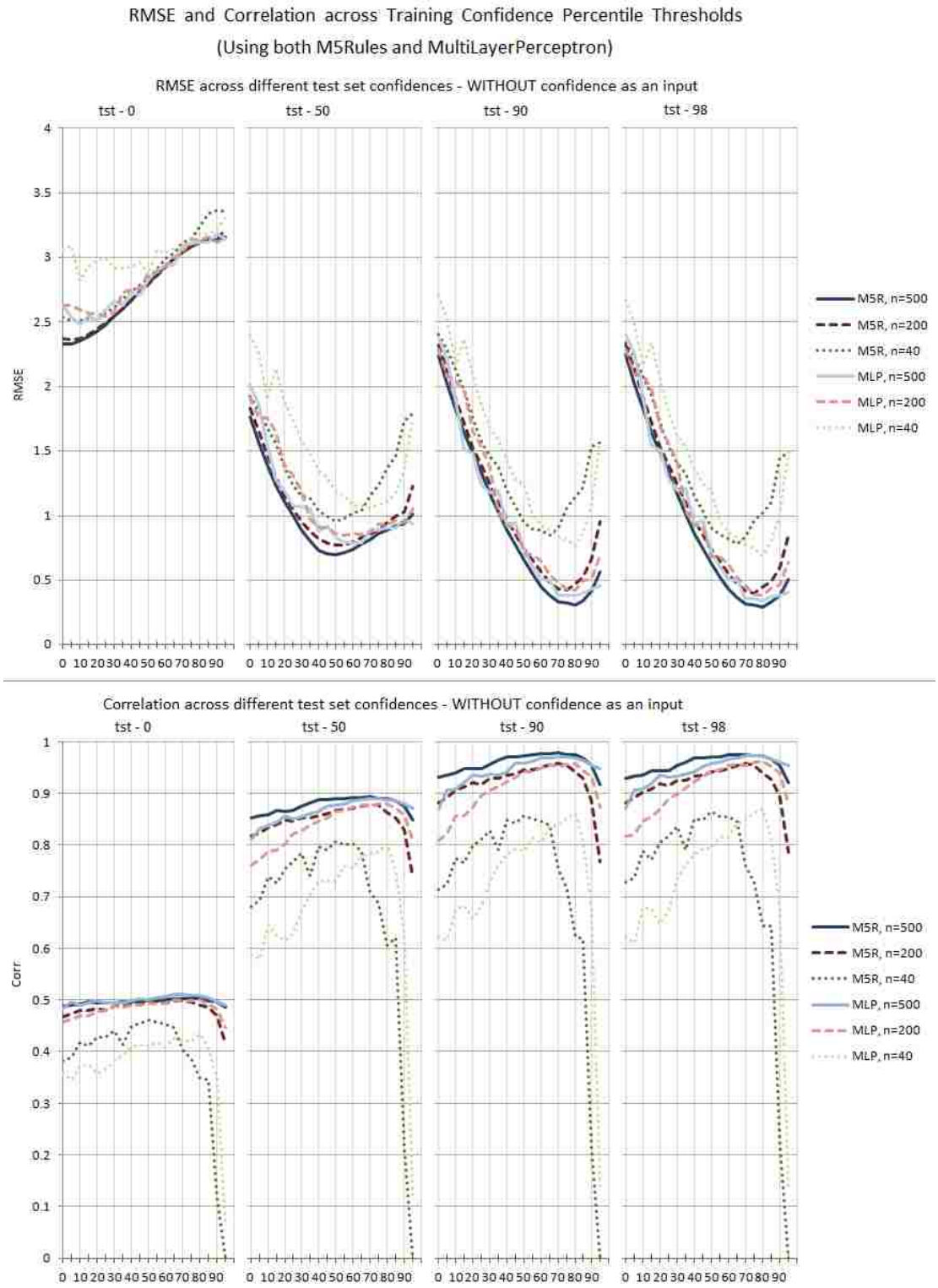

Figure 12: Results of both M5Rules and MultiLayerPerceptron, NOT using Confidence as an attribute, superimposed on one another for comparison. Layout is as described in Figure 8.

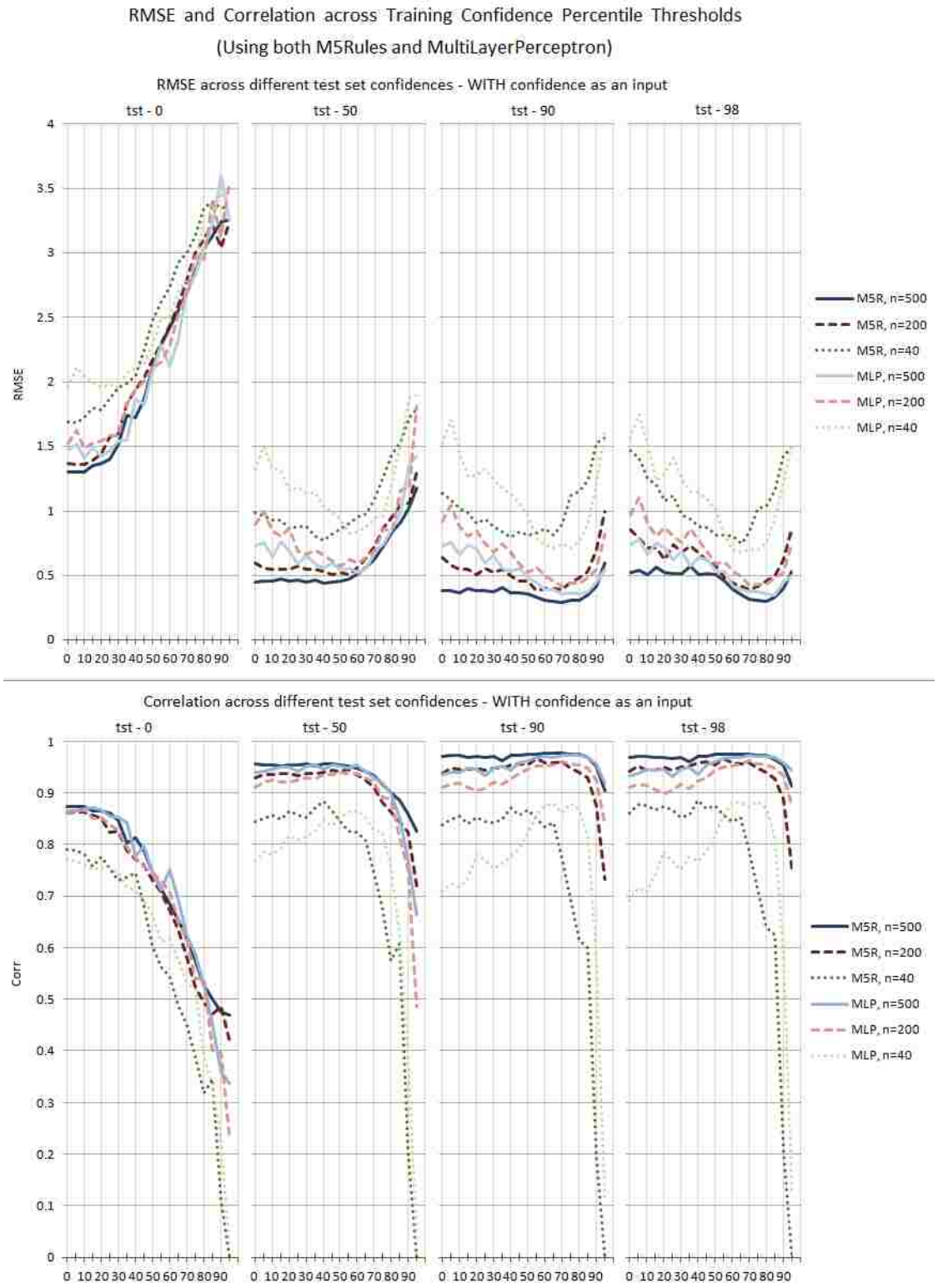

Figure 13: Results of both M5Rules and MultiLayerPerceptron, USING Confidence as an attribute, superimposed on one another for comparison. Layout is as described in Figure 8.

M5R: At 0%ile Confidence for Training:

tst%ile\n	RMSE w/o Conf 40	200	500	RMSE w/ Conf 40	200	500
0	2.54	2.37	2.33	1.69	1.37	1.30
50	1.92	1.83	1.77	0.97	0.60	0.45
90	2.40	2.32	2.24	1.14	0.64	0.38
98	2.34	2.32	2.25	1.47	0.86	0.52
	Corr w/o Conf			Corr w/ Conf		
0	0.38	0.46	0.49	0.79	0.86	0.87
50	0.68	0.82	0.85	0.84	0.93	0.96
90	0.71	0.88	0.93	0.84	0.94	0.97
98	0.73	0.88	0.93	0.86	0.94	0.97

Improvement of Optimum:

	RMSE w/o Conf 40	200	500	RMSE w/ Conf 40	200	500
0	0.03	0.01	0.00	0.01	0.01	0.00
50	0.97	1.06	1.07	0.19	0.09	0.01
90	1.55	1.89	1.93	0.35	0.26	0.09
98	1.56	1.92	1.95	0.72	0.47	0.22
	Corr w/o Conf			Corr w/ Conf		
0	0.08	0.04	0.01	0.00	0.00	0.00
50	0.13	0.06	0.04	0.04	0.02	0.00
90	0.15	0.08	0.05	0.03	0.03	0.01
98	0.13	0.08	0.05	0.03	0.03	0.01

Optimum RMSE/Corr, and associated Training Confidence %ile

tst%ile\n	RMSE w/o Conf 40	200	500	RMSE w/ Conf 40	200	500
0	2.51	2.36	2.33	1.68	1.36	1.30
50	0.95	0.77	0.70	0.78	0.51	0.44
90	0.85	0.43	0.31	0.79	0.38	0.29
98	0.78	0.40	0.30	0.75	0.39	0.30
	Corr w/o Conf			Corr w/ Conf		
0	0.46	0.50	0.50	0.79	0.86	0.87
50	0.81	0.88	0.89	0.88	0.95	0.96
90	0.86	0.96	0.98	0.87	0.97	0.98
98	0.86	0.96	0.98	0.89	0.97	0.98

	RMSE w/o Conf 40	200	500	RMSE w/ Conf 40	200	500
0	10	5	0	5	5	0
50	50	50	50	40	45	40
90	65	75	80	40	55	70
98	65	75	80	65	70	80
	Corr w/o Conf			Corr w/ Conf		
0	50	65	75	0	0	0
50	50	70	70	40	55	40
90	50	70	70	40	55	65
98	50	70	70	40	55	65

Figure 14: Results for Baseline of 0%ile Confidence on training, improvement of the optimum over the 0%ile baseline, and optimum RMSE/Correlation values along with corresponding training percentiles for M5Rules. Training set n=40, 200, 500, testing conf percentile=0, 50, 90, 98. All testing set sizes=300.

MLP: At 0%ile Confidence for Training:

	RMSE w/o Conf			RMSE w/ Conf		
tst%ile\n	40	200	500	40	200	500
0	3.06	2.63	2.63	1.96	1.53	1.48
50	2.40	1.92	2.01	1.33	0.90	0.73
90	2.70	2.29	2.38	1.53	0.92	0.73
98	2.67	2.29	2.41	1.57	0.97	0.74
	Corr w/o Conf			Corr w/ Conf		
0	0.36	0.46	0.48	0.77	0.86	0.86
50	0.59	0.76	0.81	0.77	0.91	0.94
90	0.62	0.81	0.87	0.71	0.91	0.94
98	0.62	0.82	0.87	0.69	0.91	0.93

Improvement of Optimum:

	RMSE w/o Conf			RMSE w/ Conf		
	40	200	500	40	200	500
	0.24	0.09	0.15	0.01	0.04	0.07
	1.34	1.07	1.22	0.50	0.33	0.20
	1.94	1.87	2.00	0.82	0.50	0.37
	1.98	1.91	2.06	0.89	0.55	0.39
	Corr w/o Conf			Corr w/ Conf		
	0.07	0.04	0.03	0.00	0.01	0.01
	0.21	0.12	0.08	0.10	0.03	0.01
	0.24	0.15	0.10	0.17	0.05	0.03
	0.25	0.14	0.10	0.19	0.05	0.04

Optimum RMSE/Corr, and associated Training Confidence %ile

	RMSE w/o Conf			RMSE w/ Conf		
tst%ile\n	40	200	500	40	200	500
0	2.82	2.54	2.48	1.95	1.49	1.41
50	1.06	0.85	0.79	0.83	0.57	0.53
90	0.76	0.42	0.38	0.71	0.42	0.36
98	0.69	0.38	0.35	0.68	0.42	0.35
	Corr w/o Conf			Corr w/ Conf		
0	0.43	0.50	0.51	0.77	0.87	0.87
50	0.80	0.88	0.89	0.87	0.94	0.95
90	0.86	0.96	0.97	0.88	0.96	0.97
98	0.87	0.96	0.97	0.88	0.96	0.97

	RMSE w/o Conf			RMSE w/ Conf		
	40	200	500	40	200	500
	10	25	10	30	10	10
	70	55	65	55	50	60
	80	80	80	65	70	80
	80	80	80	65	70	85
	Corr w/o Conf			Corr w/ Conf		
	80	80	70	0	10	15
	80	80	70	60	50	35
	80	80	79	65	70	80
	80	80	75	75	70	70

Figure 15: Results for Baseline of 0%ile Confidence on training, and optimum RMSE/Correlation values along with corresponding training percentiles for MultiLayerPerceptron. Training set n=40, 200, 500, testing conf percentile=0, 50, 90, 98. All testing set sizes=300.

Relating to the Performance of Confidence-Prioritization

We have noted many patterns which have emerged from these data, but to narrow down those are most salient in regards to addressing our thesis statement:

- Using Confidence-Prioritization provides an improvement across all testing percentiles as compared to the baseline of 0% confidence.
- For many data sets and training algorithms, a narrower band of 'near-optimal' training percentiles, as indicated by a narrower peak on the graphs in Figure 8 through Figure 13, occurs, signifying a greater sensitivity of performance to data set selection, and therefore situations in which using Confidence-Prioritization will most likely provide the greatest gains over any particular ad-hoc data set selection.
- Observing the relative gains of optimum Confidence-Prioritization performance over the default of using the 0^{th} training percentile, as shown in Figure 14 and Figure 15, shows that relative gains of Confidence-Prioritization increase when (A) the test set is more confidence selective, or (B) the initial test set is smaller. This suggests situations in which Confidence-Prioritization provides the greatest gains: Among more sparse data sets, and training against more accurate/confident data.

