2018

Fluvial Processes in Motion: Measuring Bank Erosion and Suspended Sediment Flux using Advanced Geomatic Methods and Machine Learning

Scott Douglas Hamshaw
University of Vermont

FLUVIAL PROCESSES IN MOTION: MEASURING BANK EROSION AND SUSPENDED SEDIMENT FLUX USING ADVANCED GEOMATIC METHODS AND MACHINE LEARNING

A Dissertation Presented

by

Scott D. Hamshaw

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfilment of the Requirements
For the Degree of Doctor of Philosophy
Specializing in Civil and Environmental Engineering

January, 2018

Defense Date: September 12, 2017
Dissertation Examination Committee:

Donna M. Rizzo, Ph.D., Advisor
Andrew W. Schroth, Ph.D., Chairperson
Mandar M. Dewoolkar, Ph.D.
Beverley C. Wemple, Ph.D.
Arne Bomblies, Ph.D.
Jarlath O'Neil-Dunne, M.S.
Cynthia J. Forehand, Ph.D., Dean of the Graduate College

ABSTRACT

Excessive erosion and fine sediment delivery to river corridors and receiving waters degrade aquatic habitat, add to nutrient loading, and impact infrastructure. Understanding the sources and movement of sediment within watersheds is critical for assessing ecosystem health and developing management plans to protect natural and human systems. As our changing climate continues to cause shifts in hydrological regimes (e.g., increased precipitation and streamflow in the northeast U.S.), the development of tools to better understand sediment dynamics takes on even greater importance. In this research, advanced geomatics and machine learning are applied to improve the (1) monitoring of streambank erosion, (2) understanding of event sediment dynamics, and (3) prediction of sediment loading using meteorological data as inputs.

Streambank movement is an integral part of geomorphic changes along river corridors and also a significant source of fine sediment to receiving waters. Advances in unmanned aircraft systems (UAS) and photogrammetry provide opportunities for rapid and economical quantification of streambank erosion and deposition at variable scales. We assess the performance of UAS-based photogrammetry to capture streambank topography and quantify bank movement. UAS data were compared to terrestrial laser scanner (TLS) and GPS surveying from Vermont streambank sites that featured a variety of bank conditions and vegetation. Cross-sectional analysis of UAS and TLS data revealed that the UAS reliably captured the bank surface and was able to quantify the net change in bank area where movement occurred. Although it was necessary to consider overhanging bank profiles and vegetation, UAS-based photogrammetry showed significant promise for capturing bank topography and movement at fine resolutions in a flexible and efficient manner.

This study also used a new machine-learning tool to improve the analysis of sediment dynamics using three years of high-resolution suspended sediment data collected in the Mad River watershed. A restricted Boltzmann machine (RBM), a type of artificial neural network (ANN), was used to classify individual storm events based on the visual hysteresis patterns present in the suspended sediment-discharge data. The work expanded the classification scheme typically used for hysteresis analysis. The results provided insights into the connectivity and sources of sediment within the Mad River watershed and its tributaries. A recurrent counterpropagation network (rCPN) was also developed to predict suspended sediment discharge at ungauged locations using only local meteorological data as inputs. The rCPN captured the nonlinear relationships between meteorological data and suspended sediment discharge, and outperformed the traditional sediment rating curve approach. The combination of machine-learning tools for analyzing storm-event dynamics and estimating loading at ungauged locations in a river network provides a robust method for estimating sediment production from catchments that informs watershed management.

CITATIONS

Material from this dissertation has been published in the following form:

Hamshaw, S. D., Bryce, T., Rizzo, D. M., O'Neil-Dunne, J., Frolik, J., & Dewoolkar, M. M.. (2017). Quantifying streambank movement and topography using unmanned aircraft system photogrammetry with comparison to terrestrial laser scanning. River Research and Applications. doi: 10.1002/rra.3183

AND

Material from this dissertation has been submitted for publication to Geomatics, Natural Hazards, and Risk on October 31, 2017 in the following form

Hamshaw, S.D., Engel, T., Rizzo, D.M., O'Neil-Dunne, J., & Dewoolkar, M. M.. (2017). Application of unmanned aircraft system (UAS) for monitoring bank erosion along a river corridor. Geomatics, Natural Hazards and Risk.

AND

Material from this dissertation has been submitted for publication to Hydrological Processes on November 5, 2017 in the following form

Hamshaw, S.D. and Rizzo, D.M.. (2017). Predicting river suspended sediment loadings using hydrometeorological variables and counterpropagation neural networks. Hydrological Processes.

AND

Material from this dissertation has been submitted for publication to Water Resources Research on November 13, 2017 in the following form

Hamshaw, S.D., Dewoolkar, M.M., Schroth, A.W., Wemple, B.C., and Rizzo, D.M.. (2017). A new machine-learning approach for classifying hysteresis in suspended-sediment discharge relationships using high-frequency monitoring data. Water Resources Research.

ACKNOWLEDGEMENTS

This body of research would not have been possible nor successful without the assistance, guidance, and encouragement from many people. I would like to first acknowledge my committee for their guidance, encouragement, and interest during the development of this dissertation. Thank you to Andrew Schroth, Jarlath O'Neil-Dunne, Arne Bomblies, and Beverley Wemple –your research endeavors at UVM are the reason why UVM is an inspiring place to do research. Thanks to my unofficial co-advisor Mandar Dewoolkar, whose support and knowledge in my research was greatly appreciated and whose combination of positive attitude and efficient approach made finishing this dissertation possible. Finally, a special thank you to my advisor, Donna Rizzo, who was unwavering in offering guidance, encouragement, and who, no matter the seemingly impossible deadlines, was dedicated to giving valuable feedback. In the fifteen years I have known her at UVM, she has been continuously committed to supporting in achieving my personal and career goals, and it was no different with supporting my teaching and research during my PhD program.

I was fortunate to be able to work with many terrific undergraduate interns and fellow graduate students in conducting this research and it would not have been possible without them. Thank you to interns Alex Morton, Hanna Anderson, Kira Kelley, Anna Waldron, Jordan Duffy, Nathalie Simoes, and Wimara Rubia Sa Gomes for their help in collecting field data. Sincere thanks to fellow graduate students Thomas Bryce, Jody Stryker, Justin Guilbert, John Hanley, Nikos Fytilis, and Ian Anderson for their collaboration and contributions in data collection and feedback on project design. A special thank you to Kristen Underwood whose collaboration was especially valued during project design, data collection and analysis phases.

A number of programs and labs at UVM provided support for this project, in particular for field data collection. Thank you to EPSCoR RACC project staff Katie Chang, Janel Roberge, Saul Blocher, and Pat Clemins for their work on testing water quality samples and supporting data management. The UVM Spatial Analysis Lab (SAL), and in particular Tayler Engel and the UAS team, contributed significantly to data collection and processing. I would also acknowledge the contributions from Beverley Wemple, Jeff Frolik, Don Ross, and Breck Bowden's labs and their many graduate students and interns that I collaborated with on fieldwork and data collection.

I am especially grateful for the contributions of a number of people in the larger community that both helped guide this research and made it possible to undertake. A number of landowners in the Mad River Valley were remarkably accommodating of having myself and other students setting up monitoring stations on their property and helping to keep it safe. Thanks to the Friends of the Mad River for their support of all the UVM research undertaken in the Mad River Valley. This research would not have been possible without the funding and support provided by the National Science Foundation (NSF), Vermont EPSCoR, the Vermont Water Resources and Lake Studies center, and the Robert and Patricia Switzer Foundation.

Finally, thank you to my friends and family who both supported me during this work and tolerated many stretches of self-doubt and incessant discussions of writing progress. Mom, Dad, Mike and Sue, thank you for all the support and encouragement in undertaking this work. And of course, my wife, Kelly, who I cannot thank enough for her encouragement and support in all of my academic endeavors. I am sure I would not have had the resolve or confidence to complete this work without her. Thank you for being my partner throughout this process, helping me realize my aspirations, and making me a better person along the way.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION AND COMPREHENSIVE LITERATURE REVIEW

Motivation and Objectives

The dynamics of flowing water moving over the landscape is an immensely complex process that plays a crucial role in shaping our environment, both natural and built. As a result, the transport of sediment within watersheds has been studied extensively to understand both natural processes and anthropogenic impacts. In Vermont, there is increased urgency to understand the sources and quantity of sediment being discharged from watersheds due to the negative effects caused by excessive sediment loading to receiving waters such as Lake Champlain. These include degradation of aquatic habitat, reduced biodiversity, additional nutrient loading, and degraded water quality for drinking water sources. The role of excessive loading of fine sediments to rivers is of primary concern in the transport of phosphorous, a nutrient believed to exacerbate harmful algal blooms in Lake Champlain.

Recently, new challenges and opportunities in catchment sediment studies have presented themselves and motivated this research: (1) the impact of hydrological shifts on sediment production caused by a changing climate, (2) the availability of instream sensors capable of providing an increased volume of high frequency suspended sediment data, and (3) the advancement of machine learning and computer vision algorithms for both data analysis and topographic data generation. To this end, the following motivational questions have guided this research:

1. Streambank erosion is a large source of sediment and sediment-bound nutrient pollution to downstream waters. *Can digital photogrammetry using unmanned aircraft systems (UAS) methods improve the measurement of bank surfaces and erosion along river corridors?*

2. The characterization of the suspended sediment – discharge relationship over a single storm event can be used to infer mechanisms of sediment processes in a catchment. *Can high-frequency sediment data and machine learning improve our ability to extract information about sediment dynamics.*

3. Prediction of sediment loading and/or concentration in a catchment is important for management decisions and scenario analysis. *Can an artificial neural network be used to capture the nonlinear relationship between meteorological data and suspended sediment discharge in rivers, and then predict sediment discharge at ungauged locations using only meteorological data?*

The availability of suspended sediment data and measurements of streambank erosion in the Lake Champlain Basin, and Vermont in general, is limited given that traditional data collection methods are both time and resource intensive. At the same time, the advent of new sensors, designed to collect high-resolution data and reduce the demand on human resources needed for field collection efforts, has resulted new challenges – namely, the computational tools needed for storing, processing, and helping domain experts analyze and visualize these data have lagged behind (Pellerin et al., 2016). As a result, a significant component of this research involved monitoring in-stream suspended sediment data and streambank topographic measurements) over a three-year period with the goal of having a robust data set necessary to develop and test new computational tools. The Mad River watershed in the Lake Champlain Basin in Vermont served as the primary study location and area of data collection.

Organization of Dissertation

Chapter 1 provides background and literature review on catchment sediment studies, event sediment dynamics, digital photogrammetry with UAS, and the application of artificial neural networks (ANNs) in hydrology and sediment studies. ANNs collectively are a considerably varied set of data-driven tools with applications existing in many subject areas. Background on two fundamental types of networks used in this dissertation, the

counterpropagation neural network (CPN) and restricted Boltzmann machine (RBM), are presented in more detail.

Chapter 2 focuses on the measurement of streambank surfaces and erosion utilizing survey data collected along the Mad River, New Haven River and Winooski River between 2015 and 2016. It evaluates the potential of digital photogrammetry using an UAS platform to measure the topography and movement of streambanks. Topographic surveys from terrestrial laser scanning (TLS) and traditional GPS ground surveys are used for comparison and evaluation of the UAS photogrammetry technology.

Chapter 3 presents a complimentary analysis of measurement of streambank erosion by comparing differences in multi-temporal digital elevation models (DEMs) generated form UAS surveys. Additionally, existing airborne lidar survey data is used to determine multi-year estimates of channel movement. The analysis of river characteristics suitable for UAS photogrammetry is discussed as well as a summary of fieldwork efforts necessary to complete the UAS survey.

In Chapter 4, a restricted Boltzmann machine (RBM) is applied to the classification of hysteresis patterns in instream suspended sediment-discharge relationships using a collection of individual storm events. The high-resolution data collection enabled the common hysteresis classification system in the literature to be expanded. The expanded hysteresis classification process was then automated using the RBM algorithm with 2-D hysteresis images as input data. Finally, the results helped infer event sediment dynamics within the Mad River watershed and its tributaries. This presents a new approach to analyzing hysteresis patterns by using images of hysteresis patterns as inputs to a model.

Chapter 5 demonstrates the application of a recurrent counterpropagation neural network (rCPN) for predicting suspended sediment discharge at ungauged locations using only

meteorological data. Data collected between 2013 and 2015 from a network of in-stream turbidity sensors and weather stations in the Mad River watershed are utilized for both training and testing of the method. The study utilized more than one rCPN in order to be able to predict both streamflow and sediment discharge. The model is applied to the Mad River and also two of its tributaries, Mill Brook and Shepard Brook.

Finally, chapter 6 presents a cumulative summary of the research and highlights key contributions as well as future directions for further study.

Catchment Sediment Studies

Catchment studies of the dynamics of sediment and discharge provide important information for understanding the state of hydrologic systems, ecosystem disturbances and stressors within watersheds, and the implications for downstream water quality. Suspended sediment is of primary concern since it plays a critical role in sediment pollution, water-quality degradation, and impairment of ecosystems (Gao 2008). The association of suspended sediment and sediment-bound nutrients such as particulate phosphorus motivates the need to better characterize watershed sediment dynamics to understand nutrient loading and potential risks to aquatic ecosystems such as eutrophication (Walling et al. 2008). Studies of watershed sediment dynamics have included identification of sediment sources and connectivity (Abban et al., 2016; Fryirs, 2013; Walling et al., 2008), quantification of sediment loading and yield (Harrington & Harrington, 2013; Uhrich & Bragg, 2003; Warrick, Melack, & Goodridge, 2015), creation of sediment budgets (Reid & Dunne, 1996; Walling & Collins, 2008; Weber & Pasternack, 2017), characterization of event sediment dynamics (Gao & Josefson, 2012; Seeger et al., 2004; Sherriff et al., 2016), measurement of geomorphological change (Bremer & Sass, 2012; Foucher, Salvador-Blanes, Vandromme, Cerdan, & Desmet, 2017; A. D. Tamminga, Eaton, & Hugenholtz, 2015), and modeling of sediment transport (Merritt, Letcher, &

Jakeman, 2003; Schmelter & Stevens, 2013; Stryker, Wemple, & Bomblies, 2017). In addition, the impact of climate change on sediment processes is currently an area of active research (Bussi, Francés, Horel, López-Tarazón, & Batalla, 2014; Gomez, Cui, Kettner, Peacock, & Syvitski, 2009; Goode, Luce, & Buffington, 2012).

Measurement of sediment processes within a catchment often starts with monitoring suspended sediment at key locations in the river network (i.e. watershed or subwatershed outlet). Analysis of suspended sediment concentrations yields information on event sediment dynamics and allows for quantification of sediment loads and yields. Suspended sediment concentrations can be determined using water quality sampling, indirect measurement of surrogates such as turbidity, use of calibrated sensors such as acoustic Doppler current profilers (Gao, 2008; Gray & Gartner, 2009), and remote sensing (Wass, Marks, Finch, Leeks, & Ingram, 1997). Fluvial sediment concentration data are then often combined with volumetric sediment deposition and erosion measurements calculated using topographical surveying methods to provide a more complete picture of sediment processes. The topographic measurements of erosion and deposition may often be accomplished using traditional bank pin and ground survey techniques, (e.g. Lawler et al. 1999), aerial and ground-based lidar surveying (e.g. Perroy, Bookhagen, Asner, & Chadwick, 2010), or photogrammetry (e.g. Miřijovský and Langhammer 2015). An additional component in many sediment studies is the characterization of the sediment composition using sediment fingerprinting in order to categorize the source (Abban et al., 2016; Davis & Fox, 2009; Allen C. Gellis & Mukundan, 2013; Walling et al., 2008).

Due to the resource intensive monitoring of sediment movement, many catchment sediment studies use models to forecast or predict sediment transport and discharge. Modeling efforts have included traditional empirical relationships such as the sediment rating

curve (SRC) and universal soil loss equation (USLE) (Gao 2008). More recent development of physics-based models simulate watershed sediment processes that include simulate sediment transport, hill slope processes, and streambank erosion (Aksoy & Kavvas, 2005; Borah & Bera, 2004; Merritt et al., 2003; Stryker et al., 2017). In studies, where suspended sediment load prediction is the primary objective, data-driven models have grown in popularity and include multiple linear regressions, artificial neural networks, genetic programming, and Bayesian methods (Mount & Abrahart, 2011; Mount & Stott, 2008; Solomatine & Ostfeld, 2008). Sediment modeling using artificial neural networks, in particular, is discussed in more detail in following sections.

Measurement of streambank erosion

Streambank erosion represents a large portion of the overall sediment and phosphorous loading to river systems in Vermont (Langendoen, Simon, Klimetz, Natasha, & Ursic, 2012) and is important to quantify as part of comprehensive catchment sediment studies. Several methods exist to measure and monitor streambank erosion and retreat. Traditional direct measurement methods include cross-sectional surveys and bank pins (Foucher et al., 2017; Lawler, 1993; Lawler et al., 1999). The availability of both airborne and terrestrial lidar (laser scanning) has resulted in more comprehensive and detailed measurement of bank movement. Airborne lidar has been used to quantify geomorphic change along river corridors (Grove, Croke, & Thompson, 2013; Thoma, Gupta, Bauer, & Kirchoff, 2005) as well as hillslopes and gullies (Perroy et al., 2010; Tseng et al., 2013). Similarly, the availability of terrestrial laser scanners (TLS) allowed for quantification of bank erosion at site specific scales at extremely high spatial resolution (O'Neal & Pizzuto, 2011; Resop & Hession, 2010). Longer term (multiple years or decades) estimates of streambank erosion rates have been successful through combining airborne lidar and historical aerial photos (De Rose & Basher,

2011; Garvey, 2012; Rhoades, O'Neal, & Pizzuto, 2009). A common approach for quantifying geomorphological change involves the creation of digital elevation models (DEMs) from sequential surveys and then subtracting the later DEM from the earlier DEM; the resulting difference represents land elevation change between the two survey dates. This approach has been utilized with survey data collected using photogrammetry, airborne lidar, and TLS (Bremer & Sass, 2012; Grove et al., 2013; Milan, Heritage, & Hetherington, 2007; O'Neal & Pizzuto, 2011; Perroy et al., 2010; Tseng et al., 2013). Recently, advances in the development of digital photogrammetry methods and unmanned aircraft systems (UAS) platforms have resulted in a resurgence of photogrammetry being used to generate topographic data and detect geomorphic change (Colomina & Molina, 2014; Cook, 2017; Westoby, Brasington, Glasser, Hambrey, & Reynolds, 2012).

Digital Photogrammetry and Unmanned Aerial Systems (UAS)

The rapid advancement of UAS technology, also referred to as unmanned aerial vehicles (UAVs) or drones, offers the ability to overcome some of the existing data collection shortcomings of ground surveys and manned aircraft. While DEMs and contours from aerial photography using photogrammetric methods has been available for decades, recent advances in image processing software, driven in part by innovations in computer vision and structure from motion (SfM) and multi-view stereo (MVS) photogrammetric algorithms, have rapidly advanced the resolution of UAS topographic data using basic camera technology at far lower costs compared to traditional aircraft aerial imagery. SfM is ideally suited for processing photos with a high degree of overlap taken from a wide variety of positions (i.e. a moving sensor) (Westoby et al., 2012). Originally developed by the computer vision field during the 1990s, SfM and variations have become widely available in desktop software packages such as Agisoft PhotoScan, Pix4D, and Microsoft Photosynth. Digital photogrammetric methods such as SfM

are applicable to imagery collected using any platform, including handheld smartphone cameras (Micheletti, Chandler, & Lane, 2015), but have been widely adopted to process imagery collected using UAS (C. H. Hugenholtz et al., 2013).

Numerous UAS based photogrammetric surveying applications exist; recent reviews by Colomina and Molina (2014), Watts (2012), and Whitehead et al. (2014) highlight UAS characteristics and applications in photogrammetry and remote sensing. Fluvial study applications include mapping bathymetry (Lejot et al., 2007), channel topography (Miřijovský, Michalková, Petyniak, Máčka, & Trizna, 2015; A. Tamminga, Hugenholtz, Eaton, & Lapointe, 2015; Woodget, Carbonneau, Visser, & Maddock, 2015) and production of very high resolution DEMs (Whitehead & Hugenholtz, 2014; Micheletti et al., 2015; Neugirg et al., 2016). In addition, UAS-derived data have shown potential in quantifying bank erosion and monitoring volumetric change in fluvial settings due to flooding (Miřijovský & Langhammer, 2015; Miřijovský et al., 2015; A. D. Tamminga et al., 2015). However, to date, UAS investigations have focused on monitoring change over relatively short sections (<1 km) of river channels and have insufficient comparison to other methods such as TLS. The comprehensive evaluation of high-resolution UAS topographic data along various river types and settings remains an area in need of further study (Woodget et al., 2015; Carbonneau & Dietrich, 2017).

Event Sediment Dynamics

Event sediment dynamics refer to the various processes of suspended sediment transport in watersheds during hydrological events. Information on the mechanisms controlling this sediment transport may be inferred from storm-runoff response and the corresponding sediment concentration response (Asselman, 1999; Gao & Josefson, 2012; Lefrançois, Grimaldi, Gascuel-Odoux, & Gilliet, 2007; Sherriff et al., 2016). The most practical

and common method for capturing event sediment dynamics is to simultaneously monitor discharge and sediment concentration, either through direct sampling or surrogate monitoring, and then compare how discharge and concentration vary over the event. Due to the non-linear processes controlling sediment transport, the relationship of suspended sediment and discharge over a single storm event often cannot be described by a simple linear or log-linear relationship (Onderka, Krein, Wrede, Martínez-Carreras, & Hoffmann, 2012). This has given rise to a robust study over the last three decades of the hysteretic nature of the suspended sediment-discharge relationship for hydrological events (Aich, Zimmermann, & Elsenbeer, 2014; Asselman, 1999; Duvert et al., 2010; C. Evans & Davies, 1998; Gao & Josefson, 2012; Lefrançois et al., 2007; Seeger et al., 2004; Sherriff et al., 2016; Williams, 1989).

The sediment-discharge hysteresis patterns that result from different peak times and shapes of the hydrograph and sedigraph were first comprehensively described by Williams (1989). Further research has attempted to connect patterns of hysteresis to watershed characteristics such as sediment availability, watershed size, and antecedent conditions (Asselman, 1999; Duvert et al., 2010; Gao & Josefson, 2012; Seeger et al., 2004). Hysteresis analysis has also been used to estimate the contribution of bank erosion caused by livestock in a catchment in France (Lefrançois et al., 2007); and the characteristics of event storm dynamics including hysteresis patterns have been shown to be connected to seasons and previous storm events (Gao & Josefson, 2012; Lefrançois et al., 2007). Due in part to the limited temporal resolution and variability frequently present in suspended sediment data, the majority of studies have used a subjective visual classification/interpretation of the basic pattern of hysteresis (i.e. clockwise, counterclockwise, figure-eight loop) similar to the patterns described by Williams (1989). An alternative approach compresses the hysteresis information into a metric (i.e., a continuous variable) that may be used in additional statistical methods.

Smith and Dragovich (2009) investigated a dimensionless similarity function to quantitatively determine the type of hysteresis present in the sediment concentration-discharge plot; several other indices have been proposed (Lawler, Petts, Foster, & Harper, 2006; Lloyd, Freer, Johnes, & Collins, 2016b; Zuecco, Penna, Borga, & van Meerveld, 2016). The use of hysteresis indices has enabled use in additional statistical methods resulting from the creation of a continuous variable. However, a limitation of currently available hysteresis indices is the loss in shape of the hysteresis plots when compressing the combined time-series data into a single metric, with it being noted that different shapes can have the same value of hysteresis index (Lloyd, Freer, Johnes, & Collins, 2016a). The development of more advanced classification and pattern recognition tools applied to suspended sediment hysteresis analysis is an area in need of further research.

Artificial Neural Networks

Artificial neural networks (ANNs) are one method, in the larger field of machine learning, capable of identifying complex non-linear relationships and patterns in large data. In general, ANNs may be considered nonparametric statistical tools whose structure is inspired by the design of the human brain and nervous system. They may be characterized by three main features: a set of artificial neurons (or nodes); connectivity among the nodes representing the architecture of the network; and an algorithm for determining the weights of the connections, usually called training or learning algorithms (De Castro, 2006). The primary differences between types of artificial neural networks lies in the variety of their architecture and learning algorithms.

The multilayer perceptron (MLP) neural network is a common network architecture in which the signals propagate through the network layers in a forward direction (referred to as feedforward) (De Castro, 2006). In this typical ANN architecture, the most common

learning algorithm for updating the weights is the error backpropagation algorithm popularized by Rumelhart et al. (1986). The feedforward backpropagation (FFBP) algorithm adjusts the network weights by propagating errors backwards through the network using a least-square error gradient descent error correction rule. The FFBP ANN is the most popular application over the last three decades in many fields due to its simple architecture and ability to operate as a universal function approximator (De Castro, 2006). The error backpropagation learning algorithm falls into the category known as *supervised learning* because it uses a set of inputs and known (measured) outputs (this combined input and output data are known as training data) to adjust the weights until differences between the predicted network outputs and known observations are minimized. Once trained, the learning algorithm ceases; the initial weights are fixed; and the algorithm may be used to interpolate or predict a non-linear mapping given measured input data not seen before (i.e. data not used for training). In addition to the common FFBP network, other types of neural networks and learning algorithms have found application in the environmental fields.

Counterpropagation Neural Network (CPN)

The counterpropagation neural network (CPN) is an ANN that combines elements of unsupervised and supervised learning. The CPN is a purely data-driven network and self-adapts to learn nonlinear mappings between predictor inputs and a set of response vectors or classes. The CPN combines two ANN algorithms – a Kohonen self-organizing map (hidden layer) and a Grossberg ouster structure (output layer) (Hecht-Nielsen, 1988). The CPN is most commonly utilized in a specific configuration referred to as forward only or forward mapping CPN in which one is not interested in the inverse relation of inputs to a predictor. Details of the CPN algorithm are presented in Chapter 4.

While not nearly as widely used in studies as FFBP, CPNs have found many applications in science and engineering due in part to a fast learning rate and its usefulness in classification and prediction. Among others, it has been used for facial recognition (Adeyanju, Awodoye, & Omidiora, 2016), chemical toxicity determination (Drgan, Župerl, Vračko, Cappelli, & Novič, 2017), and forecasting freeway travel time (Dharia & Adeli, 2003). CPNs have also found application in hydrological studies as both classifiers and forecasting tools. Besaw et al. (2009) utilized a CPN for classification of stream geomorphic conditions. It has also been used in groundwater applications to characterize aquifer properties (Rizzo & Dougherty, 1994).

The CPN has also been demonstrated to be capable of forecasting flow in rainfall-runoff modeling. Chang and Chen (2001) combined the CPN with fuzzy logic for real-time streamflow prediction and Coulibaly and Evora (2007) utilized the same type of network for filling in gaps in weather station data. Besaw et al. (2010) evaluated the CPN to forecast streamflow in ungauged basins, and showed the usefulness of modifying the network structure to have "memory" by using lagged inputs and/or having a recurrent feedback connection. Similar approaches have been used in other studies utilizing CPN or other ANNs for forecasting time series data (F.-John Chang, Chang, & Huang, 2002; Coulibaly & Evora, 2007).