Discussion of Simulation Study

Many of these results are just as would be expected: Having more data of equivalent confidence available yields models with higher accuracy, lower error, and higher correlation. Using too much low-confidence data can skew the model and increase error. Using too little high-confidence data can also over-simplify the model and increase error. Additionally, there is some range on this optimum; In some cases it can overlay a wide spread of percentiles, and in others it can be as simple as using as much data as possible – depending on how much confidence affects the data's accuracy for that data set.

Interesting Emergent Patterns

There are also some results that are not necessarily surprising, but interesting to note in regards to applying this methodology to other data sets. Specifically: (1) training with less-confident data, (2) the effects of using confidence as a parameter in training and testing, and (3) the differing effects of choice of training algorithm on the optimum confidence threshold.

Training with Less-Confident Data

One factor that often affects machine learning problems in the real world is that of imperfection in testing sets. Of course, this same limitation applies to training sets, which is the reason for this study to begin with. Often, testing sets are assumed to effectively be 'perfect,' much like training sets, but do, in reality, contain some inherent

error. Note how, as confidence in the testing set increases, the optimum confidence-percentile for training also increases. With the 50th percentile testing set, the optimum confidence percentile sits at around 55-65 for MLP and 45-55 for M5Rules. By the 98th percentile testing set, the optimum has jumped to around 75-80 for MLP and 65-80 for M5Rules.

Obviously it is ideal to have a test set that is as perfect as possible, as the 98th percentile test set represents. Oftentimes what is available in real world data sets is closer to the 90th or even 50th percentile test set here. However, as previously indicated, training against a test set with low confidence may result in a lower measured optimal training confidence threshold, as compared to training and testing against a test set with higher confidence. If one is limited to a lesser quality test set (i.e. one which would become excessively sparse if confined to a higher confidence threshold), the measured optimal training confidence threshold achieved against that low-confidence test set might be implied to be lower than a confidence threshold achieved by testing against a theoretical higher-quality (i.e. 'closer to ground truth') test set. A data miner could therefore choose to increase the measured optimum training confidence threshold (as computed against the low-quality test set), in order to generate a model in training that *may perform better against test sets that are higher quality than he has access to.*

Of course, there are many questions regarding this approach. If test data is sparse or low-confidence, training data will also likely be sparse or low-confidence, and

using a higher confidence threshold could quickly raise a sparseness problem in training as well. Also, this would be difficult to estimate quantitatively; exactly how much higher would the optimal training confidence threshold be against a higher-quality test set? This is likely to depend on multiple complicating factors, such as the training algorithms used and the nature of the data set itself. Also, it would not necessarily be a verifiable model, until higher-quality test sets became available. We only have preliminary results in this study to suggest that such a pattern exists between low- and high-quality test sets, but do not make any conclusions regarding it. It may be a point of future work to investigate whether this pattern exists consistently across data sets, and whether or not it can be used to build more optimal models.

Using Confidence as a Parameter in Training

Another notable pattern in the results is that of the effect of using confidence as a parameter in training and testing. In theory doing so should enable a learning algorithm to account for confidence or the lack thereof in a more graded fashion than binary inclusion/exclusion, in some cases possibly weighting confident data records more heavily for training. What we see in these results is that including confidence does improve performance for confidence percentiles that are lower than the optimum, but neither affects the optimum confidence percentile nor the performance of models trained near or above it. It may be that this is because having confidence available as a variable allows training algorithms to build models that treat low-confidence points

differently than high-confidence points – and when a high-confidence test set doesn't contain any low-confidence values that need to be tested, the trained model will treat them similarly, whether or not that model was trained on low-and-high-, or *just* high-confidence data. Supporting this hypothesis, the results show that in the higher percentiles, when it comes down to a narrow band of high-quality data, you see the same decrease in performance on both data sets, with and without confidence as a variable. This is because at that point the relative confidence of data points is not as useful, and the limitation for both sets is just data sparseness.

Based on these results, while it may be helpful to include confidence as a parameter in testing and training if you ***weren't*** to choose an optimum confidence percentile, it might not provide much benefit for situations where you do. Future work would be needed to investigate this more thoroughly. The purpose these confidence measures are being gathered in the first place is to choose an optimum confidence percentile and acquire as much of an accurate and representative training set as possible.

Choice of Training Algorithm

Another item of interest to mention here is that the choice of algorithm affects the optimum confidence threshold, as well as overall optimum performance. Performance of resulting models for both M5Rules and MLP did increase as the size n of the original training set increased – however, there were differences in their results. M5Rules was

able to construct an optimum model using higher-confidence-percentile data as n increased, while MLP maintained the same optimum confidence threshold regardless of n. This may be a manifestation of M5Rules being able to train on less data when more clearly observable patterns are available, whereas MLP may prefer as much data as possible in data-sparse situations.

Using a low n (=40) and the less-confident 50th percentile testing set, MLP performed worse (at an optimum confidence percentile of 70) than M5Rules (at 50), but better on the higher quality 98th percentile test set (at 80) than M5Rules (at 65). However, with more data available (n=200, 500) M5Rules performed better than MLP on all test sets. This may be due to MLP overfitting on higher n, or it may be that with M5Rules, separate rules are more effective at deriving a 'simple' pattern among high-confidence training data than by using the continuous inputs of MLP. Given this sampling, M5Rules may be a good choice of algorithm when using Confidence-Prioritization with an abundant variety of high- and low-confidence data, while MLP may be a better choice for more sparse data sets.

As Related to Performance of Confidence-Prioritization

A final result we discuss here is that, as shown in Figure 14 and Figure 15, using Confidence-Prioritization provides better results than the baseline of using all data available (i.e. the 0th percentile confidence). This is seen across all testing confidence percentiles, though unsurprisingly the effect is more pronounced against the cleaner,

more representative high-confidence-percentile test sets. The effect is *less* pronounced when training with confidence as an input variable, which is unsurprising given that having confidence available in training can mediate for the negative effects of low-confidence data in training.

Peaks

Perhaps the strongest evidence that Confidence-Prioritization can help select optimal data sets under conditions of known confidence is by observing the *shape of the performance curves* in Figure 12. In all of these curves, either end provides sub-optimal performance, while there is a gradual, generally convex curve towards the optimum in the middle. If the data were more random or less consistent, such a clear pattern would not be present; multiple local peaks would occur with intermittent drops. As it is, Confidence-Prioritization is shown to converge upon an absolute maximum correlation (or minimum RMSE) by following this curve to that optimum.

As stated earlier, there is some range on optimum performance, in that it may lie over a wide or narrow range of training percentiles. In the case of a wide margin, where there is much more allowance in data selection to achieve optimal results, the peak will be relatively flat. For example, this is seen for correlation on the n=500 data set. In such a situation it would be quite feasible to use an ad-hoc selection method that would work just as well as Confidence-Prioritization with a fair margin of error.

However, there are cases in which that a peak of performance is had among a narrow range of confidence training percentiles. For example, for n=40, that peak drops off much more quickly; quite significant drops are made in correlation for moving too many percentile from the optimum confidence threshold. Steeper peaks such as this indicate situations ***where performance is more sensitive to data selection***. These are the same situations in which using Confidence-Prioritization will yield greater returns because using an ad-hoc data selection method will ***very likely select a data set that falls off of that peak***, resulting in sub-optimal performance. Confidence-Prioritization will help the user ensure that they have selected and optimal data set for training.

The fact that these peaks show up in our simulated data suggest that they likely occur in other, real-world data sets. We begin to investigate this possibility in Chapter VI.

Conclusion

This study shows that data sets which can be assigned confidence values that ***are known to correlate*** to how noisy or inaccurate individual data points are can be more optimally processed in training and testing by using our Confidence-Prioritization approach, using data which is *ambiguously* accurately or inaccurately representative of the underlying system.

We have also seen that under all conditions, Confidence-Prioritization will, by nature, not harm performance as compared to default, all-inclusive data selection, and

in most cases it will help. Under many conditions, such as sparse data or higher-accuracy test sets, Confidence-Prioritization provides significant improvements.

Narrow peaks of optimal performance across training percentiles also suggests that there are many data sets for which Confidence-Prioritization can nail an optimal data set for training ***which would otherwise be difficult to identify with a single ad-hoc selection process***.

The next step in our process is to show that this pattern will still hold true with real-world data, where Confidence is not known to correspond directly to actual data accuracy.

CHAPTER VI: APPLICATION TO ENVIRONMENTAL SCIENCE

Earlier, in Chapters I and III, we discussed the need for novel Data Mining approaches in Environmental science, a particular example being in the field of Hydrology for estimating Diel Signals. We conduct the study in this Chapter to investigate diel signal patterns in hydrology, and utilize our methodology (as discussed in Chapter IV) on this real world, Environmental Science data.

The simulation study conducted and discussed in Chapter V helps to show that, provided that confidence *does* correspond to more accurate data, using Confidence-Prioritization *does* help to identify optimal data sets for training with ambiguously (in)accurate data. In this chapter we apply the same methodology to address the next question, of '*In a data set with human-estimated confidence values (i.e.* ***where the association of high confidence with more accurate data is uncertain****, as opposed to the simulation study), does using Confidence-Prioritization still help identify optimal data sets for training?*'.

Problem Background

Streams are often known to exhibit diel fluctuations, in which streamflow oscillates on a 24-hour cycle. Streamflow diel fluctuations are an informative indicator of environmental processes. This pattern differs according to season, from watershed to watershed, and according to multiple other factors. While the existence of diel

streamflows is well-established, research is still being conducted to answer questions regarding their cause and how different factors affect them. Some simulation studies, such as those focusing on groundwater flow [11, 20], and some relatively narrow case studies [6, 7, 10, 19, 23] have already been conducted, but no studies have utilized any Data Mining approach to elucidate patterns among larger data sets of multiple watersheds.

The HJ Andrews Experimental Forest and dozens of other sites throughout Northern America have been collecting environmental data continuously over the past several decades as part of its LTER (Long-Term Environmental Research) mission. In this study, we will gather a relatively large amount of LTER data regarding streamflow and connected environmental factors, apply our Confidence-Prioritization methodology, and recommend the resulting model for estimating diel signal strength as a potential tool in Hydrology for further investigating diel signals.

Problem Statement

Predict the volume of water lost for a day to diel streamflow fluctuation in a particular watershed given a set of daily data for environmental factors, specifically: average daily solar radiation, air temperature, precipitation, and streamflow.

Data Acquisition

We've gathered data for 9 watersheds in the HJ Andrews Experimental Forest in central Oregon, across 9 years, for a total of over 29,000 watershed-days. The raw data collected is that of 15-minute interval values for:

- Streamflow volume
- Air temperature
- Precipitation
- Solar radiation

Also collected was some per-watershed data, such as land area, minimum and maximum elevation, slope, and channel length. This data was not used in training, because individual models were generated for each watershed. However, we do discuss some per-watershed data in the results and discussion section.

All data was acquired from the HJ Andrews Experimental Forest data download site (http://andrewsforest.oregonstate.edu/lter/), itself a part of the US Long Term Environmental Research (LTER) Program [28]. Other sources for similar streamflow data do exist within and outside of the US LTER program, though they differ in type and resolution of data provided, as well as climate and geography of associated stream networks. Websites for various (LTER) locations provide such information, as does the U.S. Geological Survey (USGS). For this study, for the sake of consistency of climatic

and geographic conditions for our data, and because we felt the HJ Andrews LTER site provided plenty of data to demonstrate the applicability of our Data Mining Approach, we only used data from the HJ Andrews LTER site.

Figure 16 and Figure 17 show the HJ Andrews forest, along with a closer map of a few of its watersheds.

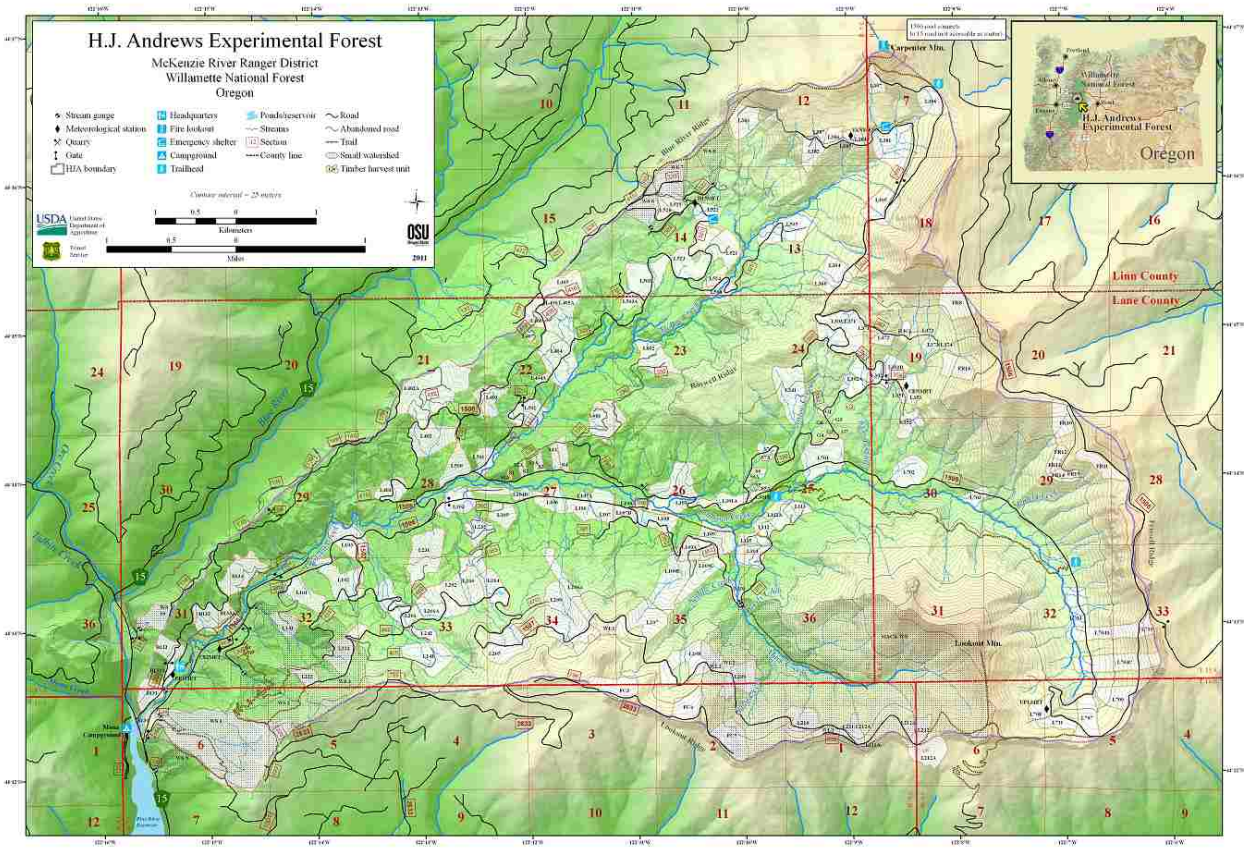

Figure 16: HJ Andrews Forest. (http://andrewsforest.oregonstate.edu/lter/about/site/map.cfm)