Restricted Boltzmann Machine (RBM)

The restricted Boltzmann machine (RBM) (Smolensky, 1986), also referred to as harmoniums, is a type of artificial neural network used for unsupervised learning. It has experienced an increase in popularity for its extended use as a classification model (Larochelle, Mandel, Pascanu, & Bengio, 2012) and more recently its use in deep learning applications (Hinton et al. 2012). The RBM is a stochastic, generative model that consists of two layers: a layer of visible nodes (input data) and a layer of hidden nodes. After training, the RBM

provides a representation of the training data; and its hidden layer may be considered a type of feature extractor. Details of the RBM network are described in Chapter 3.

While RBMs are useful for feature extraction or image generation after initial training; in many applications, they may be further extended to a classifier model. This configuration has been demonstrated in the classification of handwritten digits using the benchmark MNIST data set (Fischer & Igel, 2014), the classification of articles into newsgroups (Larochelle et al., 2012), and the classification of video segments (J. Yang, Liu, Xing, & Hauptmann, 2007). An advantage of RBMs over some other classification networks is that the initial training is unsupervised learning and does not require labeled data. The learning embedded in the trained hidden nodes can be leveraged to obtain information on inherent structure within data sets; in this is case, the hidden nodes act as feature detectors and/or a filter (Fischer & Igel, 2014; Testolin, Stoianov, De Filippo De Grazia, & Zorzi, 2013). RBMs are also frequently used in deep learning applications, where they form building blocks of larger deep learning networks. Most commonly a stack of RBMs are connected in a network commonly referred to as deep belief neural networks (DBNNs) (Hinton et al. 2006) and used in big data and machine learning applications (Hinton et al., 2012; Hinton & Salakhutdinov, 2006; O'Connor, Neil, Liu, Delbruck, & Pfeiffer, 2013a; Testolin et al., 2013). While not utilized yet in hydrological applications, RBMs and deep learning represent a promising tool that links state-of-the-art machine learning and modern models of Bayesian inference (O'Connor et al., 2013a).

Applications of ANNs in Hydrology

There has been widespread use of ANNs in the hydrology field for the past two decades that includes applications to model hydrology, rainfall-runoff, and water resource variables (Abrahart et al., 2012; Govindaraju & Rao, 2000). Initial studies on the applicability of ANNs to river forecasting almost exclusively utilized feedforward backpropagation (FFBP)

networks. More recent studies have investigated the application of novel or new features of the FFBP network and the comparison of different types of neural networks (Abrahart et al. 2012). A continued need for applied ANN research exists in the prediction of streamflow in ungauged streams because the majority of streams in the world are ungauged (Besaw et al., 2010; Razavi & Coulibaly, 2013).

ANNs and other non-parametric statistical methods have been shown to capture the complex non-linear aspects of sediment transport better than conventional models and thus, are an attractive alternative to complex physics-based models of sediment transport (Kisi & Shiri, 2012). Water resource managers are often interested in predicting and estimating suspended sediment in rivers where detailed hydraulic data are not available and only flow and/or precipitation data exists. A large amount of research has been done on prediction of streamflow in ungauged basins and how hydrological models can be transferred (regionalization). Razavi and Coulibaly (2013) recently reviewed regionalization methods for models including data-driven methods and found that consideration of climate and catchment type will impact the method used. In temperate climates, regionalization methods using linear and non-linear regression on catchment attributes have been effective, independent of the hydrological model (Razavi & Coulibaly, 2013).

Applications of ANNs to Sediment Prediction

The estimation of suspended sediment concentration or load at a location in a watershed is often a necessary component of catchment sediment studies. Empirical methods such as creation of sediment rating curves (SRCs) using simple linear or multiple regression models have been used extensively for decades to predict sediment concentration from streamflow measurements and continue to be used today. A variety of other methods have been developed including physics-based, conceptual, and data-driven models. However, due

in part to the extensive data collection requirements and complex physics associated with sediment transport, data-driven methods may be more appropriate for sediment yield forecasting (Abrahart, See, Heppenstall, & White, 2008; Merritt et al., 2003). The non-linear relationship of sediment concentration to discharge and other variables, have motivated the increased use of machine learning and genetic programming methods (Mount and Abrahart 2011).

The first published applications of ANNs to sediment prediction appeared in 2001. Abrahart and White (2001) utilized a feedforward backpropagation (FFBP) network to predict catchment soil loss using precipitation as inputs and compared the FFBP method against a multiple linear regression (MLR) model. Jain (2001) also used FFBP to predict suspended sediment concentration (SSC) using time-lagged measurements of sediment concentration, stage, and discharge, and found better prediction compared to traditional SRCs. Nagy et al. (2002) modeled SSC using a FFBP network and river hydraulic parameters as input data; they found the FFBP results comparable to, and in some instances more accurate than, SSC results generated using the empirical formulas. The FFBP methodology utilized all available parameters, thus avoiding the simplification of conventional empirical models. This FFBP approach has also been applied to prediction of bed-load transport and total load sediment transport (B. Kumar, 2012; C. T. Yang, Marsooli, & Aalami, 2009).

A variety of ANN algorithms have since been applied successfully to estimating suspended sediment in river systems using a variety of hydrometeorological variables and antecedent sediment data as inputs. The most common approach for estimating SSC or suspended sediment load (SSL) continues to be the FFBP algorithm, or a variation thereof, using a combination of discharge and antecedent sediment data as inputs (Afan, El-shafie, Mohtar, & Yaseen, 2016). Variations on the traditional FFBP have include coupling FFBP

with adaptive neuro-fuzzy inference (e.g. Rajaee, Mirbagheri, Zounemat-Kermani, & Nourani, 2009) and wavelets (e.g. Liu, Shi, Fang, Zhu, & Ai, 2013). Other applications of ANN and machine learning methods for estimating SSC or SSL include the generalized regression neural network (GRNN) (e.g. Cigizoglu & Alp, 2006), radial basis function (RBF) network (e.g. Alp & Cigizoglu, 2007), support vector machines (SVMs) (e.g. Kakaei Lafdani, Moghaddam Nia, & Ahmadi, 2013), and genetic programming (e.g. Kisi & Shiri, 2012). All of these applications utilize previous observations of suspended sediment as inputs, limiting the application of the model to sites that have measured suspended sediment data. The use of time-lagged *measured* suspended sediment data as input data continues to be the most common approach in ANN sediment prediction studies (Buyukyildiz & Kumcu, 2017; Joshi, Kumar, & Adhikari, 2016; D. Kumar, Pandey, Sharma, & Flügel, 2016; Olyaie, Banejad, Chau, & Melesse, 2015; Zounemat-Kermani, Kişi, Adamowski, & Ramezani-Charmahineh, 2016), despite Abrahart et al. (2008) observing that it makes no operational sense.

Estimation of suspended sediment in rivers has also been performed using FFBP and other algorithms to predict SSC or SSL only using discharge observations as an alternative to SRCs. Applications have included the use of single observations (e.g. Cobaner, Unal, & Kisi, 2009) or multiple lagged observations of discharge (e.g. Cigizoglu & Alp, 2006) as inputs. While it should be noted that fitting SRCs to log transformed data is considered best practice given the strong heteroscedasticity in the SSC data (Rasmussen, Gray, Glysson, & Ziegler, 2011), there is no standard approach used by ANN modelers, with raw data being the most commonly used. FFBP algorithms are known to be susceptible to overfitting; and a review by Mount and Abrahart (2011) found that overfitting to a few upper range data points was common, particularly in ANN sediment prediction studies that utilized raw data. They concluded that use of log-transformed data is one approach to minimize overfitting; however,

there may be little justification for using an ANN over a traditional SRC (Mount and Abrahart 2011); and the decision to predict SSC or SSL should be based on operational considerations of the model. Finally, there is also concern when using discharge to predict SSL given the strong correlation between the two variables.

Another common ANN approach to modeling SSC and SSL is to combine river discharge with rainfall or other variables as inputs. Rainfall is the most common input variable combined with discharge for suspended sediment prediction (Alp & Cigizoglu, 2007; Kisi & Shiri, 2012; D. Kumar, Pandey, Sharma, & Flügel, 2015; Nourani & Kalantari, 2010). Other inputs variables include water level (Jain, 2001), temperature (Zhu, Lu, & Zhou, 2007), and turbidity (Bayram, Kankal, & Önsoy, 2011). An alternative approach presented by Aiteh et al. (2015) used a FFBP ANN with various watershed characteristics as inputs to predict SRC parameters, which allowed for estimation of suspended sediment at multiple locations.

The prediction of suspended sediment data using ANNs in catchments that are ungauged both in terms of both discharge and sediment data is limited. Kamel et al. (2014) predicted SSL in ungauged catchments using physical watershed characteristics and climate inputs, but at the yearly average timescale. Alp and Cigizoglu (2007) evaluated two ANN models for predicting daily SSL using only antecedent rainfall, but found the model insufficient for reliable SSL prediction compared to using a model that included discharge as an input variable. In contrast, Zhu et al. (2007) and Raghouwanshi et al. (2006) investigated the prediction of daily SSL using only rainfall and temperature and found good model performance. The vast majority of ANN sediment prediction applications continue to rely on *measured* streamflow as input data, limiting their application to stream-gauged locations. Recently, Afan et al. (2016) reviewed studies that applied ANNs to the prediction of sediment

concentration and discharge and found that all of the studies used measured discharge as inputs.

Given the frequency with which current peer-reviewed suspended sediment prediction models use (1) *measured* streamflow and antecedent sediment data as model input, (2) FFBP networks that are overfit, and (3) a low temporal resolution for prediction (i.e., daily and monthly intervals), there remains a need for research in this area, specifically applications that can address prediction in ungauged river systems. The availability of high temporally resolved turbidity data capable of being monitored at multiple locations on small rivers provides a new means and promise for training and validating a sediment prediction ANN methodology.

CHAPTER 2. QUANTIFYING STREAMBANK MOVEMENT AND TOPOGRAPHY USING UNMANNED AIRCRAFT SYSTEM PHOTOGRAMMETRY WITH COMPARISON TO TERRESTRIAL LASER SCANNING

Abstract

Streambank movement is an integral part of geomorphic changes along river corridors and affects a range of physical, ecological, and socio-economic systems including aquatic habitat, water quality, and infrastructure. Various methods have been used to quantify streambank erosion, including bank pins, ground surveys, lidar, and analytical models, however, due to high-cost or labour intensive fieldwork these are typically feasible or appropriate only for site-specific studies. Advancements in unmanned aircraft systems (UAS) and photogrammetry provide opportunities for more rapid and economical quantification of streambank erosion and deposition at variable scales. This work assesses the performance of UAS-based photogrammetry for capturing topography of streambank surfaces and quantifying bank movement. UAS data are compared to terrestrial laser scanner (TLS) and GPS surveying from streambank sites located in Vermont that featured a variety of bank conditions and vegetation. Cross-sectional analysis of data from UAS and TLS revealed that the UAS reliably captured the bank surface within 0.2 m of TLS and GPS surveys across all sites during leaf-off conditions. Mean error between UAS and TLS was only 0.11 m in early spring conditions. Dense summer vegetation resulted in decreased accuracy and was a limiting factor in the ability of the UAS to capture the ground surface. At areas with observed bank movement, the change in cross-sectional area estimated using UAS data compared reliably to TLS survey for net cross-sectional changes greater than 3.5 m2, given a 10% error tolerance. At locations with smaller changes, error increased due to the effect of vegetation,

georeferencing, and overhanging bank profiles. UAS-based photogrammetry shows significant promise for capturing bank topography and movement at fine resolutions in a flexible and efficient manner.

Introduction

Measuring the rate and extent of bank movement is a primary component of fluvial studies, as these changes impact a wide range of physical, ecological, and socio-economic systems such as aquatic habitat, water quality, and on- and near-stream property and infrastructure (Daly, Miller, & Fox, 2015; Simon & Rinaldi, 2006). For example, streambank erosion is estimated to account for 30–80% of sediment loading into waterways (D. J. Evans, Gibson, & Rossell, 2006; Fox et al., 2007; Simon & Rinaldi, 2006). Additionally, this sediment can be a large source of nutrient loading to receiving waters (Langendoen et al., 2012). As such, management strategies, including total maximum daily load studies by the U.S. Environmental Protection Agency, require estimates of bank erosion within a river system (Collins & Walling, 2004).

Several indirect and direct methods for monitoring or estimating bank movement are only appropriate at specific spatial and temporal scales. Indirect predictive tools include process-based models and empirical/analytical models (Chen & Duan, 2006). The latter aim to predict equilibrium channel width using either regime equations developed through regression of field data (e.g., Eaton, 2006) or external hypotheses based on assumptions (e.g., alluvial channels attain equilibrium when unit stream power reaches a maximum or minimum threshold; Lai et al., 2015). The process-based methods include analytical slope stability modelling based on the limit equilibrium method (e.g., Darby & Thorne, 1996; Osman & Thorne, 1988) and often employ computer programs such as SLOPE/W (e.g., Borg,

Dewoolkar, & Bierman, 2014; Dapporto, Rinaldi, Casagli, & Vannocci, 2003) and BSTEM (e.g., Langendoen et al., 2012; Simon, Curini, Darby, & Langendoen, 2000). Although these models are flexible in application, they rely heavily on characterization of relevant soil properties and site conditions (e.g., soil classification, unit weights, shear strength parameters, soil suction, root strengths, etc.) and require extensive fieldwork because procedures are time intensive and have to be repeated at multiple locations due to the spatial heterogeneity of soils (e.g., Borg et al., 2014; Simon et al., 2000). Direct methods for monitoring streambank movement often include the use of repeat plan form or cross-sectional surveys, bank erosion pins, and historical photography (Lawler, 1993); and although these methods are valuable for estimating sediment loads in small watersheds, they are labour intensive and typically only practical for site-specific studies (Resop & Hession, 2010).