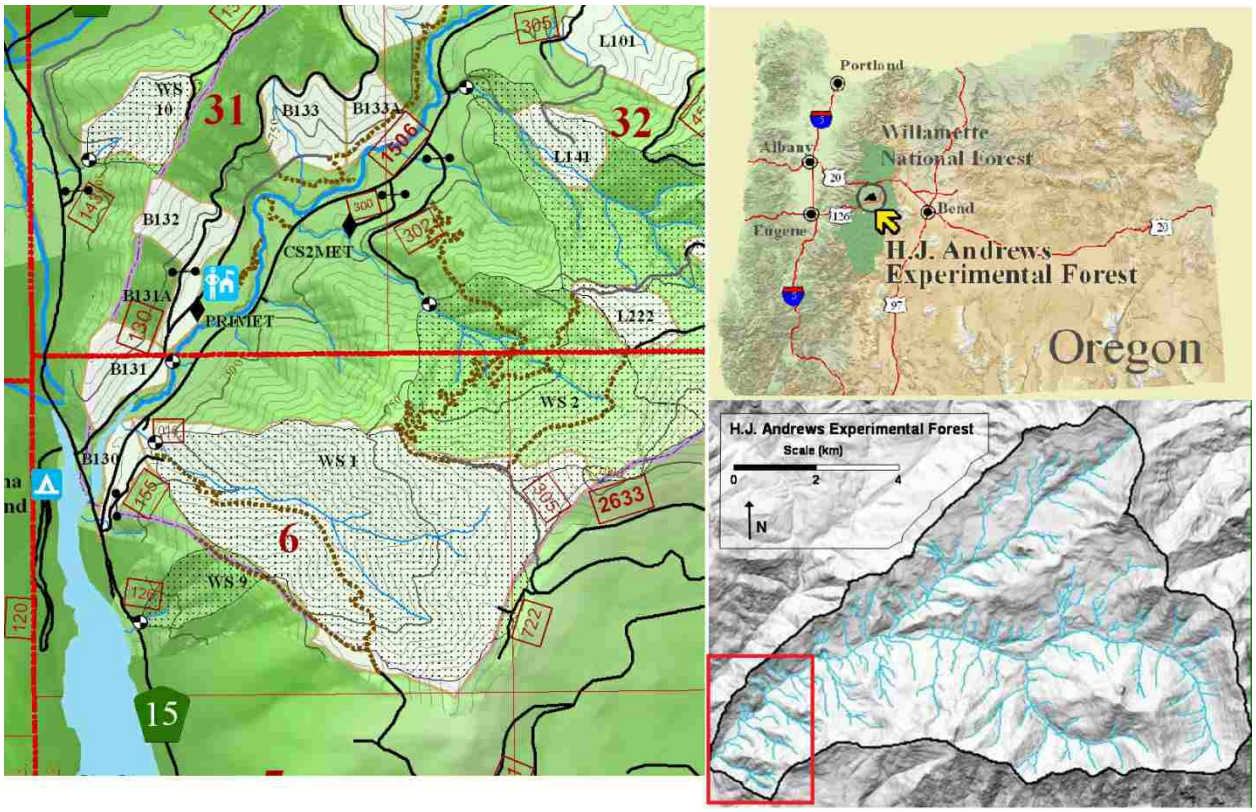

Figure 17: Upper Right: Zoomed out view of Oregon and the location of the HJ Andrews Forest. Lower Right: topographical view of the HJ Andrews forest. Red Square denotes zoomed portion. Left: Zoomed portion, showing WS 1 (below center), WS 9 (Southwest of WS 1), WS 2 (NW of WS 1), and WS 10 (upper left). (http://andrewsforest.oregonstate.edu/lter/about/site/map.cfm and http://water.engr.psu.edu/gooseff/network_hz_proj.html)

Data Processing and Confidence Determination

In this section we discuss the data processing stage, which is synonymous with the data collection phase (as discussed in Chapters IV and V on our methodology and simulation study) because it prepares individual data records as they will be when run through learning algorithms.

Besides preparing individual daily records for training and testing, in this data collection phase we also assign a confidence value to each day based on the clarity of and trust in the associated diel signal for that day.

After processing the 15-min-interval data provided to us from the HJA LTER site into daily values and assigning confidence values to the associated diel signal for each day, we will continue into the training and testing stage, wherein learning algorithms are run and models are generated.

Daily Summary Values

Daily summary values were made for precipitation, temperature, solar radiation, and streamflow by averaging (for temperature, solar radiation, and streamflow) or summing (for precipitation) the instantaneous measured values across all 15-minute periods for that day, from midnight to midnight.

Diel Signal Extraction

There are previously-established methods that have been used to extract diel signals from hourly (or so) streamflow data [8]. The standard technique (which we use here) is easily visualized on a streamflow-volume vs. time graph, as shown in Figure 18. To identify the maximum-flow point for one day, a line between the maximum flow points for two adjacent days is made, and the area below that line and above the measured instantaneous streamflow volume represents water lost that day to diel fluctuation, otherwise known as diel signal strength.

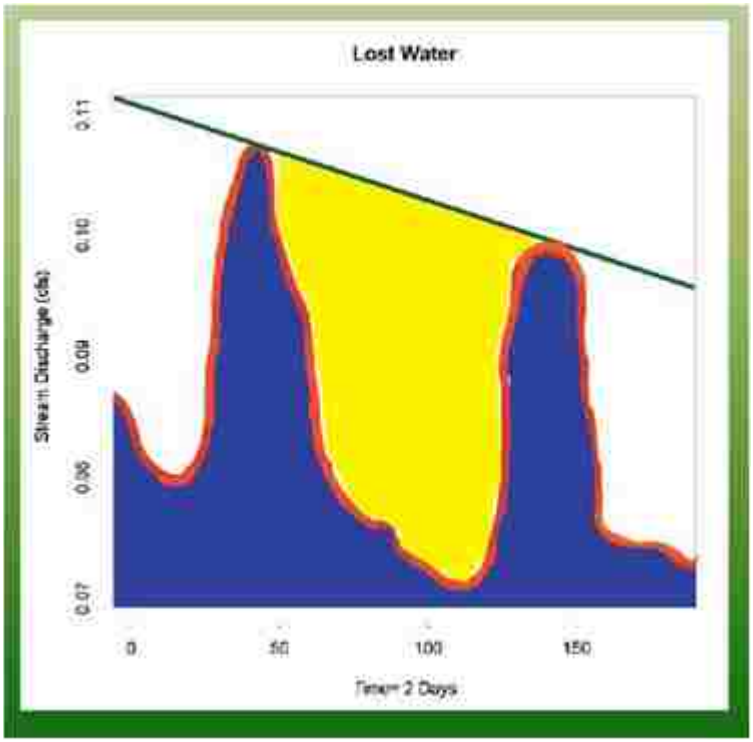

Figure 18: Example of water loss estimation. Red line represents instantaneous measured streamflow (15-min resolution). Blue area represents implied streamflow output over time. Area above streamflow curve and below line connecting the two days' maximum flow points represents water lost that day to the diel fluctuation.

The calculation to measure water loss day to day is fairly straightforward, however, various circumstances arise which introduce some noise and other complications.

Sources of complication

Almost any data source may be considered as having noise. Besides humans sometimes inadvertently misusing equipment or measures, measured data sets rely on

equipment with limited accuracy. And beyond human error and equipment error, in environmental systems such as this one, there is a relatively *high* amount of inherent noise, due to these factors as well as the expansively complex nature of ecology. Soil and bedrock topology, occasional shifts in streamflow obstruction, temperature-related changes in water viscosity, and numerous other factors could affect signal output.

Additionally, numeric and nominal data record fields often fail to capture the full complexity of the data at hand. In this situation, for example, we are creating a metadata value that summarizes 96 points (24 hrs of 15 min data) for what we can best interpret to be water loss. However, the shape of that curve, the daily peak volume difference, and notably the noisiness or reliability of that curve is not accounted for in that single water loss value. By adding a confidence measure, we include additional information which *does* account for some of this information, namely data noisiness or reliability.

Flow Restoration

When speaking in terms of streamflow diel signals, typically a stream has what you might call a 'ground' state, a state in which streamflow follows a gradual reduction trend as the watershed drains, and doesn't show any 24-hr periodic fluctuation. This ground state is represented by the line between maximum flow points, as shown in Figure 18. Diel signals are the difference between that ground state and actual measured flow.

In some cases, however, particularly late summer, a stream fails to return to its 'ground' state before the following days' water loss begins. This means that the maximum measured points for that day misrepresent the ground state, and thus underestimate water loss.

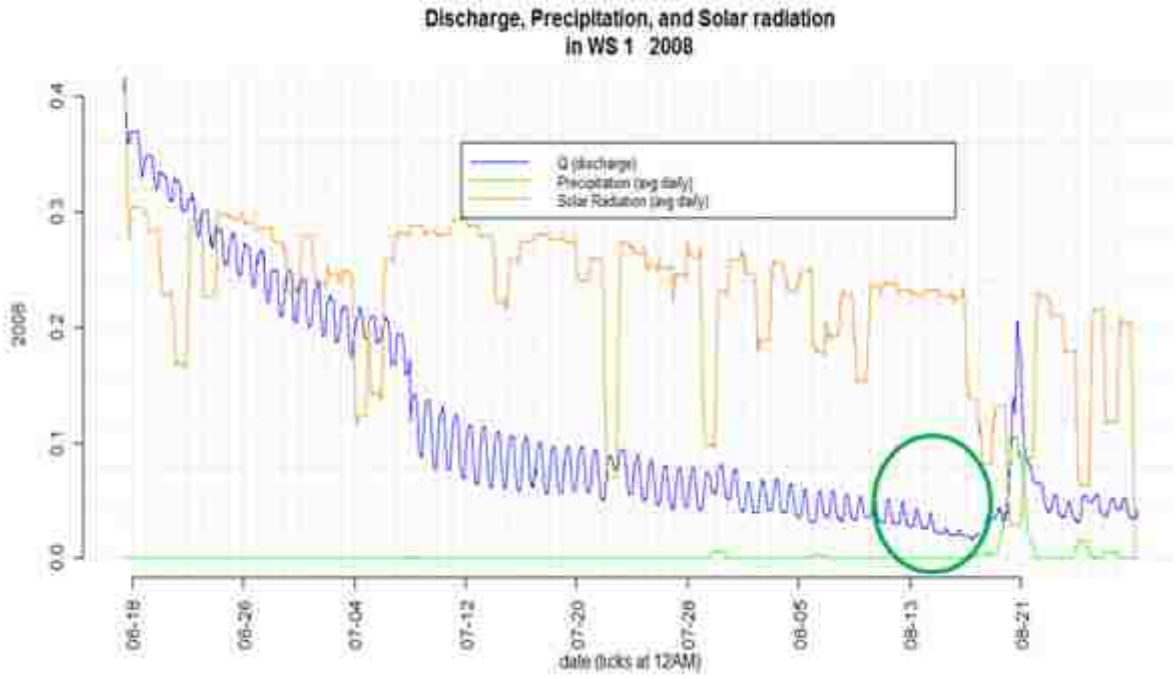

Figure 19: Example of incomplete flow restoration. Earlier days in this summer show a peak that is smooth, representing a transition back to maximum streamflow that slowly edges back into its natural state. Later, drier summer days (green circle) show sharper peaks, which are cut short again by the following days' diel streamflow fluctuation.

Signals of differing source

Another source of complication is that there are multiple types of diel signals [8]. In these watersheds, signals exist for both summer groundwater flow and spring snowmelt. They both operate on a 24-hr cycle, but form under different conditions, and

result in signals of generally different magnitude (snowmelt signals often being much larger). Diel signals have also been known to occur as a result of rainfall in some regions, where precipitation itself can be patterned with a daily bias [8].

Seasonal Bias

Like how there are multiple types of diel streamflow signals that occur in watersheds, there are also tendencies for some types of signals to occur in some parts of the year and not others. Spring often manifests snowmelt signals, and summer more regular signals from groundwater loss. When potentially similar atmospheric conditions occur in other seasons (eg. An exceptionally warm and sunny winter day or week), they may not exhibit the same intensity of diel signal.

Inaccurate 'ground' state flow estimation

While the water loss estimation method discussed in [8] is reasonable and widely used, the inherent noise of Environmental Systems makes it difficult to employ any perfect estimation method. In truth, a 'ground' flow state is only estimated by the max-flow points between two adjacent days; if the actual ground flow state were to have increased or decreased during that day for whatever reason (eg. streamflow obstruction, or a minor rainfall), that change in flow would not be accounted for in predicted ground flow. Thus, when estimating water loss, it would result in additional inaccuracy in water loss estimation.

Differing max-flow times each day

A critical piece of the water loss estimation method we are using here is to establish the time-of-day for maximum flow, for each day. Typically, in a series of days exhibiting a very clear signal, these max-flow times occur within about the same hour or two on each day. However, the max-flow time can also shift to several hours earlier or later in the day depending on how dry or hot or sunny the watershed becomes. It can also differ based on the size or other properties of groundwater flow in each watershed. Precipitation and other confounding factors can also tweak max-flow times for days with otherwise legitimate diel streamflow signals.

Max-flow times on days without clear signals are more often scattered from day to day, but not necessarily are. Also, for some signals, such as those being exhibited in watersheds during a drastic decrease in average daily streamflow volume, there may not be a 'peak' maximum daily flow time, rather, instantaneous streamflow decreases continually at alternately greater and lesser rates. Figure 20 shows an example of this.

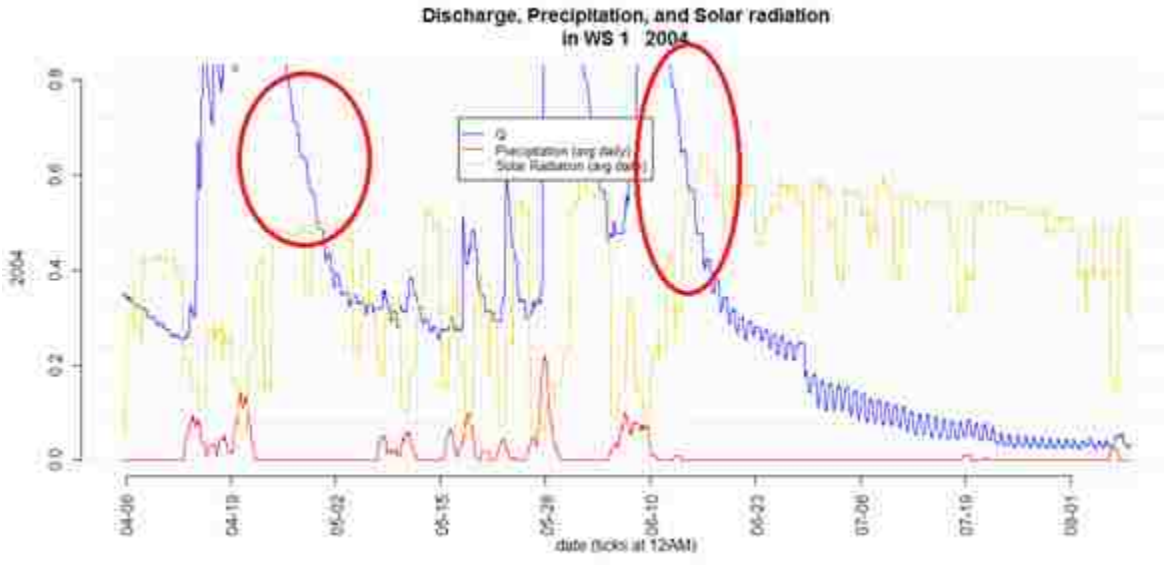

Figure 20: Example of legitimate diel signals occurring during extreme streamflow reduction. In mid-June (under the right circle) there are only rough peaks, and in late April and early May (left circle) there are no maximum peaks in streamflow. In both of these cases, legitimate signals exist, but the standard max-peak-to-max-peak line estimation method fails to represent them properly.

As shown in Figure 20, heavy streamflow loss can result in skewing (or entirely missing) the peak streamflow in regards to diel fluctuations, because as the watershed drains, the overall watershed streamflow is decreasing at a significantly greater rate. In order to derive more accurate streamflow maxima and minima, each day is treated as residual streamflow from a 24-hr windowed average. This is a method we approached independently; by comparison, [8] uses a curve-estimation model to connect daily maximums and form a 'base flow' estimate, but does not show examples of such estimations for diel signals during significant baseflow recession as we are discussing here. We've found this 24-hr windowing method to be effective for our purposes.

So the effective streamflow for 6am on a given day is the streamflow measured at 6am minus the average streamflow from 6pm the previous day to 6pm that evening. Figure 21 shows an example of this effect.

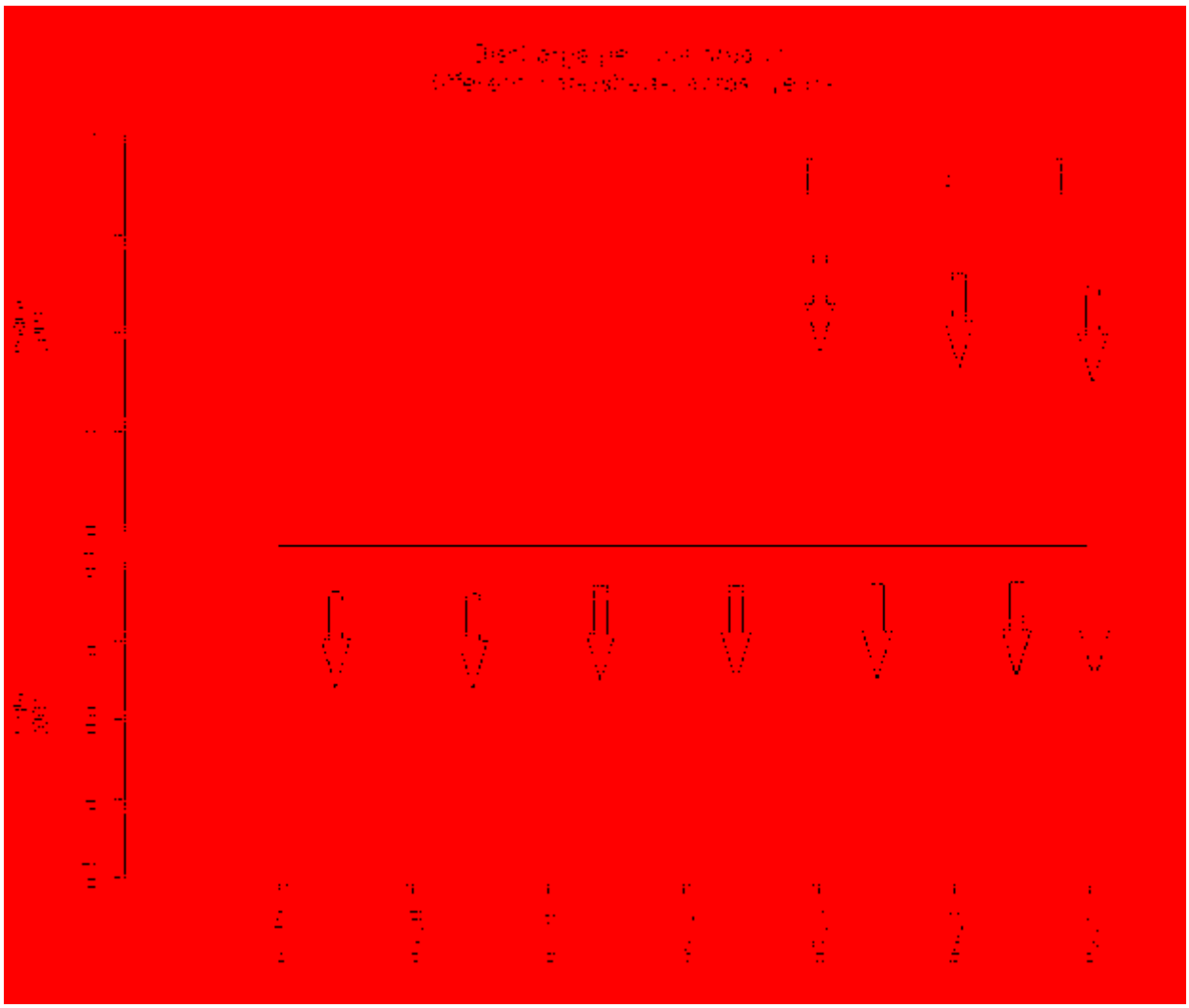

Figure 21: An example of extracting flow maximums from original streamflow (top half) and residuals off the daily average (bottom half) during a week in May 2001 for WS 1. Black arrows indicate a local maximum for each day, gray arrows indicate a maximum flow for that given day that is *not* local, i.e. for streamflow decreasing continuously throughout the day this would be streamflow at midnight.

Of course, this results in clearer peaks, and thus (a) more accurate water-loss estimation and (b) more precise detection of actual diel signals, which exhibit a regular

timing pattern wherein the maximum and minimum streamflow occur in about the same hour of consecutive days.

Confidence

We compute the confidence value for each data record (i.e. each day) largely off of the complicating factors mentioned above. A clearer, more measureable signal corresponds to a higher confidence value. Complicating factors which are not accounted for in confidence calculation are discussed in the following subsection (on *Seasonal Separation*).

We construct a set of rules to attribute 'confidence points' to each data record according to each complicating factor. By having observable, calculable properties of clearer signals, a given day's signal gains higher confidence.

Confidence Calculation

In order to calculate confidence for each day in each watershed, we consider three main factors: the **timing** of the maximum and minimum streamflow within each day, the **variance** of those maximum and minimum streamflow periods, and **adjacent days'** signal validities.

To attribute greater signal confidence to days which exhibit the regular **timing** indicative of clearer diel signals, a penalty is applied to each day if its time-of-day of maximum or minimum flow is more than a few hours from the time-of-day of maximum or minimum flow (respectively) in either adjacent day. If it is within

tolerance, no penalty is applied, and outside of that confidence is reduced on a sliding scale, greater reductions being caused by being farther off timing.

Timing alone is not always clarifying, however, as some days may have regular maxima and minima, but have diel fluctuations which are small compared to the resolution of measurement, or peaks which are otherwise not vividly discrete. Thus a measure of **variance** of the distribution of highest and lowest points is made for each day by taking the highest and lowest 20% of streamflow measurements for that day and calculating the variance of the time-of-day among each set. Days which have under a threshold of variance indicate a tightly clustered maximum (or minimum) and receive no penalty. The farther above that threshold the variance is, the greater the penalty.

Finally, the diel signal confidence score for any given day is affected by that of **adjacent days**. Diel signals *can* occur individually or in very short sequences, but most generally occur in series. Adjacent days having a higher signal confidence causes that particular day's signal confidence to increase.

The final R script used to calculate confidence for each day is shown in Figure 22 and Figure 23. Constants used are determined empirically by the data collector to create a representative heuristic for confidence.

```
foreach (k in days)
{
  #cur_day_flux: Residual streamflow from 24-hr avg window (15min * 96)
  #minmax_time: The time (i.e. index in 15min increments) of min/max flow
  #min/max_avgd_time: The average time of the min/max 20% values

  #set constants (empirically determined for the RULE-BASED CONFIDENCE
  # DETERMINATION process, empirically determined by the data collector)
  top_x_for_timing <- .20 #top 20% used for measuring variance of max/min
  weight_max_timing <- 30 #30 points based on the timing of flow maximum
  weight_min_timing <- 20 #20 points based on the timing of flow maximum
  weight_max_avgd_timing <- 30 #30 points for how tight the avgd maxs are
  weight_min_avgd_timing <- 20 #20 points for how tight the avgd maxs are

  #Adding conf from days on either side - signals tend to be sequential
  weight_of_prev_day <- .15 #ratio of points from prev day
  weight_of_prev_prev_day <- .07 #ratio of points for day before last
  weight_of_next_day <- .15 #ratio of points for next day
  weight_of_next_next_day <- .10 #ratio of points for day after next

  #1 hr shift either way in timing on max can get full credit -
  # Can still get partial credit past that. 4 15-minute increments = 1 hr
  allowed_var_from_prev_day <- 4

  # confidence is kind of a point system - Start with 0, add up
  daily_signal_confidence[k] <- 0 #for signal confidence for each day

  #The closer one day's max/min flow times are to the other's, the higher
  # score.  ecdf(x)() returns a percentile of the given value among x.
  max_percentile <- ecdf(cur_day_flux)(cur_day_flux[prev_day_max_time])
  min_percentile <- ecdf(cur_day_flux)(cur_day_flux[prev_day_min_time])

  #The closer the avg times of max/mins from one day to the next are....
  max_avgd_time_dif <- abs(prev_day_max_avgd_time - cur_day_max_avgd_time)
  min_avgd_time_dif <- abs(prev_day_min_avgd_time - cur_day_min_avgd_time)

  ratio_on_max_timing_score <- (max_percentile - 1 + .2 /
(1.1-max_percentile))
  if (ratio_on_max_timing_score > 1)
    ratio_on_max_timing_score <- 1
  ratio_on_min_timing_score <- (-min_percentile + .2 /
(.1+min_percentile))
  if (ratio_on_min_timing_score > 1)
    ratio_on_min_timing_score <- 1

  ratio_on_max_avgd_timing_score <- (1 + allowed_var_from_prev_day)/
(1 + max_avgd_time_dif)
  if (ratio_on_max_avgd_timing_score > 1)
    ratio_on_max_avgd_timing_score <- 1
  #If lots of var...  cut it off.  Not a clear minimum for that day
  if (prev_day_max_avgd_var > 100)
    ratio_on_max_avgd_timing_score <- 0
(CONTINUED ON NEXT PAGE)
```

Figure 22: R script for Confidence Determination (more description on next page).

```
  ratio_on_min_avgd_timing_score <-
(1 + allowed_var_from_prev_day)/(1 + min_avgd_time_dif)
  if (ratio_on_min_avgd_timing_score > 1)
    ratio_on_min_avgd_timing_score <- 1
  #If lots of var,  cut it off.  Not a clear minimum for that day
  if (prev_day_min_avgd_var > 100)
    ratio_on_min_avgd_timing_score <- 0

  daily_signal_confidence[k] <- daily_signal_confidence[k] +
                weight_max_timing * ratio_on_max_timing_score +
                weight_max_avgd_timing * ratio_on_max_avgd_timing_score
  if (k>1)
  { daily_signal_confidence[k-1] <- daily_signal_confidence[k-1] +
                  weight_max_timing * ratio_on_max_timing_score +
                  weight_max_avgd_timing * ratio_on_max_avgd_timing_score
  }
  daily_signal_confidence[k] <- daily_signal_confidence[k] +
                weight_min_timing * ratio_on_min_timing_score +
                weight_min_avgd_timing * ratio_on_min_avgd_timing_score
  if (k>1)
  { daily_signal_confidence[k-1] <- daily_signal_confidence[k-1] +
                  weight_min_timing * ratio_on_min_timing_score +
                  weight_min_avgd_timing * ratio_on_min_avgd_timing_score
  }

  #Signal is more valid if previous/next day(s) is(are) valid
  #2-day window (on either side) smoothes out confidence
  if (k>1)
  { to_add_to_cur_day <- ceiling(daily_signal_confidence[k-1] *
weight_of_prev_day)
    to_add_to_prev_day <- ceiling(daily_signal_confidence[k] *
weight_of_next_day)

    if (k>2)
    { to_add_to_prev_prev_day <- ceiling(daily_signal_confidence[k] *
weight_of_next_next_day)
      to_add_to_cur_day <- to_add_to_cur_day +
ceiling(daily_signal_confidence[k-2]*weight_of_prev_prev_day)
      daily_signal_confidence[k-2] <- daily_signal_confidence[k-2] +
to_add_to_prev_prev_day
    }
    daily_signal_confidence[k] <- daily_signal_confidence[k] +
to_add_to_cur_day
    daily_signal_confidence[k-1] <- daily_signal_confidence[k-1] +
to_add_to_prev_day
  }
}
```

Figure 23: Calculations for confidence for individual days' measurements of water loss. Rules and constants were determined empirically by the data collector as a representative heuristic for confidence.

Seasonal Separation

Streamflow diel signals can be caused by multiple distinct sources [8], and there is a wide amount of variation in diel signal strength, timing, and season depending on which type of signal is being expressed. In the spring, very large surges can occur in high elevations from snowmelt over short periods, in tropical regions diurnal precipitation can cause another kind of signal, and in the summer evapotranspiration can drive another diel signal – The type that is most commonly expressed in Pacific Northwestern watersheds, and upon which there is more interest currently [6, 8, 19, 29, 30]. Since that signal, as opposed to snowmelt or tropical rain signals, is almost always expressed in the summer season and very rarely in other seasons, we've narrowed the data space down to the months of May, June, July, August, and September. We address the question of what affects summertime diel signals, as opposed to snowmelt or other diel signals – for which we have less data and less interest in hydrology to investigate.