More recently, bank erosion has been measured using technologies such as lidar and advanced photogrammetry. For estimating streambank movement at larger watershed scales, high-resolution topographic data and digital elevation models (DEMs) derived from airborne and satellite imagery photogrammetry and aerial lidar can provide a reliable means to monitor channel change and bank erosion (De Rose & Basher, 2011; Garvey, 2012; Hughes, McDowell, & Marcus, 2006; Reinfelds, 1997). Some have raised concerns that topographic data, derived from remote sensing, can be cost and time prohibitive and lack the necessary temporal or spatial resolution for studying fluvial systems (Hugenholtz et al., 2013). However, recent advances in fine-scale topographic data-collection systems, such as terrestrial laser scanners (TLSs) and digital photogrammetry, allow increased flexibility in the scale (both spatial and temporal) at which fluvial data may be collected. One example is the creation of repeat DEMs that enable straightforward quantification of volumetric change along river

corridors due to erosion and deposition (Milan, Heritage, & Hetherington, 2007; Tseng et al., 2013; Wheaton, Brasington, Darby, & Sear, 2010).

Modern TLSs, introduced in the early 1990s, have seen widespread adoption in fluvial applications (Brasington, Vericat, & Rychkov, 2012; Brodu & Lague, 2012; Hohenthal, Alho, Hyyppa, & Hyyppa, 2011; Lague, Brodu, & Leroux, 2013; Molina, Rodríguez- Gonzálvez, Molina, González-Aguilera, & Espejo, 2014; Williams et al., 2011). Currently, available TLS systems have scanning distances that range from 0.1 to 4,000 m and operate at subcentimetre accuracies (Hohenthal et al., 2011). Given this flexible range and the ability to scan thousands of data points per second, TLS can generate very fine-scale topographic data quickly and at the most precise levels (Molina et al., 2014). Fluvial TLS applications have included streambank retreat measurement (O'Neal & Pizzuto, 2011; Resop & Hession, 2010), gully erosion (Jaboyedoff et al., 2009; Perroy, Bookhagen, Asner, & Chadwick, 2010), mapping of hydraulic biotopes (Milan, Heritage, Large, & Entwistle, 2010), and fine-scale topography of riverbeds and channels (Bangen, Wheaton, Bouwes, Bouwes, & Jordan, 2014; Brodu & Lague, 2012; Heritage & Milan, 2009; Hodge, Brasington, & Richards, 2009). Despite the applicability of TLS for high-resolution data collection in river systems, the equipment remains quite expensive; post-processing of data is labour intensive and is limited by water absorption of near-infrared lasers, and data collection within river channels may be difficult depending on the flow conditions.

The rapid advancement of unmanned aerial system (UAS) technology, also referred to as unmanned aerial vehicles or drones, offers the ability to overcome some of the existing data-collection shortcomings. Production of DEMs and contours from aerial photography using photogrammetric methods has been available for decades. However, recent advances in

UAS platforms and image-processing software, driven in part by innovations in computer vision and the development of structure from motion (SfM) and multiview stereo photogrammetric approaches, have helped advance the resolution of UAS topographic data using basic camera technology at far lower costs compared to traditional aircraft aerial imagery. Recent reviews by Colomina and Molina (2014); Watts, Ambrosia, and Hinkley (2012); and Whitehead et al. (2014) highlight characteristics and UAS applications in photogrammetry and remote sensing. The use of nonmetric photogrammetry methods (i.e., SfM) has been compared to other survey methods in numerous settings (James & Robson, 2012; Javernick, Brasington, & Caruso, 2014; Smith & Vericat, 2015). A meta-analysis performed by Smith and Vericat (2015) found SfM-based photogrammetry reliable over different spatial scales with errors proportional to the range (altitude) of data collection.

The increased technology and flexibility of UAS-based photogrammetry has quickly led to applications in the environmental fields including fluvial geomorphology. Very high-resolution UAS-based imagery (at <5-cm resolution) has been used to map bathymetry (Lejot et al., 2007), channel topography (Miřijovsky , Michalková, Petyniak, Máčka, & Trizna, 2015; Tamminga, Hugenholtz, Eaton, & Lapointe, 2015; Woodget, Carbonneau, Visser, & Maddock, 2015) and to produce very high resolution DEMs (Micheletti, Chandler, & Lane, 2015; Neugirg et al., 2016; Whitehead & Hugenholtz, 2014). In addition, UAS-derived data have shown potential in quantifying bank erosion and monitoring volumetric change in fluvial settings due to flooding (Miřijovský & Langhammer, 2015; Miřijovský et al., 2015; Tamminga, Eaton, & Hugenholtz, 2015). However, to date, UAS investigations have focused on monitoring changes over relatively short sections (<1 km) of river channels and have insufficient comparison to other methods such as TLS. The comprehensive evaluation of

high-resolution UAS topographic data along various river types and settings remains an area in need of further study (Carbonneau & Dietrich, 2017; Woodget et al., 2015) and is the focus of this manuscript. There is a continued need to obtain channel geometry data over a large spatial expanse because it is still standard practice when modelling river flows (e.g., HEC-RAS) and bank stability (e.g., BSTEM) to use cross-sectional survey data as inputs (Buchanan, Walter, Nagle, & Schneider, 2012; Cook & Merwade, 2009). This survey data are used for development and calibration of the models and often require large amounts of human resources for field collection efforts.

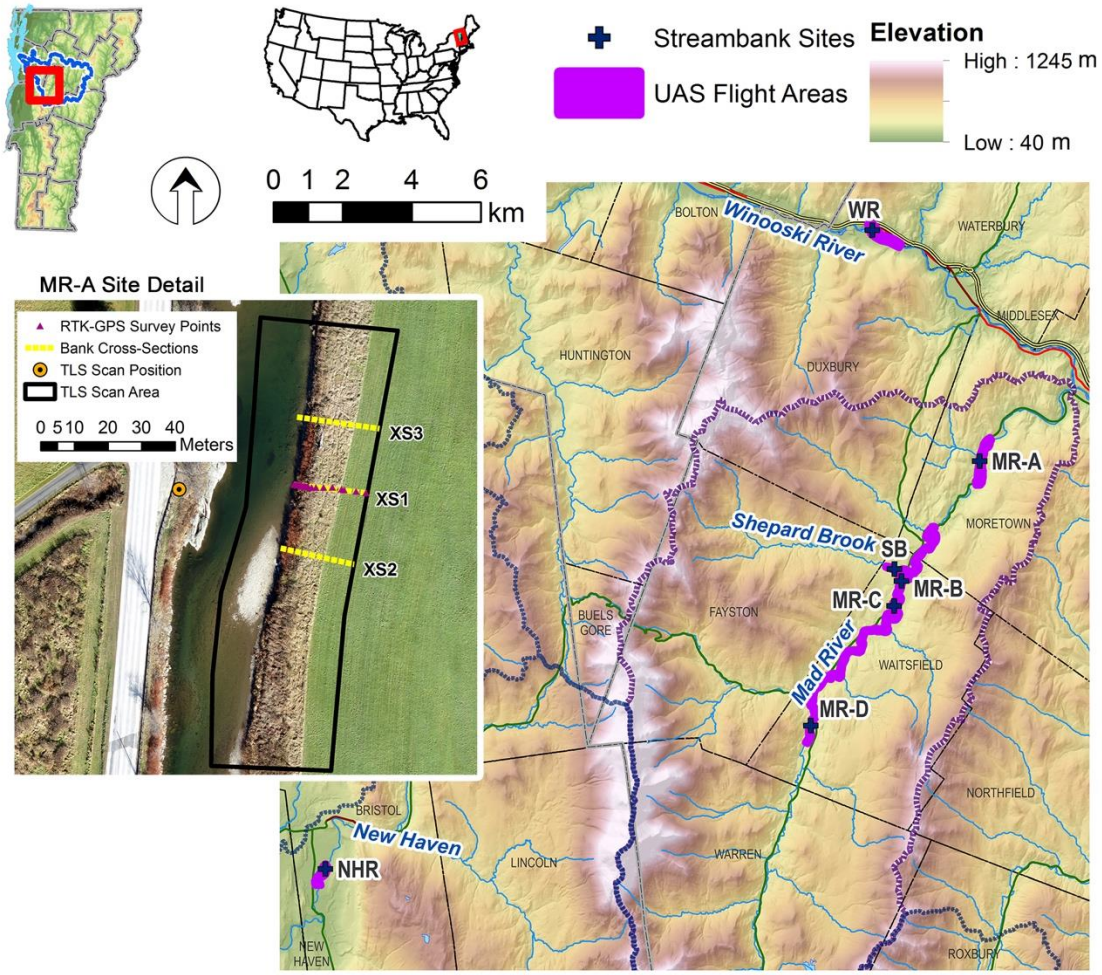
In this study, we evaluate opportunities for the reliable capture of topographic data using a fixed-wing UAS platform capable of collecting data covering many kilometres of river length during a single field outing. Our focus is on measuring streambank change under conditions of interest to agencies responsible for monitoring river systems (i.e., UAS settings suitable for data collection at the scale of river networks rather than individual stream reaches). We compare the data collected along streambank cross sections with TLS and realtime kinematic (RTK)-GPS measurements at sites, to assess the accuracy in capturing bank topography and quantifying bank movement. We also describe current limitations in UAS technology development within the context of streambank erosion and deposition monitoring applications.

Methods

Study Site

The project study area is in central Vermont within four watersheds that are part of the Lake Champlain Basin. Our seven study sites were located along the Winooski River, Mad

River (MR), Shepard Brook, and New Haven River (NHR; Figure 2.1), and the total UAS flight area covered approximately 20 km of river corridor. Whereas forested areas dominate



Site Name	River	Bank Height (m)	Drainage Area (km^2)	Bankfull Channel Width (m)	Bank Soil Type	Vegetation	Erosion Sensitivity
WR	Winooski	3.7	2,121	62 - 88	Fine sand	Tall Grass	High
MR-A	Mad	2.8	347	38 - 40	Fine sandy loam	Grass / brush	Low
SB	Shepard	1.4	44	22 - 25	Silt loam	Tall Grass	Medium
MR-B	Mad	2.2	247	25 - 30	Very fine sandy loam	Heavy brush & trees	Medium
MR-C	Mad	2.1	245	25 - 30	Fine sandy loam	Brush & trees	Low
MR-D	Mad	2	145	42 - 60	Fine sandy loam	Grass / brush	High
NHR	New Haven	1.9	183	30 - 45	Very fine sandy loam	Grass / brush	High

Figure 2.1 Map of project area showing locations of streambank monitoring sites and sections of river corridors flown with UAS along with detail of MR-A site and site characteristics. RTK = real-time kinematic; TLS = terrestrial laser scanner; UAS = unmanned aircraft system

the study area watersheds, the project sites are located across a variety of land uses and geologic settings including small towns and agricultural areas on the valley floor. Site selection was based in part on the desire to (a) have a variety of bank soil types, vegetation, bank heights, and upstream drainage (Figure 2.1), and (b) ensure a range of sensitivity to streambank movement (i.e., to have some sites with noticeable erosion and others that remain relatively stable) over 3-year field observations.

The Winooski River flows through alluvial soils on the valley floor, and the watershed encompasses a larger area that includes the MR watershed. The geological setting of the MR watershed is dominated by glacial tills and features evidence of glaciation including kame terrace deposits, moraines, outwash areas, and lake sediments. In sloped areas, the soils are predominantly rocky tills, and on the valley floors, fine sandy loams and silty loams are common. The NHR watershed similarly features glacial till-blanketed slopes in the headwaters, but in the lower watershed, the NHR traverses the broad Champlain Valley through alluvial soils. Streambanks in the study area typically featured a cohesive fine sandy loam or silt loam layer ranging from 1 to 2 m overlaying a loose gravel layer. Active eroding sections typically feature steep banks with failure occurring from undercutting of the cohesive layer.

The MR, NHR, and Winooski River all have a history of flooding that dates back to early settlements along the river corridor when historical deforestation of the watershed resulted in increased delivery of sediment to the river channel in the 18th and 19th centuries (Fitzgerald & Godfrey, 2008; Underwood, 2004). During the 20th century, river channel management activities included straightening, dredging, and bank armouring that resulted in increased flood flow capacity and decreased access to floodplains (Fitzgerald & Godfrey, 2008), increasing the river corridors' susceptibility to erosion and channel movement. As these

rivers continue to adjust, data collected using UAS methods regarding the current state of the river and its geomorphology are useful for broader watershed studies.

Data Collection

In this study, a senseFly eBee fixed-wing UAS (Figure 2.2a) surveyed the river corridor study sites. The study employed two eBee models: the standard eBee and eBee RTK. The standard eBee has a basic GPS receiver and requires ground control points (GCPs) or other methods to georeference the data with submetre accuracy. The eBee RTK is a survey-grade system that features a more accurate GPS receiver that allows direct georeferencing of the data. The eBee RTK uses a virtual or local GPS base station option that can eliminate the need for GCPs. With virtual base station operation, the eBee system uses cellular connection to a continuously operating GPS base station network to obtain the most accurate (nominal accuracy of 5 cm) GPS positioning available.