Training & Testing

At this point, the daily summary data has been collected – average daily solar radiation, temperature, precipitation, water loss, and diel signal confidence, an example of all of which is shown along with their relations to one another in Figure 24 – is assembled for use in training and testing. The purpose of the generated models is to estimate how much water loss any particular watershed will experience on any

particular day, provided environmental factors for that day: temperature, precipitation, solar radiation, and average daily streamflow.

Among our 9 years of data, we do leave-one-year-out cross-validation: Each year's data is used as a testing set against a model trained on the other 8 years. This ensures our models are a more robust against year-to-year variation, as compared to a random cross-validation.

A few interesting features of this environmental data set call for special attention in testing. As discussed above, due to seasonal bias we separate summer diel signals from spring snowmelt and other relatively minor signals by selecting the months of May through September. This decreases the amount of noise present as well as the problem space, in that we are not addressing snowmelt and other minor signals. Such signals occur much less frequently anyways, and would be more difficult to account for due to data scarcity.

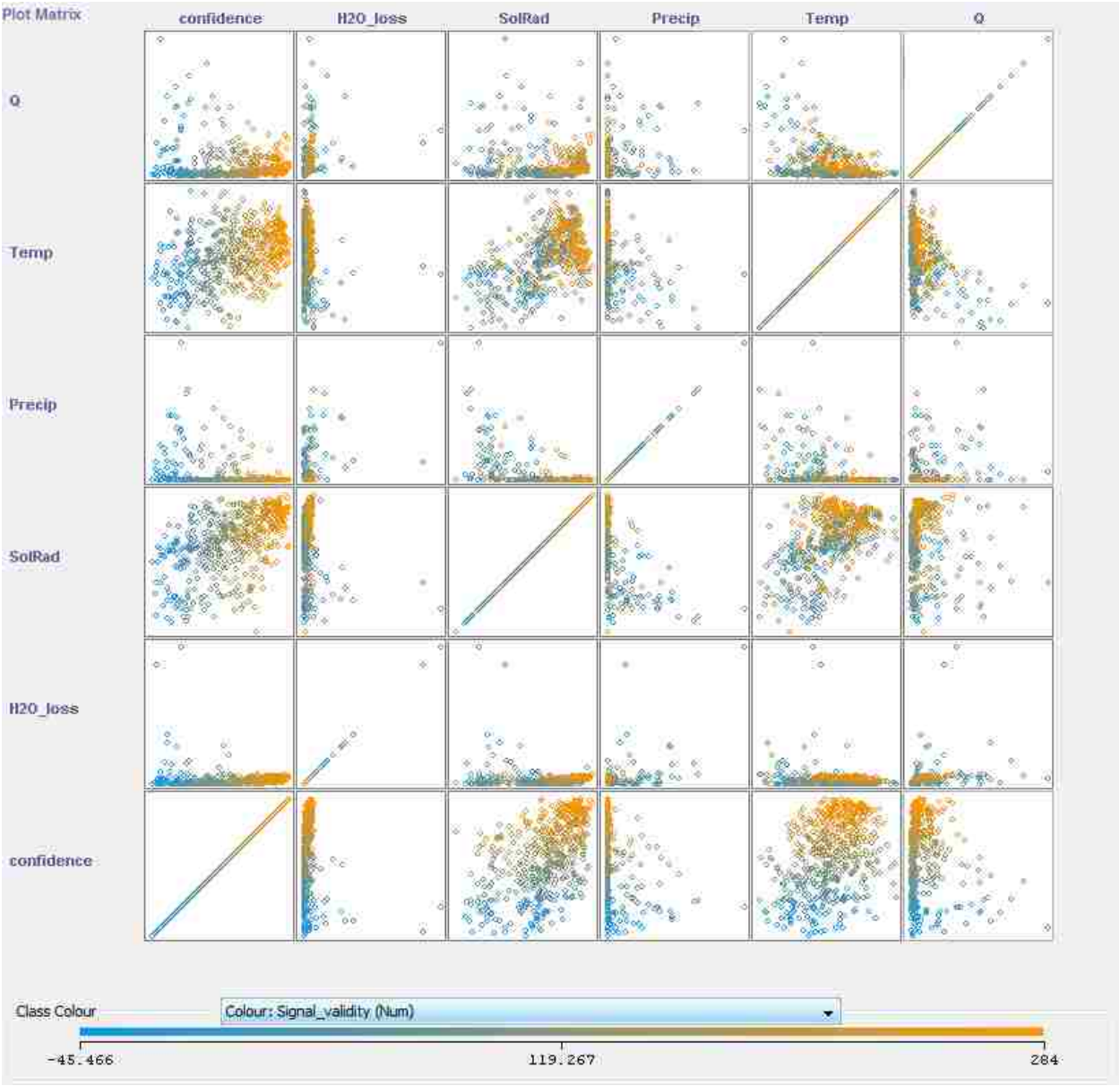

Figure 24: Daily summary data collected for Watershed 1 from May through September for 2002 through 2009. Q is streamflow. Q, Temp(erature), and Sol(ar)Rad(iation) are all daily averages, Precip(itation) is a daily sum, and H2O_loss is measured water loss.

Additionally, since each watershed does vary from the others in its characteristics, a model generated from one watershed may accurately estimate signals for that same watershed, but be completely unrepresentative of other watersheds.

Slope, vegetation type and density, spatial watershed topology and soil profile are all complicating factors for which we cannot entirely account here with our generated prediction models, due to a low number (statistically speaking) of watersheds and incomplete data on soil profile and vegetation. To account for this, we handle watersheds individually, training models and testing using only days for that particular watershed.

While our models do not have sufficient per-watershed data to empirically derive many inferences regarding the effects of a watersheds' physical characteristics on its expressed diel signals, there are environmental reasons to expect some effects. A steeper topological slope in a given watershed would suggest faster groundwater propagation and perhaps implicitly, clearer (or in some cases stronger) diel signals [6]. Different types of trees pull varying amounts of water, per-tree and per-square-meter. In particular, younger trees are sometimes coupled more strongly to diel signals [10] and may implicitly pull more water. Also, larger watersheds have farther for water to travel before being measured at the measuring weir, as well as a greater surface area, which can both serve to attenuate diel signals, decreasing their signal clarity and perhaps calculated water loss as well. Our observations of watersheds are consistent with these patterns: Watershed 1 in particular is small, has a steep slope, and young stands of trees, and exhibits particularly clear diel signals as compared to the other watersheds, as well as typically higher daily water loss than most.

Learning Algorithms

We use three different learning algorithms to generate diel signal prediction models: **M5Rules**, **MultiLayerPerceptron**, and **KStar**. We chose M5Rules and KStar because they are fairly good at separating different *types* of instances for training and testing, as opposed to a more linear regression. M5Rules Uses M5 to build a model tree and then constructs each 'leaf' into a rule (http://wiki.pentaho.com/display/DATAMINING/M5Rules). This can be particularly helpful in Environmental data, which sometimes has multiple mechanisms and pathways to account for different environmental phenomena. Separating different 'types' of data points for effectively training multiple 'sub-models', as M5Rules does, can therefore be an effective strategy. KStar is an instance-based classifier that uses an entropy-based distance function, with an extension to handle numerical data [31]. Using instance-based 'classification' here is another strategy that could be effective for environmental data with complex patterns that simpler regressions would have difficulty capturing. MultiLayerPerceptron relies on empirically trained 'neurons' which reduce estimation error over several iterations. MultiLayerPerceptron is also known to build models which extract difficult or complex patterns, and are more difficult to decipher 'meaning' or patterns out of by a human than, say, a rule-based model. We include MultiLayerPerceptron here as a generally effective learning algorithm, and as a sanity check for the other two.

Testing Runs

Each of the three algorithms mentioned above was used on a testing run of each individual watershed. Within each watershed's testing run, Training percentiles were made every 10 percentile points, and percentile thresholds for test sets (i.e. the one-off year) were made at 0, 50, 90, and 95 percentile. Among all of those runs, results for each training-percentile/testing-percentile pair were averaged across all nine years of the leave-one-year-out cross-validated results.

Control Condition

The selection of months as discussed above, accounting for seasonal bias and narrowing signals to the range of summer months, constitutes a realistic data collection. In the data collection process, the summer months as selected were shown to exhibit a range of intensity of summertime diel signals, and not seen to exhibit characteristics of alternative diel signals (eg. Inverted phasing indicative of snowmelt diel signals, etc.). Some variation was seen to be had in the clarity of diel signal from day to day – the same variation that led to the inception of Confidence-Prioritization and this project – but selecting those months constituted already a fair assessment of selection for days pertinent to streamflow diel signals.

Therefore we use the 0^{th} percentile data set for training as a control, as we do in the Simulation Study, to compare our Confidence-Prioritization results against.

Results of Environmental Science Study

We found that among all of the watersheds, WS1 and WS10 provided the most interesting results. Other watersheds, for the most part, exhibit much lower correlation (around .5), and more random patterns along their training-percentile curves. This is not surprising, however, as WS1 and WS10 have particularly high signal-to-flow ratios – which is itself not necessarily surprising. Both watersheds have particularly high slopes, and WS1 has a young stand of trees (which consume more water per hectare), while WS10 also has by far the highest ratio of channel length to area. All of these factors would promote stronger and clearer diel signals. With clearer, more representative and higher quality data available, we could expect to more consistently see that peak performance point somewhere along the training-percentile curve, as discussed in Chapter V. Here, we focus on WS1 and WS10, particularly WS1. Complete results can be found in the appendix. Figure 25, Figure 26, and Figure 27 summarize results from Watershed 10. Figure 28, Figure 29, and Figure 30 summarize results from Watershed 1.

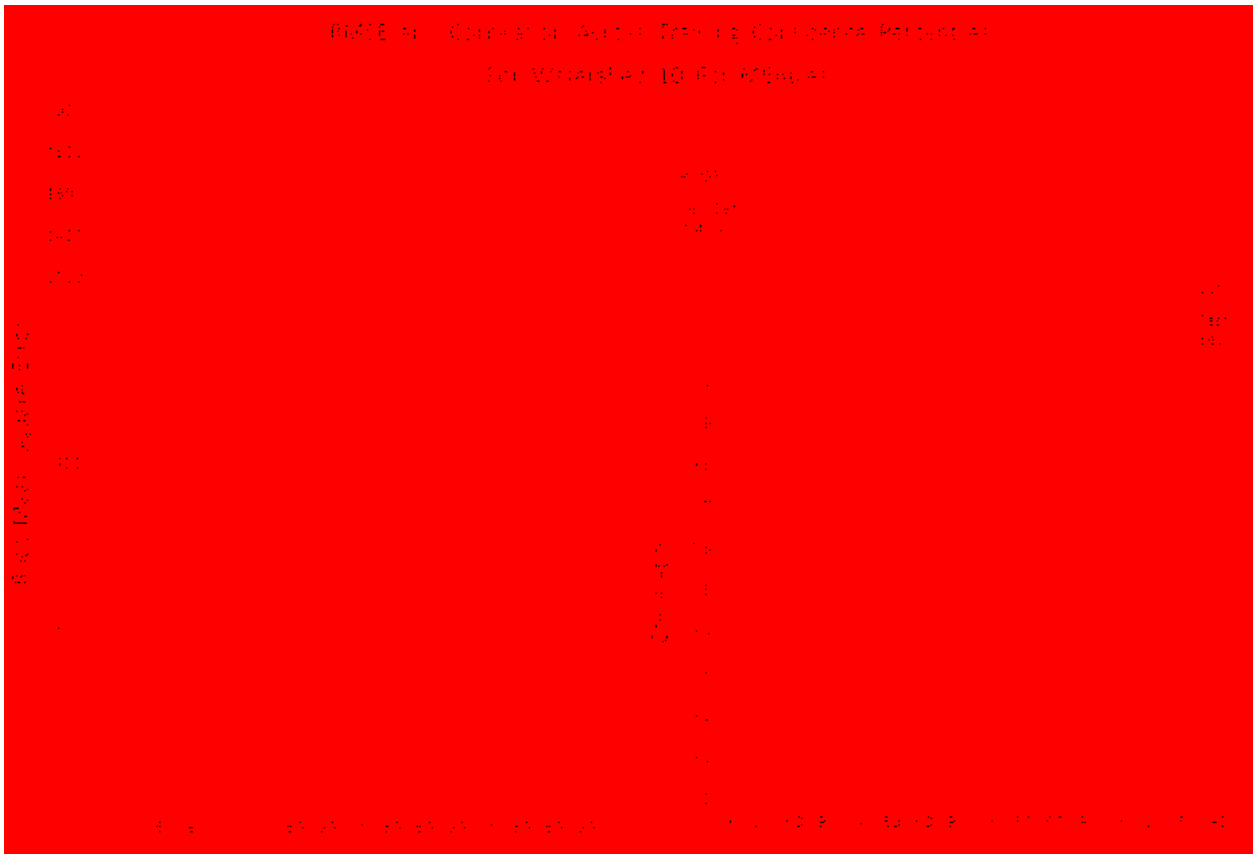

Figure 25: RMSE and Correlation results from Watershed 10, using M5Rules, 4 testing percentiles (0, 50, 90, 95) and 10 training percentiles (0, 10, 20, …90). The red 'Test Set' line is proportional to test set size, and is provided to help distinguish the 4 different testing percentile test sets (i.e. 0, 50, 90, 95).