Surveys occurred over four campaigns: April-May 2015, June-August 2015, November 2015, and April-May 2016. The altitude above ground level was approximately 100 m for all UAS flights. The ground sample resolution was 3.6 cm with a lateral and longitudinal image overlap of 70%. Processed UAS data resulted in point clouds with average point spacing of 12 cm. For all sites, UAS flight times ranged from 25 to 35 min and covered ~600 m of river length and 0.50 km^2 ground area. Typically, a single outing accommodated 4–5 UAS flights with companion TLS and GPS surveys completed simultaneously. Much higher data-collection resolutions are possible with UAS platforms, in particularly multirotor UASs (e.g., Woodget et al., 2015), if imagery is taken at low altitudes (< 25 m) to increase the ground pixel resolution. However, the increased collection times become impractical with current technology compared with a fixed-wing UAS when operating as part of a river network-wide

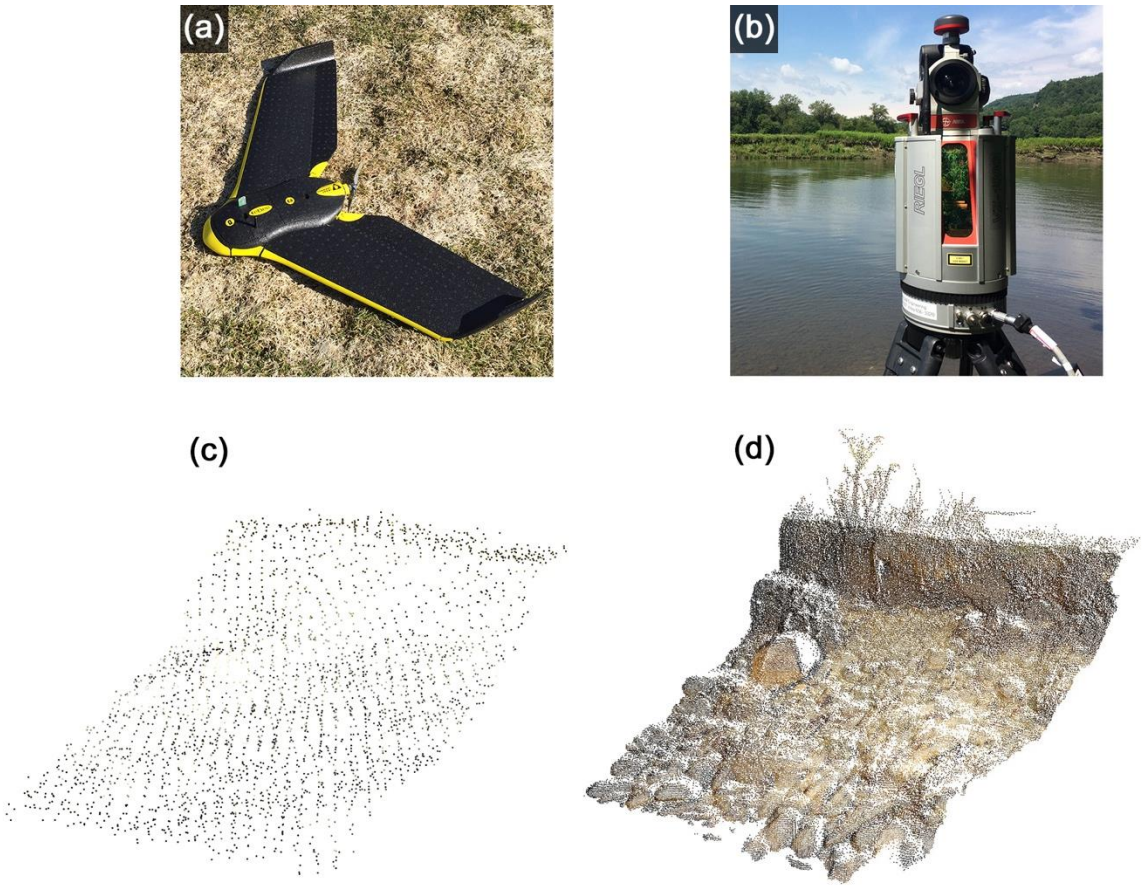


Figure 2.2 Surveying systems used in the study: (a) senseFly eBee UAS; (b) RIEGL VZ- 1000 terrestrial laser scanner; and section of raw point cloud data along streambank at Shepard Brook site from (c) unmanned aircraft system flight and (d) terrestrial laser scanner scan collected on November 10, 2015

analysis. River flows during UAS-flights were generally below average based on stream-gage records, but occasionally high enough to be unsafe for wading and collection of traditional ground-based surveying. GPS and TLS data were collected on the same day as UAS flights, except when river levels made ground surveying inaccessible. During spring and autumn, UAS flights and TLS scans occurred in "leaf-off" conditions, whereas summer data collection had denser vegetation. The first flights occurred in spring 2015 and were collected with the eBee; all subsequent flights used the eBee RTK.

UAS performance was assessed at selected streambanks using a RIEGL VZ-1000 TLS (Figure 2.2b) to collect detailed ground survey data. The extent of the TLS survey varied by site ranging anywhere from 50 to 300 m of river length. The point density of TLS streambank survey data varied, but typically averaged less than 1 cm. When multiple scan positions were needed, reflector targets were used for data registration.

When the virtual base station was not used, the UAS data were georeferenced using a minimum of four GCPs and surveyed with a TopCon HiperLite + GPS receiver. Using a small number of GCPs with a direct georeferencing UAS such as the eBee models has been found to be sufficient for removing any overall bias in the positioning of the data (e.g., Carbonneau & Dietrich, 2017). We confirmed this at one of the NHR site flights by deploying 10 GCPs but using only four in the photogrammetric processing and reserving six as checks on the positional accuracy. The mean positional errors of the six reserved GCPs were 0.003, −0.006, and 0.04 m in the x, y, and z directions, respectively. These performance accuracies are comparable to those reported for the eBee RTK system (Hugenholtz et al., 2016; Roze, Zufferey, Beyeler, & McClellan, 2014) and therefore justified our approach of using a smaller number of GCPs in combination with direct georeferencing. The GCP positions were collected with the GPS rover in a semikinematic ("stop-and-go") mode, and the GPS base station positions were corrected using the Online Position User Service provided by the National Geodetic Survey of the National Oceanic and Atmospheric Administration. TLS data were also georeferenced using GCPs and used to both validate the TLS bank surface surveys and assess the accuracy of UAS-derived bank profiles; ground survey points along three cross sections were collected at each streambank study site.

On June 1, 2015, a bankfull flow event in the MR caused widespread minor bank erosion. To capture the potential change, the two most active sites (MR-D and Shepard Brook) were re-surveyed on June 22, 2015, and August 26, 2015, respectively. An early spring rainfall and snowmelt event on February 26, 2016, also caused significant erosion (>1-m retreat) at the NHR project site and minor erosion along Winooski River and MR. Field observations showed that only the NHR site experienced significant bank erosion. Thus, we performed a detailed comparison of both systems and expanded the number of cross sections at the NHR site to 10 to capture bank retreats that ranged from 0 to ~13 m.

Data Processing

UAS imagery was post-processed using senseFly's eMotion software package and then passed to the Pix4D (Pix4D, Inc.) software package for photogrammetric processing. Pix4D, like other digital photogrammetric UAS solutions, has a seamless workflow that ingests UAS imagery, generates a 3D point cloud from the overlapping images, and uses the point cloud to produce an orthorectified image mosaic and raster digital surface model. TLS data processing occurred using RiScan PRO version 2.0. Tie points fixed to surveyed GPS assisted in registering multiple scan position data. Examples of the processed data from both methods (exported as LAS points clouds with RGB in the UTM Zone 18 N WGS84 coordinate system [EPSG: 32618]) are presented in Figure 2.2c,d.

Data Analysis

To assess the accuracy of the UAS-derived elevation data and validate the TLS data as ground truth data, the UAS- and TLS-derived streambank profiles were compared to GPS cross-sectional survey data collected using the RTK-GPS system during the autumn 2015 data-

collection period. The distances between the derived profiles and each GPS survey point were used to compute the mean, median, standard deviation, and root mean square error (RMSE).

At each site, the reference cross sections (labelled XS1), and two additional cross sections offset 10 m from the reference (e.g., Figure 2.1 inset), were used to compare the UAS data to the "true ground" TLS data. Within a 0.25-m buffer along the cross sections, TLS and UAS point cloud data were extracted for further analysis. The bottom of the bank (edge of water) to the "top of bank," determined via visual inspection of the data, defined the streambank face. The TLS- and UAS-derived bank profiles were generated using a horizontal and vertical reference plane and using 0.05-m-spaced intervals for TLS and 0.20-m-spaced intervals for UAS, respectively. The point with the minimum elevation in each horizontal interval (the 2-D equivalent of a simple minimum-z or 2.5 D filter used in bare-earth DEM creation; Figure 2.3a), and the maximal distance in each vertical interval (Figure 2.3c) formed the basis of the bank profile.

At each cross section, the differences (at 0.20 m increments) between the TLS and UAS data in the vertical direction were computed by subtracting TLS profile from the UAS profile, and for the horizontal direction, the UAS profile was subtracted from the TLS profile. Therefore, a positive value in either the vertical or horizontal direction implies the same direction of error (i.e., UAS overestimates elevation of ground surface). The differences in the mean, median, standard deviation, and RMSE were computed for each cross section. Differences were recomputed between survey dates to quantify change over time. At each cross section, the amounts of streambank erosion and deposition were calculated by computing the change in cross-sectional area for both UAS and TLS survey data.

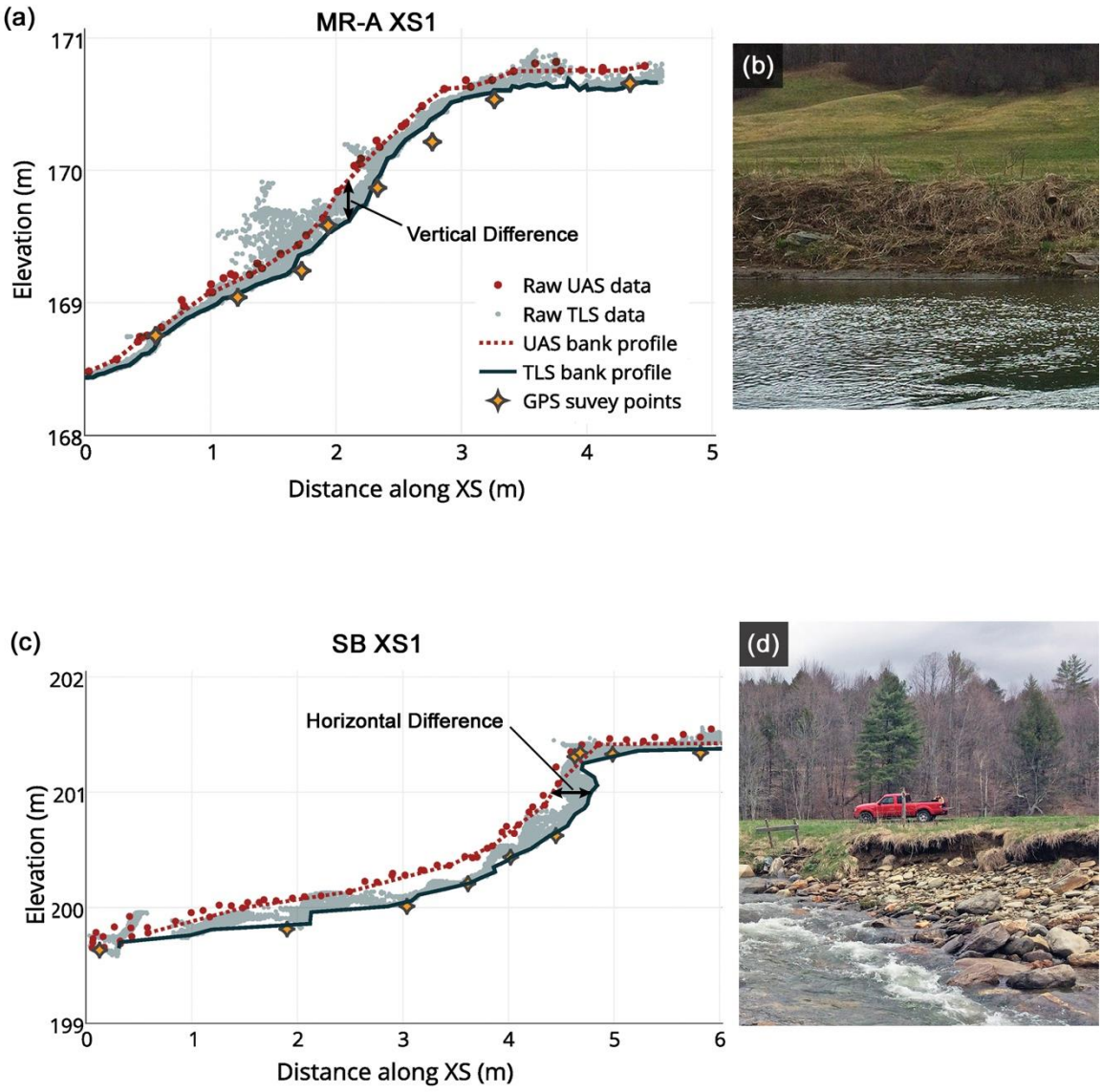


Figure 2.3 Cross sections showing raw point cloud data from unmanned aircraft system (UAS) and terrestrial laser scanner (TLS) surveys with realtime kinematic GPS points as well as derived streambank profiles. (a) XS1 at Mad River (MR)-A site on May 4, 2016; (b) photo of streambank at XS-1 location at MR-A; (c) XS1 at Shepard Brook (SB) site on May 4, 2016; and (d) photo of streambank at XS-1 location at SB

Results

UAS and TLS Comparison to GPS Survey

Streambank profiles were evaluated using UAS and TLS data from three cross sections at each of the seven sites. Examples of the raw data collected along the reference cross sections (XS1) for two sites and photographs as they appeared when surveyed are provided in Figure 2.3. Figures 2.3a and 2.3c highlight the spread in the UAS (larger dots) and TLS (smaller dots) point cloud data, especially areas where significant vegetation was present along the bank.

Table 2.1 shows a summary of the error metrics between surveyed RTK-GPS points and the TLS- and UAS-derived bank profiles. The median error between surveyed points and TLS-derived bank profile ranged from 0.009 to 0.119 m, and median errors ranged from <0.001 to 0.103 m. The overall median error across all surveys was 0.047 m. Because the UAS manufacturer specifies a maximum vertical accuracy of 0.05 m, using TLS data to generate ground surface profiles along streambanks seems justifiable. TLS measurements are capable of higher accuracy, but the presence of vegetation and soft soils suggest that attaining more precise (< 1 cm) surveying in the field will be unlikely.

The comparison between RTK-GPS survey and UAS-derived bank profiles shows that error metrics varied more across the seven streambank sites than the TLS data. Mean errors ranged from 0.011 to 0.636 m with median errors between 0.006 and 0.580 m; the overall RMSE across all sites for UAS was 0.289 m, compared to 0.089 m for TLS data.

Table 2.1 Error in TLS- and UAS-derived bank profiles compared to RTK-GPS survey points along bank cross sections

Site	Error in TLS-derived bank profiles compared to RTK-GPS				Error in UAS-derived bank profiles compared to RTK-GPS			
	Mean (m)	Median (m)	Standard deviation (m)	RMSE (m)	Mean (m)	Median (m)	Standard deviation (m)	RMSE (m)
Autumn 2015								
MR-A	0.119	0.103	0.097	0.151	0.636	0.580	0.297	0.698
MR-B	0.059	0.033	0.080	0.093	0.422	0.124	0.532	0.636
MR-C	0.104	0.088	0.120	0.154	0.249	0.160	0.308	0.382
MR-D	0.009	< 0.001	0.020	0.022	0.011	0.006	0.031	0.033
SB	0.042	0.035	0.047	0.061	0.072	0.075	0.071	0.099
NHR	0.072	0.067	0.040	0.081	0.164	0.114	0.128	0.203
Spring 2016								
MR-A	0.045	0.049	0.038	0.058	0.131	0.114	0.104	0.164
SB	0.058	0.018	0.079	0.095	0.143	0.135	0.112	0.180
NHR	0.055	0.027	0.069	0.086	0.130	0.108	0.163	0.205
All surveys	0.062	0.047	0.065	0.089	0.218	0.157	0.194	0.289

Note. MR = Mad River; NHR = New Haven River; RMSE = root mean square error; RTK = real-time kinematic; SB = Shepard Brook; TLS = terrestrial laser scanner; UAS = unmanned aircraft system.

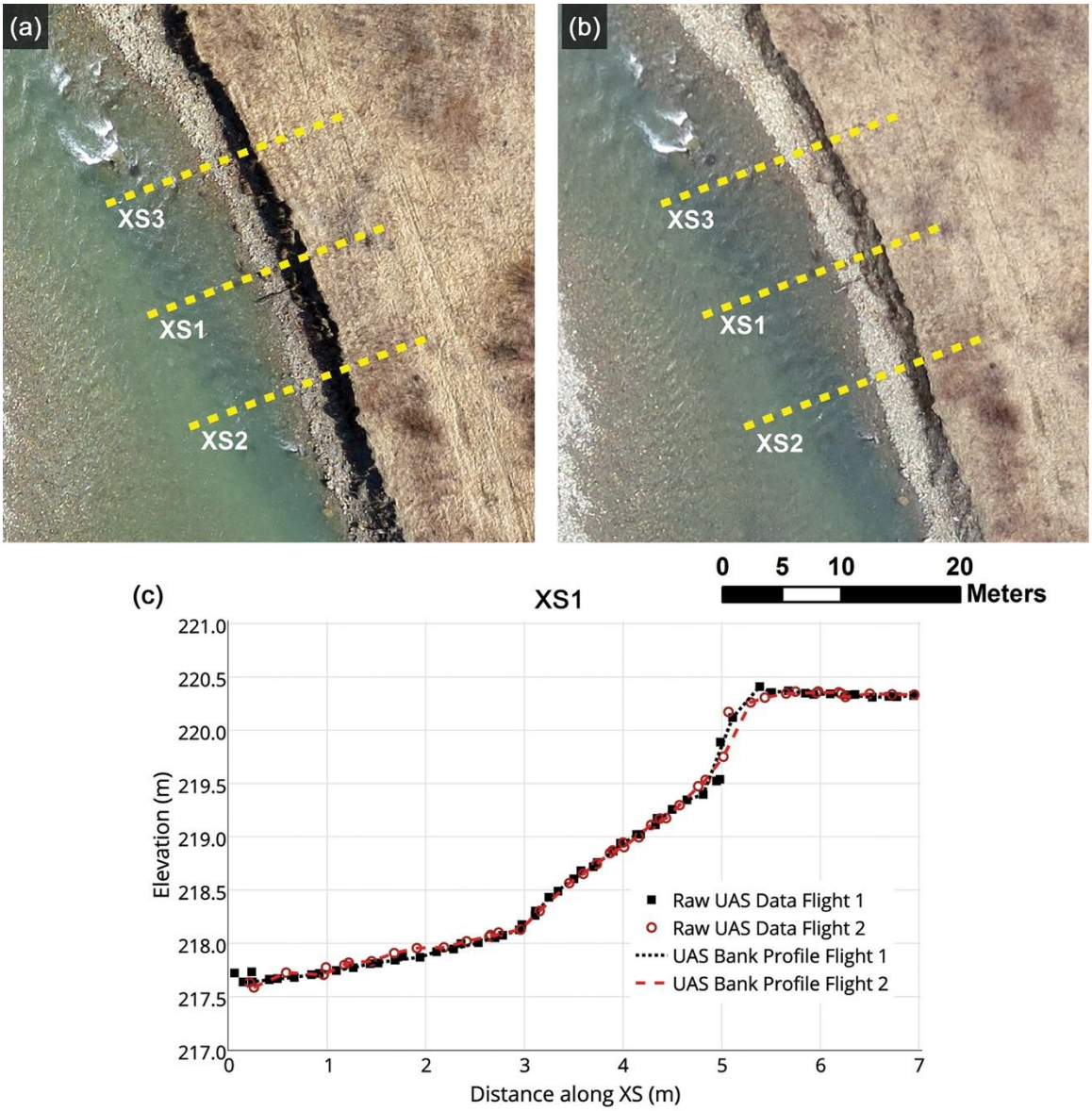


Figure 2.4 Comparison of data from two unmanned aircraft system (UAS) flights at streambank site at Mad River D site from April 22, 2015, flown within an hour of each other: (a) Flight 1 and (b) Flight 2 orthoimagery from UAS showing different shadowing along bank; and (c) cross-section XS1 showing raw point cloud data and derived bank profiles from the two flights

Repeatability of UAS Data

Streambank site MR-D was flown twice on April 22, 2015, using the same UAS flight parameters. Although the flights were completed back to back, the varying cloud cover and lighting conditions altered the location of the shadows (Figure 2.4a,b). The data from these

two flights allowed us to evaluate (a) the repeatability of the UAS and (b) the effects of shadows along streambanks. Figure 2.4c shows a streambank cross section (XS-1) at the MR-D site and the raw point cloud data from both flights; the data from both flights agree well and show little effect of shadows. When the error is averaged across three cross sections (XS-1 through XS-3), the absolute median vertical differences between the two UAS flights were 0.03 and 0.071 m for the horizontal differencing approach. The RMSE was 0.09 and 0.25 m for vertical and horizontal differences, respectively.

Comparison of UAS to TLS

For a comprehensive analysis of using UAS data across sites with varying vegetation and at different times of the year, the UAS-derived streambank profiles were compared to the TLS-derived ground surfaces. With three cross sections from each site and multiple UAS and TLS survey dates, 56 paired UAS and TLS data sets were available for comparison. Two examples are shown in Figure 2.3a,c using both a horizontal and vertical reference plane. Table 2.2 summarizes the error metrics for the MR-A site; this site had very dense vegetation along the streambank. Values of mean and median errors were nearly all positive indicating that UAS-derived elevations were more biased than the TLS surveyed ground surface. Errors at the MR-A site were smaller for the May 4, 2016, survey date with an overall RMSE of 0.19 m across all three cross sections for vertical differences and 0.26 m for horizontal differences. These errors were significantly lower than the November 9, 2015, survey date, where RMSE calculated using vertical and horizontal differences were 0.84 and 1.40 m, respectively. In all cases, RMSE was greater when using horizontal differences than vertical differences. For spring and autumn survey dates, vegetation on the bank was bare of leaves; however, dead

Table 2.2 Differences between bank profiles derived from UAS and TLS data computed in horizontal and vertical directions for the MR-A streambank site

Site: MR-A	May 6, 2015				November 9, 2015				May 4, 2016			
	XS1	XS2	XS3	All XS	XS1	XS2	XS3	All XS	XS1	XS2	XS3	All XS
Vertical difference												
Mean (m)	0.47	0.19	0.48	0.38	0.75	0.77	0.47	0.67	0.14	0.00	0.34	0.16
Median (m)	0.32	0.14	0.44	0.30	0.64	0.75	0.25	0.55	0.13	−0.01	0.34	0.15
Standard deviation (m)	0.33	0.17	0.21	0.24	0.42	0.62	0.52	0.52	0.07	0.04	0.13	0.08
RMSE (m)	0.57	0.25	0.52	0.45	0.85	0.98	0.69	0.84	0.16	0.04	0.36	0.19
Horizontal difference												
Mean (m)	0.82	0.43	0.65	0.63	1.36	1.94	0.71	1.33	0.19	0.02	0.38	0.20
Median (m)	0.90	0.37	0.59	0.62	1.55	2.01	0.76	1.44	0.19	0.02	0.32	0.18
Standard deviation (m)	0.28	0.31	0.18	0.25	0.44	0.56	0.33	0.44	0.05	0.09	0.33	0.15
RMSE (m)	0.86	0.52	0.67	0.68	1.42	2.00	0.77	1.40	0.19	0.08	0.49	0.26

Note. MR = Mad River; RMSE = root mean square error; TLS = terrestrial laser scanner; UAS = unmanned aircraft system.

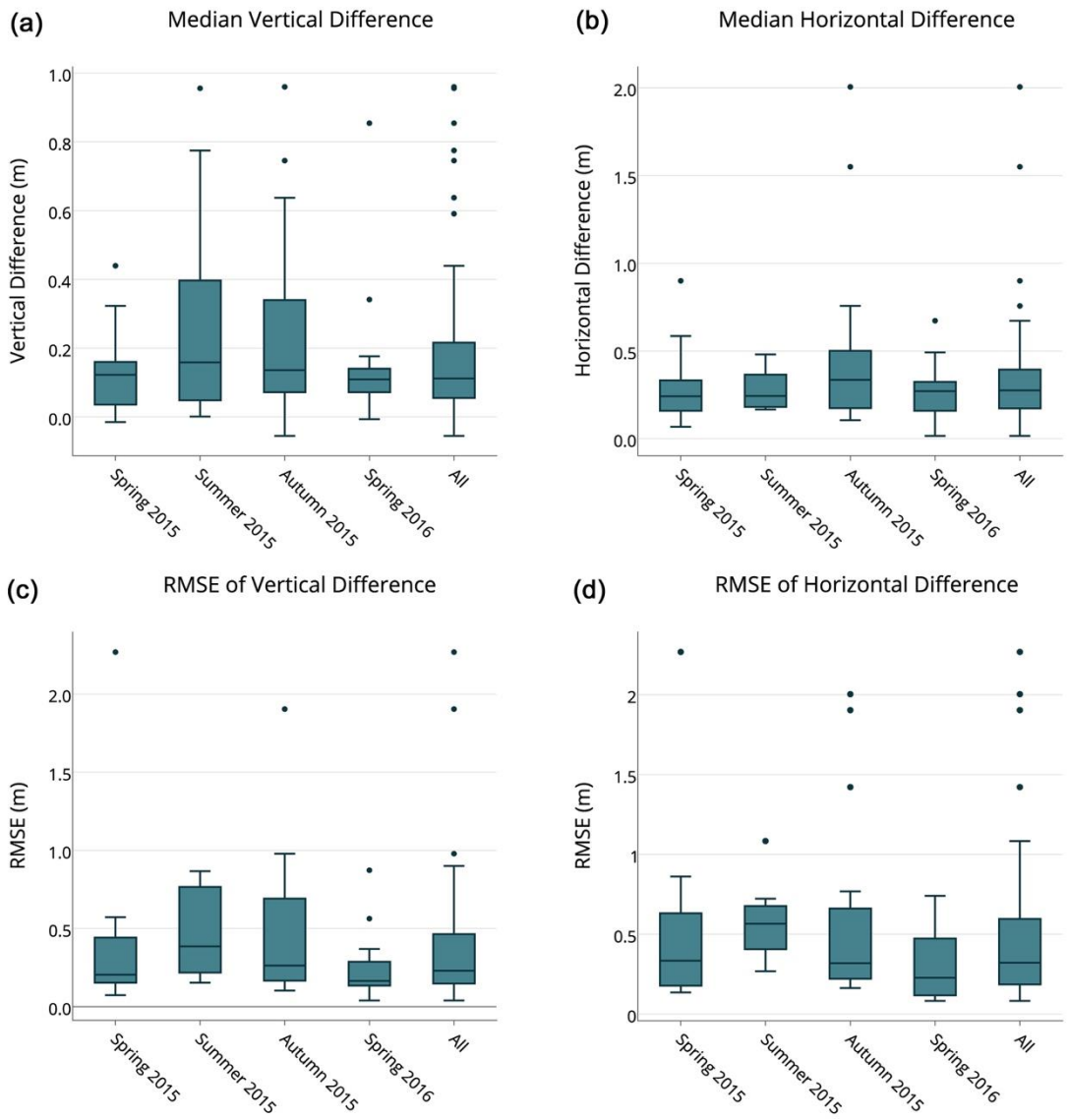


Figure 2.5 Box plots of (a) median vertical and (b) median horizontal differences between unmanned aircraft system and terrestrial laser scanner bank profiles across all sites and cross sections; (c) root mean square error (RMSE) of vertical differences; and (d) RMSE of horizontal differences. Median values are represented by the horizontal line inside the box with box limits representing the upper and lower quartile of the data. Whiskers extend to the minimum and maximum of the data, excluding outliers (points greater than 1.5 times the upper quartile and less than 1.5 times the lower quartile)

standing brush was denser in autumn compared to spring when this vegetation was generally more matted down.