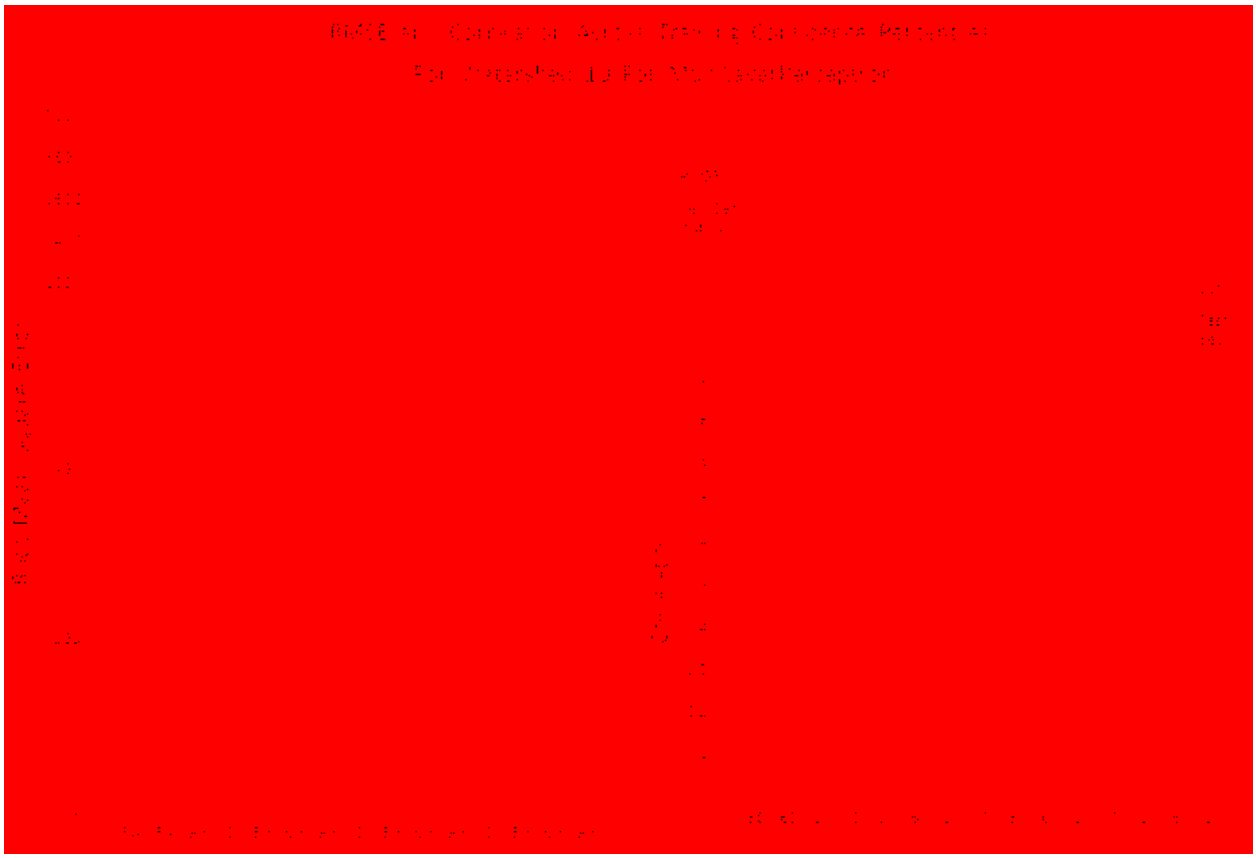

Figure 26: RMSE and Correlation results from Watershed 10, using MultiLayerPerceptron.

Formatting is otherwise the same as in Figure 25.

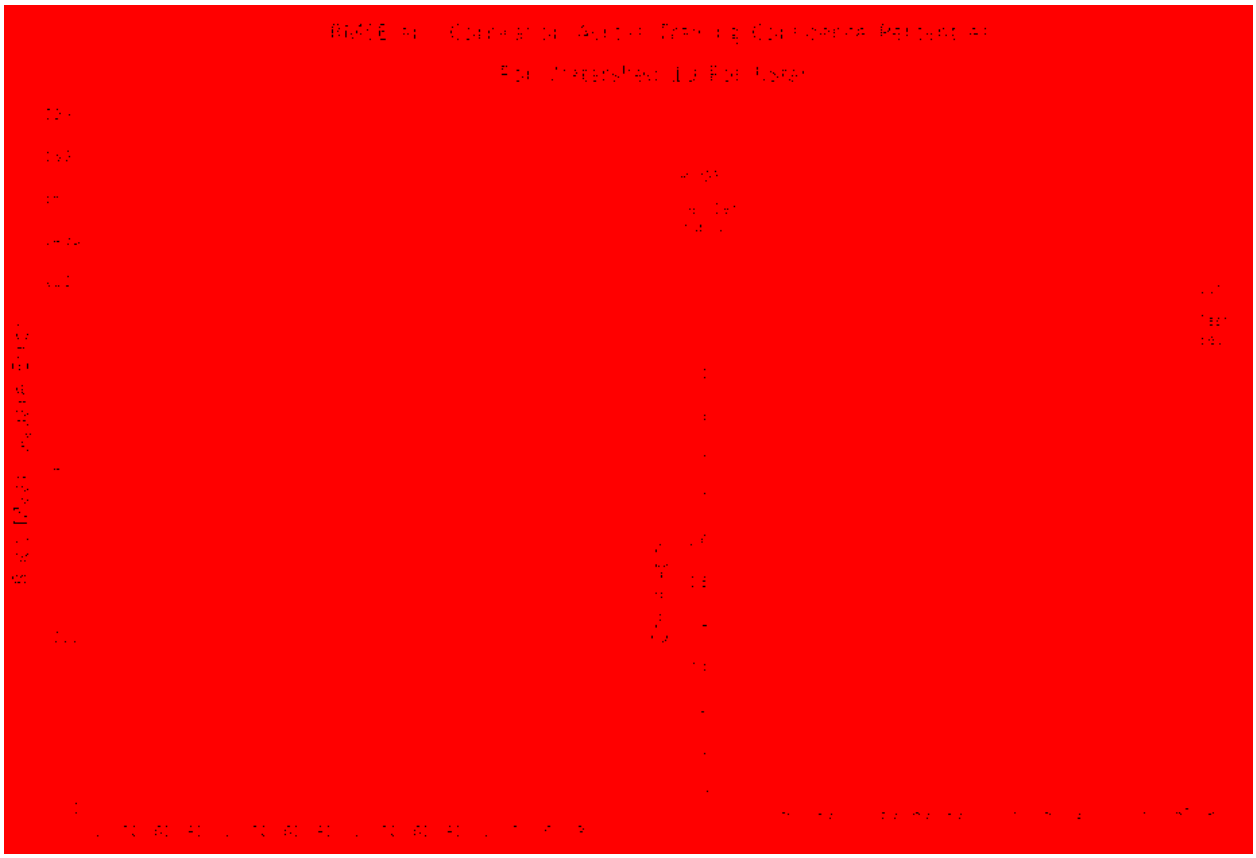

Figure 27: RMSE and Correlation results from Watershed 10, using KStar. Formatting is otherwise the same as in Figure 25.

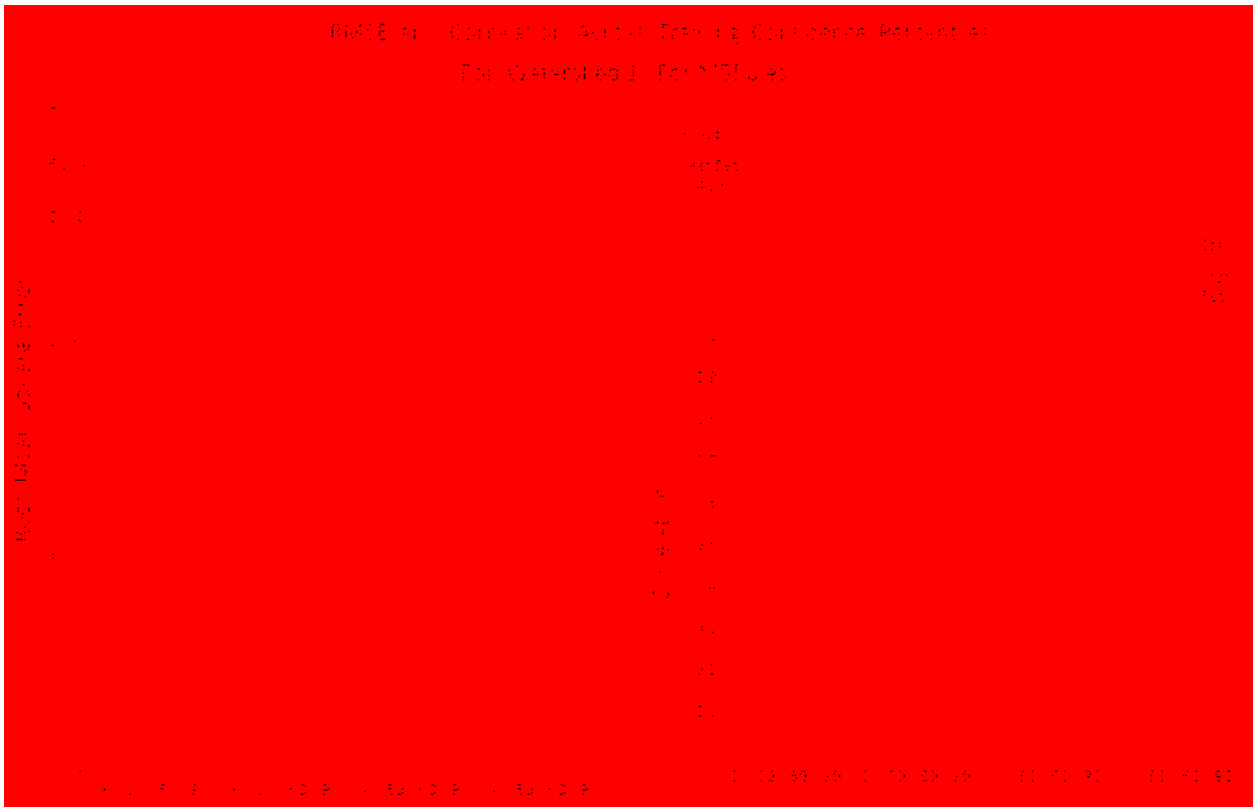

Figure 28: RMSE and Correlation results from Watershed 1, using M5Rules. Formatting is otherwise the same as in Figure 25.

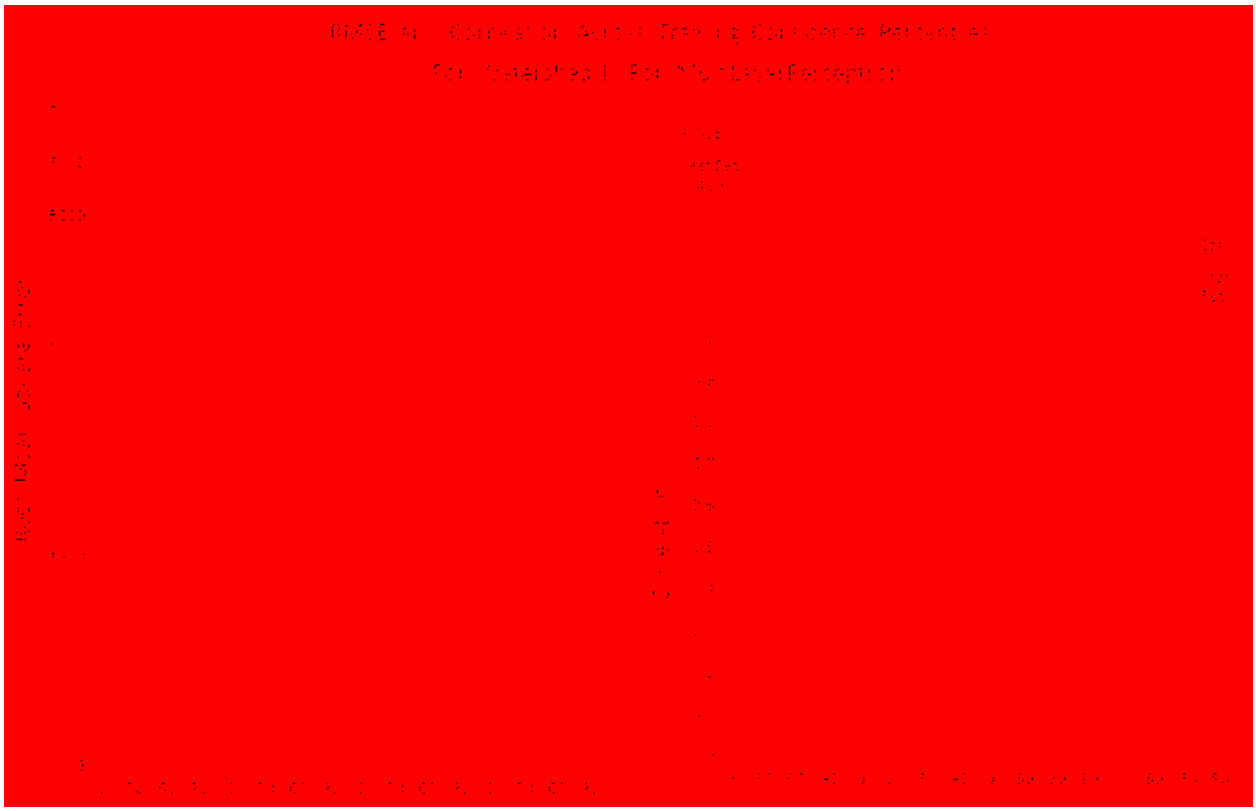

Figure 29: RMSE and Correlation results from Watershed 1, using MultiLayerPerceptron.

Formatting is otherwise the same as in Figure 25.

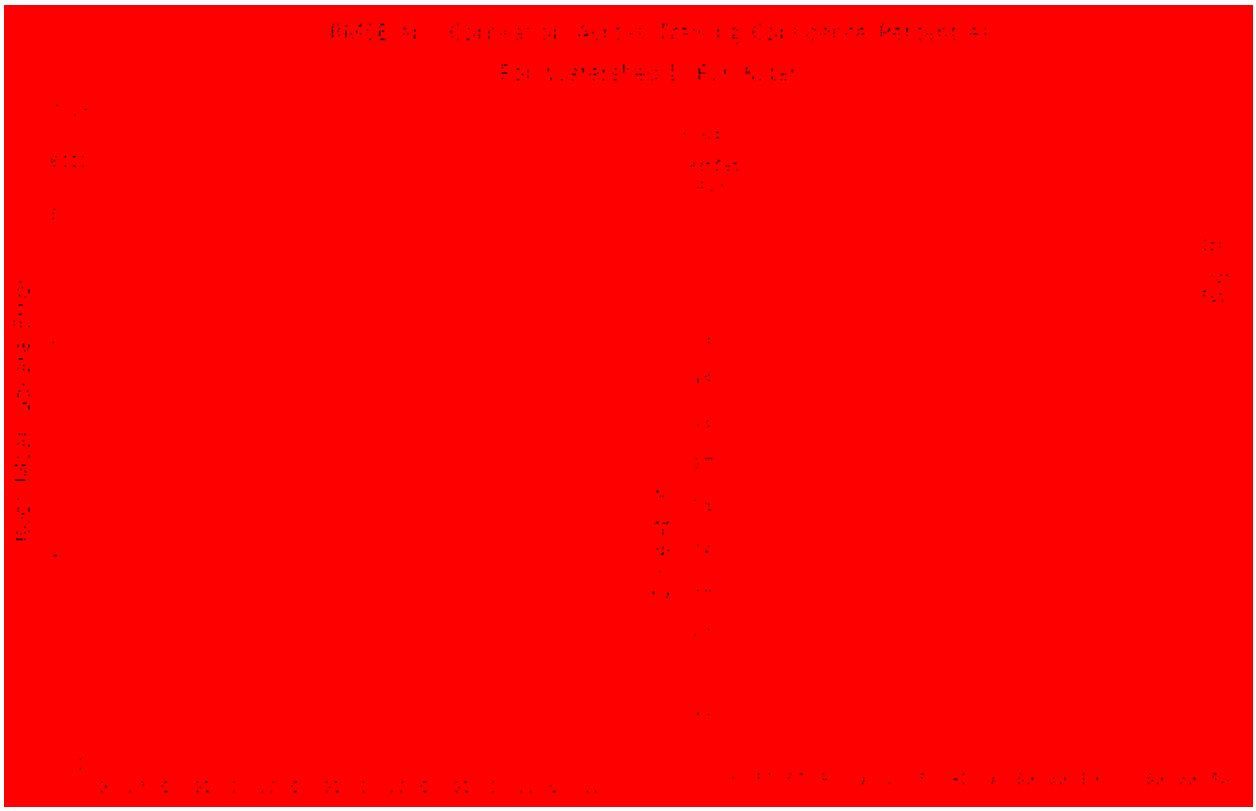

Figure 30: RMSE and Correlation results from Watershed 1, using KStar. Formatting is otherwise the same as in Figure 25.