Observed trends in UAS-derived bank profile error metrics at the MR-A site were similar across all sites. In general, the lowest errors occurred during spring conditions compared to summer and autumn. Figure 2.5 summarizes the median and RMS error metrics calculated for both horizontal and vertical differences across the different seasonal survey periods. Overall median errors of 0.11 and 0.28 m, and RMSE of 0.32 and 0.39 m for vertical and horizontal differences, respectively, were obtained across all sites during spring survey dates. Greater errors (i.e., median vertical errors of 0.16 and 0.14 m, an increase of 45% to 28%, respectively) were observed during the summer and autumn surveys. Median errors were positive for each survey date at all sites, indicating UAS yielded higher elevation values compared to the TLS surveyed bank surfaces.

Measurement of streambank movement

A snowmelt event on February 26, 2016, caused significant streambank erosion at two locations along the NHR site (Figure 2.6a); cantilever bank failures were observed along the channel resulting in bank retreats up to ~13 m. This event provided an opportunity for direct comparison of bank erosion between the UAS and TLS data at multiple cross sections (Figure 2.6a). Table 2.3 shows the mean, median, and standard deviation of bank retreat measurements (lateral change made along the bank profile) for each of the cross sections using both the UAS and TLS data.

Bank erosion in Area 1 (Figure 2.6a) was significant enough to cause median bank-retreat between 0.03 and 0.92 m as computed from TLS surveys. In Area 2, median bank retreats measured between 9.67 and 12.71 m. Negligible erosion was observed at the reference cross section (XS1) and at XS2 and XS3 where median bank retreats all measured less than

0.05 m, which is equivalent to the overall mean error observed in TLS measurements (Table 2.3).

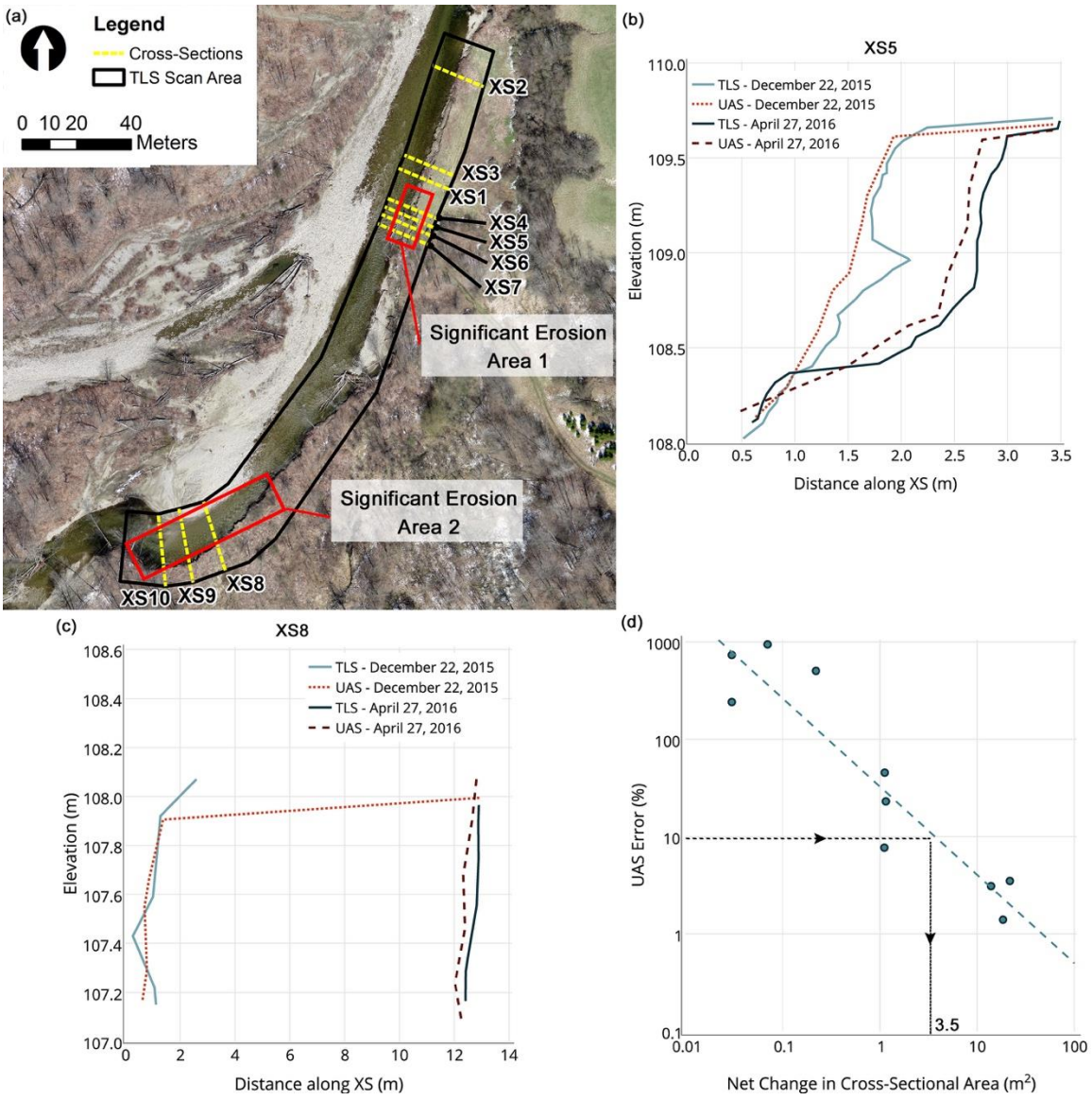


Figure 2.6 (a) New Haven River (NHR) site with cross sections and area of terrestrial laser scanner (TLS) scan acquisition. Cross sections from Areas 1 and 2 at NHR site showing bank profiles as measured using TLS and unmanned aircraft system (UAS) on December 22, 2015, and April 27, 2016; (b) UAS and TLS bank profiles at XS2

Cross-sectional erosion and deposition were computed at the NHR site to quantify the net change and error between UAS and TLS measurements (Figure 2.6b,c). The cross-sectional area of erosion as measured by TLS ranged from 0.08 to 21.71 m^2 across the 10 cross sections,

and the depositional area varied from 0 up to 0.59 m^2. The net change at cross sections with negligible movement (XS1, XS2, and XS3) ranged from −0.03 to −0.22 m^2, where a negative area represents net erosion. At Area 1, where moderate erosion occurred, net changes ranged from −0.03 to −1.63 m^2, and at Area 2, the net change varied from −13.89 to −21.71 m^2 across the three cross sections. The UAS-based measurements were compared to the TLS measurements, and the error in net change ranged from 0.08 to 1.11 m^2 with a mean of 0.44 m^2 across the 10 cross sections. Cross-sectional area and the corresponding net change are included in Table 2.3 along with UAS measurement error and percent error of UAS data compared to TLS data.

Vegetation conditions on December 22, 2015, and April 27, 2016, were very similar at the NHR site allowing for comparison of UAS and TLS to detect change over time. Figure 2.6d shows percent error in the net change of UAS measurements as a function of measured TLS net change. Percent error decreased by an order of magnitude corresponding to an order of magnitude increase in the observed change of area. A simple linear regression model estimated the percent error given a measured net change in cross-sectional area. For a target of 10% error, this corresponds to a 3.5 m^2 change in area (Figure 2.6d).

Table 2.3 Comparison between UAS and TLS measured horizontal bank retreat along profile of cross section and cross sectional area of bank erosion and deposition

Cross section	Mean bank retreat (m)		Median bank retreat (m)		Std. dev. bank retreat (m)		Erosion (m^2)		Deposition (m^2)		Net change (m^2)		UAS error (m^2)	Percent error (%)
	TLS	UAS	TLS	UAS	TLS	UAS	TLS	UAS	TLS	UAS	TLS	UAS		
XS1	−0.01	−0.21	0.00	−0.17	0.07	0.22	0.08	0.47	0.05	0.19	−0.03	−0.28	0.25	735.0
XS5	−0.77	−0.75	−0.92	−0.90	0.39	0.37	1.18	1.24	0.07	0.05	−1.11	−1.19	0.09	7.7
XS2	−0.05	−0.25	0.04	−0.02	0.30	0.68	0.19	0.81	0.13	0.12	−0.07	−0.69	0.62	943.6
XS3	0.04	−0.92	−0.04	−0.52	0.27	0.71	0.36	1.34	0.14	0.01	−0.22	−1.33	1.11	505.5
XS4	0.01	−0.05	−0.03	0.15	0.75	0.52	0.62	0.46	0.59	0.35	−0.03	−0.12	0.08	241.6
XS6	−0.53	−0.71	−0.24	−0.51	0.64	0.70	1.18	1.64	0.06	0.01	−1.12	−1.63	0.51	45.4
XS7	−0.63	−0.41	−0.78	−0.15	0.56	0.48	1.17	0.94	0.02	0.05	−1.15	−0.89	0.27	23.0
XS8	−12.54	−12.15	−12.71	−14.42	0.40	5.01	21.71	21.29	0.00	0.35	−21.71	−20.94	0.77	3.5
XS9	−9.60	−6.12	−9.67	−9.30	0.19	3.92	13.89	14.72	0.00	0.40	−13.89	−14.32	0.43	3.1
XS10	−11.68	−11.43	−11.82	−11.48	0.36	0.26	18.45	18.19	0.00	0.01	−18.45	−18.18	0.26	1.4

Note. TLS = terrestrial laser scanner; UAS = unmanned aircraft system.

Effect of vegetation conditions on accuracy

The UAS-derived cross sections have greater errors in locations with dense vegetation. We used the two streambank sites with negligible erosion over the study period to quantify the effects of vegetation on the repeatability and accuracy of the UAS-derived topography. Figure 2.7 shows a streambank cross section (at MR-D) that was nearly completely bare of vegetation (Figure 2.7a), and a bank (at MR-A) that features dense vegetation (Figure 2.7b). The UAS data from each survey date were compared with all other survey dates using both horizontal and vertical differences; the median and RMSE values are reported. The median vertical difference between all four survey dates averaged across three cross sections at the vegetation-free MR-D bank was <0.01 m with a corresponding RMSE of 0.17 m. At the vegetated MR-A bank, the differences were nearly 3 times greater with a median of 0.10 m and RMSE of 0.46 m. Horizontal differences showed similar trends between the two sites.

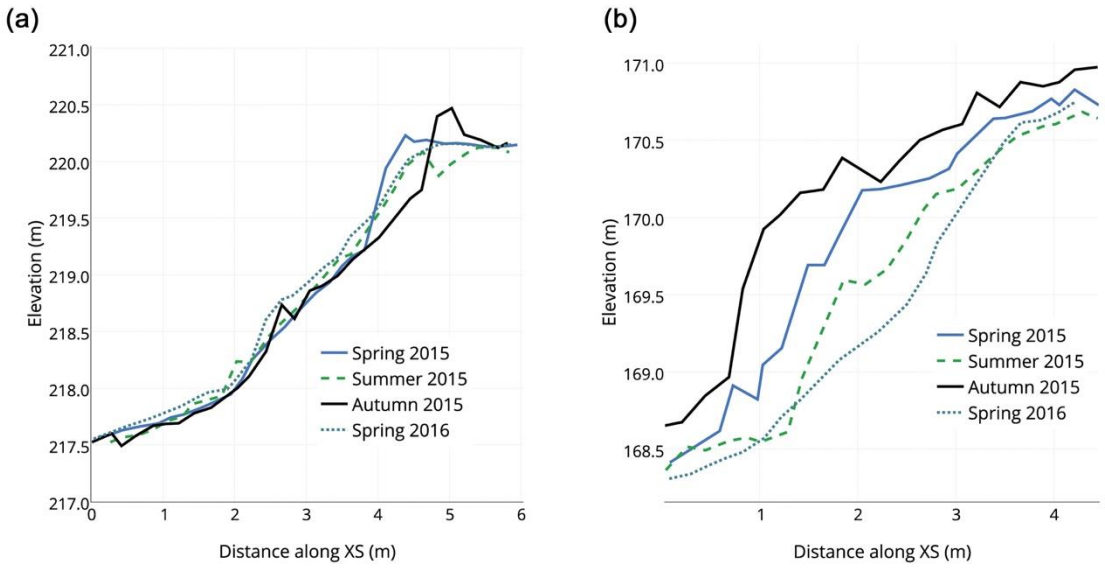


Figure 2.7 Comparison of unmanned aircraft system-derived bank profiles at two stable streambank sites where negligible erosion was observed during the study period; (a) cross section XS1 at Mad River (MR)-D site and (b) cross section XS1 at MR-A site

Discussion

Accuracy of UAS-derived topography along streambanks

In this study, the topographic data obtained from UAS were compared to TLS and GPS survey data at seven streambank sites representing a variety of river geomorphologies. Results demonstrate that the UAS is capable of capturing high-quality topographic data even in challenging landscapes, such as river corridors, provided water levels are low and vegetation is minimal. The UAS accuracy (0.14 m) and precision (0.18 m) achieved, whereas not representative of the best possible under ideal settings, are similar to existing findings by James and Robson (2012), Flener et al. (2013), Woodget et al. (2015), and Carbonneau and Dietrich (2017) in vegetative areas and along river corridors. These earlier studies focused at the stream reach scale, whereas study presented here achieved similar accuracies at a much broader geographic scale. In addition, UAS results are repeatable in areas with minimal vegetation, with overall median differences of less than 1 cm across four sample dates of the exposed bank MR-D site. Nearly all of the median RMS errors had positive values indicating a UAS bias in yielding higher elevation values. Because UAS data are generated from imagery that includes all surface vegetation, this result is expected, but it does indicate that overestimation of erosion is possible in vegetated areas.