Watershed 10

Watershed 10 seems to generally exhibit increased performance in the middle of its training-percentile curve. This is evidenced a little more sharply at higher testing-percentiles, particularly the 95th, where correlation increases significantly – Another indicator that the model is performing as expected on this data. With weaker correlation at the 0th, 50th, and 90th testing percentiles, one might conjecture that only the top 5 most confident percentile or so are consistent enough for accurate testing in Watershed 10.

Distinct Peaks

Also important to note are the *distinct peaks* manifested along the performance curve for each learning algorithm. We will define '*distinct peak*' here as a performance curve that does not achieve any correlation within.02 (for single underline, or .05 for double underline as shown in Figure 31) of the optimum correlation, outside of a two-step (20 percentile) window containing the optimum. This signifies when performance is ***sensitive to confidence***, in that there is a narrow (20 percentile) range of confidence which yields consistently better results than using some confidence threshold (and thus data selection) outside of that. This provides additional evidence that Confidence-Prioritization assists in data selection: beyond comparing performance against a constant control, it is shown that performance can achieve significant gains by being *selectively* determined, something that may not ever be noticed or addressed in ad-hoc data selection but which will come naturally with a Confidence-Prioritization approach.

These are also displayed in the summary table in Figure 31, again with underlined entries signifying a correlation margin of .02, and double underline signifying a margin of .05. Both assume a window of 20 percentile.

Watershed 10 Summarized Results

At 0%ile Confidence

Corr	tst-0	tst-50	tst-90	tst-95
M5Rules	0.40	0.34	0.43	0.56
MLP	0.51	0.40	0.33	0.60
KStar	0.47	0.33	0.27	0.51

Improvement over 0%ile

Corr	tst-0	tst-50	tst-90	tst-95
M5Rules	0.03	**0.19**	**0.12**	**0.23**
MLP	0.06	0.10	0.19	0.14
KStar	**0.11**	0.18	0.24	0.27

Optimum Correlation

Corr	tst-0	tst-50	tst-90	tst-95
M5Rules	0.43	**0.53**	**0.55**	**0.79**
MLP	0.57	0.49	0.52	0.74
KStar	**0.58**	0.51	0.51	0.78

and associated Training Confidence %ile

opt. trn	tst-0	tst-50	tst-90	tst-95
M5Rules	20	**80**	**50**	**80**
MLP	40	60	50	40
KStar	**30**	60	60	70

Figure 31: Summarized results for Watershed 10. *Top-left*: Baseline/Control (0th percentile confidence) results. *Top-right*: Improvement of Confidence-Percentile optimal over Control. *Bottom-left*: Optimal correlations found with Confidence-Prioritization. *Bottom-right*: Training confidence percentiles corresponding to optimum correlations. **Bold**ed correlations are those which are the highest among learning algorithms for that test %ile. Those which are underlined exhibit *distinct peaks* along their training percentile curves, defined as having no correlations within .02 (for single underline, or .05 for double underline) of the optimum outside a two-step (20 percentile) window that contains the optimum.

Watershed 1

Watershed 1 also exhibits distinct peaks, and at lower testing percentiles than in Watershed 10. The 0th *test*ing percentile keeps correlation down around .5 (as all testing percentiles do on other watersheds), but the 50th, 90th, and 95th all allow correlation to reach .9 – though only for a distinct, relatively narrow band of training percentiles, beyond which correlation drops on either side.

Some of these peaks are wider than 20 percentile, and thus less sensitive to particular data selection, but still quite clearly drop off steeply on either side. Figure 30 shows several examples of this. Figure 32 summarizes the optimum training percentiles and correlations for each training algorithm and test percentile.

Watershed 1 Summarized Results

At 0%ile Confidence

Corr	tst-0	tst-50	tst-90	tst-95
M5Rules	0.53	0.76	0.74	0.80
MLP	0.52	0.72	0.68	0.73
KStar	0.53	0.46	0.57	0.60

Improvement over 0%ile

Corr	tst-0	tst-50	tst-90	tst-95
M5Rules	0.00	0.10	0.07	0.00
MLP	0.06	**0.17**	0.22	0.13
KStar	**0.06**	0.40	**0.36**	**0.28**

Optimum Correlation

Corr	tst-0	tst-50	tst-90	tst-95
M5Rules	0.53	0.86	0.81	0.80
MLP	0.58	**0.89**	0.90	0.86
KStar	**0.59**	0.86	**0.93**	**0.88**

And associated Training Confidence %ile

opt. trn	tst-0	tst-50	tst-90	tst-95
M5Rules	0	50	60	0
MLP	30	**50**	60	60
KStar	**30**	50	**60**	**60**

Figure 32: Summarized results for Watershed 1. Formatting is the same as is given in Figure 31.

Discussion of Environmental Science Study

Comparing Learning Algorithms

A few interesting things to note about these results: M5Rules seemed more robust in that the correlation produced by resulting models did not vary as much across training percentiles as did MLP and KStar. However, MLP and KStar achieved better results, but only within a certain range of training percentiles. Additionally, on Watershed 10, on the 95th percentile testing set, there were distinct peaks and some

higher correlations (.74-.79) at optimum training percentiles, but that optimum training percentile changed between training algorithms. For KStar it was the 80th (at a correlation of .79), for M5Rules it was the 40th (.74), and for MLP it was the 70th (.78).

What we might infer from this is primarily that the effectiveness of Confidence-Prioritization may be affected by choice of training algorithm. The margin of error for training confidence percentile with some training algorithms may be wider (eg. M5Rules) or narrower (eg. MLP or KStar), and the relative gains may be greater (such as for KStar on Watershed 1) or smaller (such as for M5Rules on Watershed 1). We may draw some tentative conclusions that instance-based classifiers such as KStar will gain more significantly from Confidence-Prioritization – which is not surprising – however a more thorough analysis of differing algorithms' response to Confidence-Prioritization is not a study we attempt here.

Discussion of Watershed 10

In terms of hold-out test sets, Watershed 10 also manifested an interesting pattern in that correlation was relatively low (below .6) across all test sets except that of the 95th percentile confidence test set. Correlation for the 0, 50, and 90th percentile confidence test sets stayed around 50%, which is comparable with correlation across ALL test sets for all other watersheds – where prediction was worse, as we discussed earlier in the chapter (Refer to the Appendix for results on watersheds besides Watershed 1 and Watershed 10). This suggests that the data in Watershed 10 is

generally less confident and more uncertain/noisy than in Watershed 1 (i.e. the 90th percentile most confident data points in Watershed 10 would exhibit more noise and uncertainty than, say, the 90th percentile in Watershed 1, even though both are 'top 10%').

Nonetheless, Watershed 10 did exhibit performance improvements using Confidence-Prioritization, as evidenced by (a) improved correlation over the Control data set, and (b) manifesting *distinct peaks* in performance which provide evidence for Confidence-Prioritization making effective, narrow selections of data that would be much less likely to achieve under a standard ad-hoc data selection.

Discussion of Watershed 1

Watershed 1 also exhibited great performance improvements over the control, as well as consistent *distinct peaks* on performance curves across confidence percentile. Perhaps most notably, KStar achieved the greatest results across most test sets, but only under a Confidence-Prioritization approach. Using the default data set resulted in significantly worse results using KStar. Also, as with the distinct peaks shown in Watershed 10, the peaks in Watershed 1 indicate that performance results are ***sensitive to data set selection***, and performance would drop if much variation in data set selection were made.

This, again, supports our argument that Confidence-Prioritization helps to select optimal data sets for training, over ad hoc approaches, because it allows data selection

to hone in on an optimal data set which might otherwise escape selection by a human data collector. Furthermore, it does so with an iterative selection process based off of confidence as determined ***before*** training – as opposed to an ad-hoc selection process which would merely drop points which are 'difficult to classify.' **This helps to ensure that salient, accurate outliers are not excluded**.

These results also address our earlier question of, "*In a data set with human-estimated confidence values (i.e.* ***where the association of high confidence with more accurate data is uncertain****, as opposed to the simulation study), does using Confidence-Prioritization still help identify optimal data sets for training?*" concluding that Confidence-Prioritization does indeed still help identify optimal data sets for training, among real-world data with uncertain confidence estimations.

Data Set Specificity

In regards to different watersheds exhibiting different strengths of diel signals, one thing this study did show is that, at least partly due to variability in confidence, some watersheds may lend themselves to numerical analysis and data mining more readily than others. For example, in this study, Watershed 1 and to a degree Watershed 10 both had diel signals which we were able to predict with a fairly high correlation (being above .9 for Watershed 1, and above .75 for Watershed 10); with all other watersheds we did not achieve any correlations much higher than .5 or .6. This is not to say that data mining could not successfully be done to predict diel signals in these other

watersheds, just that it may be a more difficult problem. Confidence-Prioritization, like any Data Mining tool, may be considered as ***one link in a chain of multiple factors***, a tool which can help greatly when the other factors required are present – factors such as sufficiently abundant, ambiguously noisy data that holds estimable confidence.

As far as spatial-temporal qualities of watersheds (Figure 36 in the Appendix shows such qualities for watersheds in the HJA forest) go, it seems that if comparing across watersheds, it may be prudent to manually analyze comparisons unless you have data for many watersheds. For example, in this study we observed that the steeper slope, high channel-length-to-area ratio, and young tree stands between Watershed 1 and Watershed 10 may contribute to stronger and clearer diel signals. Most studies only incorporate a single watershed, or handful (2-3), while even our study incorporated less than a dozen. If one were to conduct a data mining study across, say, 100 watersheds, perhaps some interesting patterns could be more reliably extracted with data mining.

Comparison to Control

Also, there are some noteworthy comparisons between the baseline, control training set (i.e. under "at 0%ile confidence" in Figure 32) and the other optimum results. With the 0th percentile confidence **test** set, MLP and KStar show a marked improvement in correlation by using Confidence-Prioritization, KStar being more especially noteworthy because of the steep peaks it exhibits across training percentiles

(see Figure 30), evidencing the ability of Confidence-Prioritization to find that optimal middle ground, acquiring *enough* high- quality data to train a valid model without acquiring *too much* low-quality data that would skew results. The improvements of Confidence-Prioritization over the baseline are greater in the higher test percentiles (50, 90, and 95), where distinct peaks are also seen, across all training algorithms. Notably the improvements are *greatest* with KStar, which also has the highest results overall across training algorithms. That Confidence-Prioritization could be particularly effective for an instance-based classifier such as KStar, where results can be particularly good but also sensitive to *data selection*, is significant.

Conclusions

Some important implications we draw from these results are:

As Related to Choice of Algorithm:

A. **Confidence-percentile-based testing provides a flexibility in data to train optimally with each of multiple training algorithms**, something that the traditional fixed-testing-set approach does not provide,
B. The margin of error for training confidence percentile with some training algorithms may be wider (eg. M5Rules) or narrower (eg. MLP or KStar),
C. Some algorithms will gain significantly in performance from Confidence-Prioritization, one potential class being instance-based classifiers such as KStar.

As Related to Performance:

D. Confidence-Prioritization is shown to provide significant performance improvement over reasonable ad-hoc data set selections,

E. As evidenced by the *distinct peaks* shown in performance curves for correlation, Confidence-Prioritization identifies a narrow band of confidence threshold for training that results in markedly better models than data selection outside that band would result in – **Providing evidence that Confidence-Prioritization does indeed help select optimal data sets for training**, and does so selectively, even among data sets where confidence is estimated and therefore not perfectly correlated to accuracy.

CHAPTER VII: DISCUSSION AND CONCLUSION

We've provided background, built a methodology, exercised it against a Simulation Study to provide an example of implementation and proof of concept in a controlled environment, and exercised it again against a real world, noisy data set to provide a further example and evidence for the effectiveness of Confidence-Prioritization. Through all of this we evaluate the central statement, "*Using a Confidence-Prioritization approach to data collection and data mining can assist in selecting optimal data sets for training over standard, ad-hoc data collection, particularly for data sets with both accurate and inaccurate data and an unclear distinction between them.*"

Outcomes

As shown in our Simulation Study in Chapter V, under conditions of confidence values that are *known to correspond to each data point's actual accuracy*, among data with variable confidence, Confidence-Prioritization does help to select optimal data sets for testing. Our Environmental Science Study further provides evidence that even in situations where *confidence is estimated and therefore not fully known to correspond to actual accuracy*, Confidence-Prioritization is shown to help find optimal data sets for training.

This is evidenced both by a simple comparison of significantly improved performance against a control data set selection, and perhaps more importantly by *the*

shape of the performance curves across training confidence percentiles. As can be seen in, for example, Figure 29 and Figure 30, distinct peaks occur within narrow ranges of confidence, and drop off to either side, providing evidence that ***performance in these situations is sensitive to data set selection***. Using too much inaccurate data, or too little [accurate] data, can damage performance, and ***Confidence-Prioritization can help identify this narrow band of selection***, essentially helping the human data collector to avoid selecting a data set – one that effectively falls off of either side of this curve – using some single ad-hoc approach.

Further Emergent Patterns

Besides evidence to support the central thesis, some interesting patterns have emerged regarding how Confidence-Prioritization performs.

Optimal Confidence Percentiles

One pattern is that when testing against low-quality data such as the 0th and 50th percentile test sets, optimum training percentiles tend to remain around the test percentile (i.e. 0 or 50, respectively). However, when testing against high-quality, 90th+ percentile-confidence data sets, optimum training percentiles are lower, in the 65th to 85th percentile range. This is even more accentuated on the environmental data set, where the 60th percentile seems to be best for training against higher-confidence data sets. This may indicate an issue of data sparseness, further supporting our argument that training on very sparse high-quality data is often inferior to accepting some

variance of quality along with a larger data set, even when training against high quality data – Of course, this is to be balanced with not training on too much low-quality, misrepresentative data. That balance is central to this study.

Variance among Algorithms and Data Sets

Also, using Confidence-Prioritization seems to make a larger difference for some data sets and learning algorithms than others. For instance, not surprisingly, the impact of training set size is such that in small training data sets, there are more significant performance losses for training on sub-optimal confidence thresholds than with larger test sets. This is observable on the steeper curve for smaller (n=200 and n=40) training sets than with larger (n=500) training sets in Figure 8, Figure 9, and Figure 11, particularly for Correlation without confidence as an input, and RMSE with confidence as an input.

Consistent with what is known about instance-based classifiers, the results for the environmental data study (Figure 28) show steep performance curves for KStar, consistent with KStar (and other instance-based classifiers) being particularly susceptible to misrepresentative (i.e. low-confidence) data. However, as shown in this case, instance-based classifiers may have the potential to provide high-performance models *when provided with accurately representative, confidence-prioritized data* – higher performance than both other training algorithms (M5Rules and MLP) we used in this study.

Human Investment

Confidence-Prioritization does require some investment to apply data-record-level confidence values, however, using Confidence-Prioritization over traditional training-set selection certainly does not hurt, and under our observations, **it is very likely to help select optimal data sets for training, under any situation in which there is some variability as to the data collector's trust or confidence in collected data points**.

Future Work

Some future work may be done in training with confidence as an attribute; *weighting* data points for training by confidence is one area in particular in which we see potential gains. Augmenting the confidence-assignment process with **Active Learning** is also a likely place for significant efficiency gains to be had in this process. Also, evaluating the relative performance gains and qualities of Confidence-Prioritization on Boosting and other training algorithms could provide further insight. Likewise, using Confidence-Prioritization on input variables as well as output variables may provide additional insights as well.

Fully automating confidence determination, relinquishing the need for a human data collector's input, may also be an interesting area of research. However, due to the potential complexity of this approach, according to our judgment, approaching the other areas of future work first may provide greater returns.

Finally, we suggest that in these and other future work, significant gains can be made, simply with the general approach of **utilizing a data collector's variable degree of confidence in how accurately a data record represents the underlying system**. Diel Signal Estimation shows one example in a broad field of Environmental Science for which Confidence-Prioritization has shown a potential to augment the data mining process by accounting for varying degrees of confidence in inherently noisy systems.

BIBLIOGRAPHY

[1] A. Ben-David and E. Frank, "Accuracy of machine learning models versus “hand crafted” expert systems–A credit scoring case study," *Expert Systems with Applications,* vol. 36, pp. 5264-5271, 2009.

[2] J. Attenberg and F. Provost, "Why label when you can search?: alternatives to active learning for applying human resources to build classification models under extreme class imbalance," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 423-432.

[3] J. A. Fails and D. R. Olsen Jr, "Interactive machine learning," in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, Miami, Florida, 2003, pp. 39-45.

[4] H. J. Escalante, "A comparison of outlier detection algorithms for machine learning," in *Proceedings of the International Conference on Communications in Computing*, 2005, pp. 228-237.

[5] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review,* vol. 22, pp. 85-126, 2004.

[6] H. Barnard, C. Graham, W. Van Verseveld, J. Brooks, B. Bond, and J. McDonnell, "Mechanistic assessment of hillslope transpiration controls of diel

subsurface flow: a steady-state irrigation approach," *Ecohydrology,* vol. 3, pp. 133-142, 2010.

[7] B. J. Bond, J. A. Jones, G. Moore, N. Phillips, D. Post, and J. J. McDonnell, "The zone of vegetation influence on baseflow revealed by diel patterns of streamflow and vegetation water use in a headwater basin," *Hydrological Processes,* vol. 16, pp. 1671-1677, 2002.

[8] Z. Gribovszki, J. Szilágyi, and P. Kalicz, "Diurnal fluctuations in shallow groundwater levels and streamflow rates and their interpretation-A review," *Journal of Hydrology,* vol. 385, pp. 371-383, 2010.

[9] S. M. Wondzell, M. N. Gooseff, and B. L. McGlynn, "An analysis of alternative conceptual models relating hyporheic exchange flow to diel fluctuations in discharge during baseflow recession," *Hydrological Processes,* vol. 24, pp. 686-694, 2010.

[10] G. W. Moore, J. A. Jones, and B. J. Bond, "How soil moisture mediates the influence of transpiration on streamflow at hourly to interannual scales in a forested catchment," *Hydrological Processes,* vol. 25, pp. 3701-3710, 2011.

[11] S. M. Wondzell, M. N. Gooseff, and B. L. McGlynn, "Flow velocity and the hydrologic behavior of streams during baseflow," *Geophysical Research Letters,* vol. 34, p. L24404, 2007.

[12] B. Settles, "Active learning literature survey," University of Wisconsin, Madison2010.

[13] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th International Conference on Machine Learning*, ed. Montreal, Quebec, Canada: ACM, 2009, pp. 41-48.

[14] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning,* vol. 36, pp. 105-139, 1999.

[15] A. J. G. Hey, S. Tansley, and K. M. Tolle, *The fourth paradigm: data-intensive scientific discovery*. Redmond, WA: Microsoft Research, 2009.

[16] J. S. Clark, *Models for ecological data: an introduction*: Princeton University Press Princeton, New Jersey, USA, 2007.

[17] N. McMillan, S. M. Bortnick, M. E. Irwin, and L. Mark Berliner, "A hierarchical Bayesian model to estimate and forecast ozone through space and time," *Atmospheric Environment,* vol. 39, pp. 1373-1382, 2005.

[18] C. K. Wikle, "Hierarchical models in environmental science," *International Statistical Review,* vol. 71, pp. 181-199, 2003.

[19] D. D. Cadol, S. K. Kampf, and E. E. Wohl, "Diel discharge cycles as indicators of evapotranspiration rates, with implications for groundwater dynamics," in *Proceedings of the American Geophysical Union Fall Meeting*, 2010, p. 05.

[20] J. C. Koch, D. M. McKnight, and R. M. Neupauer, "Simulating unsteady flow, anabranching, and hyporheic dynamics in a glacial meltwater stream using a

coupled surface water routing and groundwater flow model," *Water Resources Research,* vol. 47, p. W05530, 2011.

[21] N. Móricz, "Water balance study of a groundwater-dependent oak forest," *Acta Silvatica et Lignaria Hungarica,* vol. 6, pp. 49-66, 2010.

[22] J. Zhu, M. Young, J. Healey, R. Jasoni, and J. Osterberg, "Interference of river level changes on riparian zone evapotranspiration estimates from diurnal groundwater level fluctuations," *Journal of Hydrology,* vol. 403, pp. 381-389, 2011.

[23] A. R. Burke and T. Kasahara, "Subsurface lateral flow generation in aspen and conifer-dominated hillslopes of a first order catchment in northern Utah," *Hydrological Processes,* vol. 25, pp. 1407-1417, 2011.

[24] C. S. Lowry, J. S. Deems, S. P. Loheide II, and J. D. Lundquist, "Linking snowmelt-derived fluxes and groundwater flow in a high elevation meadow system, Sierra Nevada Mountains, California," *Hydrological Processes,* vol. 24, pp. 2821-2833, 2010.

[25] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: a new source of inexpensive, yet high-quality, data?," *Perspectives on Psychological Science,* vol. 6, pp. 3-5, 2011.

[26] K. Clement, N. Gustafson, A. Berbert, H. Carroll, C. Merris, A. Olsen, M. Clement, Q. Snell, J. Allen, and R. J. Roper, "PathGen: a transitive gene pathway generator," *Bioinformatics,* vol. 26, pp. 423-425, 2010.

[27] D. P. Solomatine and D. L. Shrestha, "AdaBoost. RT: A boosting algorithm for regression problems," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2004, pp. 1163-1168.

[28] J. E. Hobbie, S. R. Carpenter, N. B. Grimm, J. R. Gosz, and T. R. Seastedt, "The US long term ecological research program," *BioScience,* vol. 53, pp. 21-32, 2003.

[29] E. J. Albright, N. Gustafson, M. B. Nelson, J. M. Ramirez, B. M. Rodriguez-Cardona, C. M. Shughrue, and J. A. Jones, "Diel fluctuations in summer streamflow depend on stream channel sediment storage and valley-floor vegetation in the forested western Cascades of Oregon, USA," in *Proceedings of the American Geophysical Union Fall Meeting*, 2010, p. 1041.

[30] J. Szilágyi, Z. Gribovszki, P. Kalicz, and M. Kucsara, "On diurnal riparian zone groundwater-level and streamflow fluctuations," *Journal of Hydrology,* vol. 349, pp. 1-5, 2008.

[31] J. G. Cleary and L. E. Trigg, "K^*: An instance-based learner using an entropic distance measure," in *Proceedings of the 12th International Conference on Machine Learning*, San Francisco, 1995, pp. 108-114.

APPENDIX

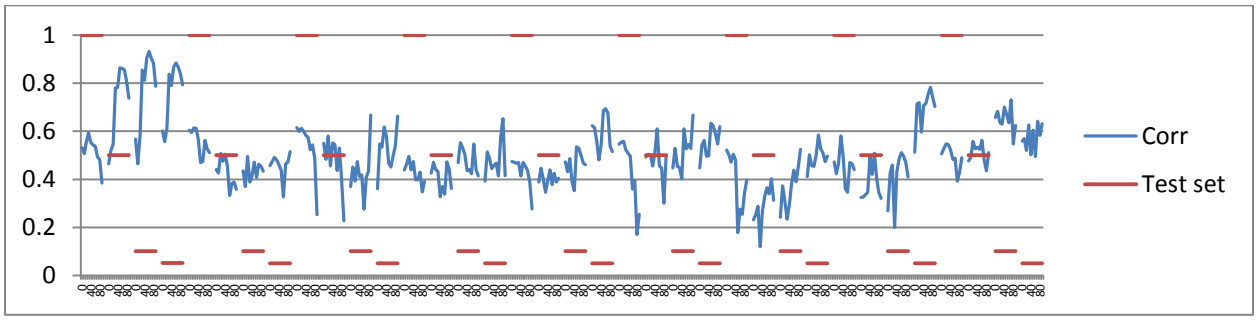

Figure 33: KStar results on our Environmental data sets. Using (from left to right, and highest level to lowest level): Watersheds: WS1, WS2, WS3, WS6, WS7, WS8, WS9, WS10, WSMC; Testing percentiles: 0, 50, 90, 95; Training percentiles: 0, 10, 20, 30, 40, 50, 60, 70, 80, 90.

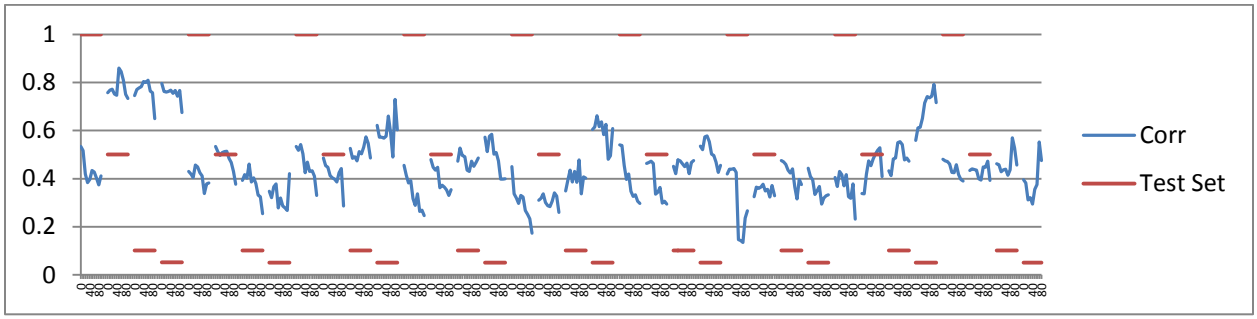

Figure 34: M5Rules results for Environmental Data sets. Layout and data is same as for that in Figure 33.

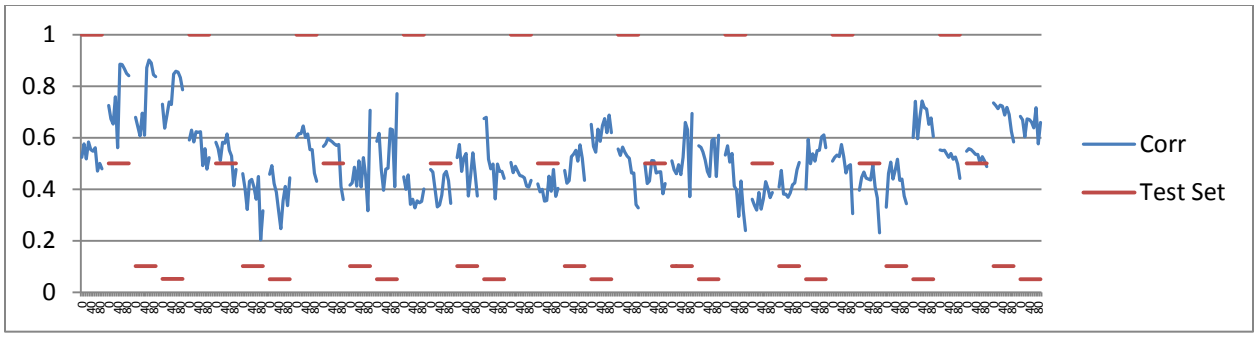

Figure 35: MultiLayerPerceptron results for Environmental Data sets. Layout and data is same as for that in Figure 33.

TAG	AREA	ASPECT	MIN_ ELEV	MAX_ ELEV	MEAN_ANN _PRECIP	MEAN_ ANN_RAD	SLOPE	CHANNEL _LEN	DRAIN_ DENS_min
LOOK-OUT	6242	267	428	1627	2400	12	40.5	141032	0.47
MACK	581	306	758	1625	2500	12.25	48	11120	0.53
WS01	95.9	286	457	1027	2300	12.25	59	2808	0.46
WS02	60.3	318	548	1078	2300	12.25	53	1861	0.32
WS03	101.1	313	418	1080	2300	12.25	52	2771	0.39
WS06	13	165	897	1029	2200	12	25	112	0.86
WS07	15.4	158	938	1102	2200	12	34	125	0.81
WS08	21.4	165	993	1182	2200	12	26	318	0.82
WS09	8.5	247	432	731	2260	12.25	58	NA	NA
WS10	10.2	250	473	679	2260	12.25	58	456	NA

Figure 36: Watershed properties in the HJ Andrews Experimental Forest.