Georeferencing the data is one of the most critical aspects for obtaining reliable results. This is especially true if data are generated using multiple survey methods such as GPS, TLS, and UAS. The use of GCPs, typically surveyed using an RTK-GPS system, is an effective method for georeferencing data. We assessed the ability of using georeferenced UAS data to capture streambank topography and detect change. Other UAS-based photogrammetry

studies (Carbonneau & Dietrich, 2017; Hugenholtz et al., 2016; Shahbazi, Sohn, Théau, & Menard, 2015) have presented detailed analysis of georeferencing methods and practices. Hugenholtz et al. (2016) assessed the accuracy of the same eBee and eBee RTK UASs used in this study with and without GCPs. They found that the eBee RTK achieved a horizontal accuracy <0.025 m without using GCPs and also obtained vertical accuracy to <0.1 m. Carbonneau and Dietrich (2017) found that using a sparse (3 to 4) set of GCPs was sufficient to achieve similar accuracies provided a GPS-equipped UAS is used that allows direct georeferencing of the photogrammetric data. Our check on the georeferencing (i.e., horizontal accuracy of <0.01 m and vertical accuracy of <0.05 m) showed a similar level of accuracy. Therefore, larger differences between UAS and TLS data indicate causes other than georeferencing, such as effects of vegetation, which are discussed in further detail in subsequent sections.

Measurements of bank retreat and elevation change

Selecting a reference plane orientation is inherent when calculating changes in elevation between surveys collected at different dates. A comparison of the UAS and TLS data across the seven sites shows the effects of a reference plane orientation (i.e., vertical or horizontal). The overall mean and median differences between UAS- and TLS-derived bank profiles derived from a horizontal reference plane were more than double those measured vertically. Many traditional streambank erosion methods (e.g., bank erosion pins) enable bank retreat to be measured horizontally (Lawler, 1993). Additionally, detailed studies of streambank topography using survey instruments such as TLS have utilized a vertical reference plane to compute horizontal retreat along the bank (O'Neal & Pizzuto, 2011; Resop & Hession, 2010). Therefore, when calculating horizontal bank retreats utilizing UAS-derived

topography, care should be taken when computing UAS data error metrics (i.e., typical RMSE) using vertical differences, because they are not directly transferrable.

Effects of vegetation on accuracy

Topographic data derived from photogrammetric methods using UAS imagery are inherently limited by the need for line of sight from the UAS to the ground. One of the primary limitations in capturing topographic data along river corridors is having the ground obscured by vegetation. In their study, Woodget et al. (2015) found that UAS was capable of producing precise and accurate DEMs in nonvegetated and exposed areas. Unfortunately, vegetation is often present along streambanks and river corridors, and as such, one of our research goals was to identify the impact of vegetation on accuracy. The highest UAS accuracy (overall median error of 0.11 m) was achieved during spring conditions, prior to the onset of vegetation. During summer and autumn conditions, the overall median errors worsen to 0.14 and 0.16 m, respectively, resulting in an increased error of 28% to 45%. We expected the effect of vegetation in autumn to be similar to that in spring because both are leaf-off conditions. One explanation for the observed discrepancy may be that dead vegetation becomes more matted over the winter, resulting in more exposed ground surface compared to the autumn. We observed the latter at multiple cross sections including the one in Figure 2.7b, where the autumn UAS-derived topographic data captured the dead standing vegetation.

UAS-based photogrammetry is not currently applicable for capturing the ground surface in locations completely obscured by vegetation as was observed in this study. However, it did prove surprisingly capable for capturing ground surface along vegetated streambanks under the right conditions. At the MR-B and MR-C sites, for example, the bank is lined with large trees and light brush during early spring, and the UAS surveys were very effective at

capturing the ground surface (median vertical errors of 0.05 and 0.11 m, respectively). Cross sections with the poorest accuracies featured dense, low brush where median vertical errors approached 1 m. However, as noted, these UAS surveys should be conducted when ground surfaces are mostly exposed, such as during early spring for regions similar to Vermont.

UAS for measuring streambank erosion

Results showed the UAS reliably estimates large amounts of bank movement within 10% of the change captured by TLS surveys along a typical streambank (Figure 2.6d). The threshold of horizontal retreat on streambanks with heights that typically allow this margin of error (e.g., 2 m) was found to be about 1.8 m, assuming a slab failure. Quantifying erosion or deposition in areas with smaller rates of retreat is more sensitive to the effects of vegetation and other sources of error. Estimating small amounts of cross-sectional erosion (i.e., totalling less than 0.25 m^2) was highly inaccurate, with percent errors approaching 1,000%; this is likely the result of multiple factors including vegetation and residual errors in georeferencing being of similar magnitude to the net change in area. With vertical or nearly vertical banks, shadows are commonly present in the imagery, which has been found to affect the accuracy of the UAS topographic data in other studies (Shahbazi et al., 2015). In this study, however, the effects of shadows were negligible based on comparison of two flights done on the same day under different daylight conditions (Figure 2.4). A median vertical difference of only 3.4 cm was observed, which is approximately equal to ground sample resolution of the UAS data. Vertical banks and those with undercuts were common at several of our study sites including the NHR site where significant erosion occurred. Undercutting in particular creates a challenge for estimating bank erosion areas and bank retreat rates using fixed-wing UAS, the data from which are typically generated from orthomosaics from airborne imagery collected at nadir, and

undercut surfaces are not visible and present in the data. Advances in multirotor UAS flight capabilities that can capture oblique imagery offer promise to survey areas with extensive undercut banks.

Conclusions and Future Work

There is considerable interest in monitoring the changing geomorphology of river corridors using UAS-based photogrammetry at site specific to river-network scale. In this work, the efficacy of using UAS-based photogrammetry for capturing the topography of the streambank surfaces at fine detail and from long sight distances was assessed and compared to TLS and GPS surveying methods at seven sites encompassing a variety of bank conditions.

The results show that UAS reliably captures the bank surface and compares well to TLS and GPS data. However, the accuracy depends on the density of the vegetation along the bank. The highest accuracies were observed during leaf-off conditions in early spring, when dead vegetation is matted down along the banks, and new growth has not yet begun compared to the higher errors observed during autumn leaf-off conditions. Under the optimal spring conditions, median vertical errors of 0.11 m were observed in UAS data. Vegetation during summer introduced the greatest errors. When surveys were performed over large spatial monitoring campaigns (i.e., 10s of kilometres of river length) and when vegetation was minimal, UAS-based photogrammetry reliably captured the bank surface topography within 0.2m. This matches or improves upon the data resolution currently available from airborne lidar surveys, yet it provides greater flexibility in the lead time and ability to capture more focused areas. For detailed study of small bank sections, TLS still provides the highest resolution and accuracy, albeit at approximately an order of magnitude greater cost for equipment and labour.

The largest challenge in using UAS-derived topography for monitoring streambank erosion is the effect of vegetation on data accuracy. Dense vegetation introduces large data errors and, in some cases, can completely obscure the ground surface. The usefulness of UAS for streambank monitoring application is, therefore, limited in areas where vegetation is present year round (i.e., subtropical and tropical climates).

Currently, UAS data acquisition and processing are reasonably efficient over small areas (<1 km of river length); however, automating the process over larger areas could be a challenge. Additional research into efficient georeferencing of data over large areas is justified. The more advanced UAS platforms featuring built-in GPS, and in particular, the more advanced GNSS/RTK on-board receivers show great potential for eliminating or at the least minimizing the number of GCPs required and therefore simplifying field operations. Further research into scaling up the application of UAS-based topography for monitoring bank erosion at the river network level is also needed, including automating the delineation of streambank areas using DEMs to monitor bank erosion, and identifying changes detected as due to vegetation effect or channel movement. In its current state, the UAS technology shows promise in capturing bank erosion and deposition in areas where banks are not obscured completely by vegetation.

Acknowledgements

This research was supported by the Vermont Water Resources and Lakes Studies Center. Additional support provided by Vermont EPSCoR with funds from the National Science Foundation (NSF) Grant EPS-1101317 and EPS-1556770, NSF Grant CMMI-1229045, NSF Graduate Research Fellowship under Grant DGE-0925179NSF, the Robert & Patricia Switzer Foundation, and Grant OASRTRS-14-HUVM from the US Department of

Transportation is also gratefully acknowledged. The views, opinions, findings, and conclusions reflected in this paper are solely those of the authors and do not represent the official policy or position of any funding sources or endorse any third-party products or services that may be included in this presentation or associated materials. The authors acknowledge the additional contributions of the UVM Spatial Analysis Lab UAS team, Hannah Anderson, Tayler Engel, Kristen Underwood, and Anna Waldron.

References

Bangen, S. G., Wheaton, J. M., Bouwes, N., Bouwes, B., & Jordan, C. (2014). A methodological intercomparison of topographic survey techniques for characterizing wadeable streams and rivers. *Geomorphology*, 206, 343–361. https://doi.org/10.1016/j.geomorph.2013.10.010

Borg, J., Dewoolkar, M. M., & Bierman, P. (2014). Assessment of streambank stability-a case study. Presented at the Geo-Congress 2014. Atlanta, Georgia. 1007–1016 pp. [online] Available from: http://ascelibrary.org/doi/abs/10.1061/9780784413272.098. Accessed 23 June 2016

Brasington, J., Vericat, D., & Rychkov, I. (2012). Modeling river bed morphology, roughness, and surface sedimentology using high resolution terrestrial laser scanning. *Water Resources Research*, 48. W11519. DOI: https://doi.org/10.1029/2012WR012223

Brodu, N., & Lague, D. (2012). 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. *ISPRS Journal of Photogrammetry and Remote Sensing,* 68, 121–134. https://doi.org/10.1016/j.isprsjprs.2012.01.006

Buchanan, B. P., Walter, M. T., Nagle, G. N., & Schneider, R. L. (2012). Monitoring and assessment of a river restoration project in central New York. *River Research and Applications*, 28, 216–233. https://doi.org/10.1002/rra.1453

Carbonneau, P. E., & Dietrich, J. T. (2017). Cost-effective non-metric photogrammetry from consumer-grade sUAS: Implications for direct georeferencing of structure from motion photogrammetry. *Earth Surface Processes and Landforms*, 42, 473–486. https://doi.org/10.1002/esp.4012

Chen, D., & Duan, J. G. (2006). Modeling Channel morphologic change in the West Jordan River, Utah. Presented at the World Environmental and Water Resource Congress 2006. Omaha, Nebraska. 1–10 pp. [online] Available from: http://ascelibrary.org/doi/abs/10.1061/40856%28200%2963. Accessed 2 August 2016

Collins, A. L., & Walling, D. E. (2004). Documenting catchment suspended sediment sources: Problems, approaches and prospects. *Progress in Physical Geography*, 28, 159–196. https://doi.org/10.1191/0309133304pp409ra

Colomina, I., & Molina, P. (2014). Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92, 79–97. https://doi.org/10.1016/j.isprsjprs.2014.02.013

Cook, A., & Merwade, V. (2009). Effect of topographic data, geometric configuration and modeling approach on flood inundation mapping. *Journal of Hydrology*, 377, 131–142. https://doi.org/10.1016/j.jhydrol.2009.08.015

Daly, E. R., Miller, R. B., & Fox, G. A. (2015). Modeling streambank erosion and failure along protected and unprotected composite streambanks. *Advances in Water Resources*, 81, 114–127. https://doi.org/10.1016/j.advwatres.2015.01.004

Dapporto, S., Rinaldi, M., Casagli, N., & Vannocci, P. (2003). Mechanisms of riverbank failure along the Arno River, central Italy. *Earth Surface Processes and Landforms*, 28, 1303–1323. https://doi.org/10.1002/esp.550

Darby, S. E., & Thorne, C. R. (1996). Development and testing of riverbank stability analysis. *Journal of Hydraulic Engineering*, 122, 443–454. https://doi.org/10.1061/(ASCE)0733-9429(1996)122:8(443)

De Rose, R. C., & Basher, L. R. (2011). Measurement of river bank and cliff erosion from sequential LIDAR and historical aerial photography. *Geomorphology*, 126, 132–147. https://doi.org/10.1016/j.geomorph.2010.10.037

Eaton, B. C. (2006). Bank stability analysis for regime models of vegetated gravel bed rivers. *Earth Surface Processes and Landforms*, 31, 1438–1444. https://doi.org/10.1002/esp.1364

Evans, D. J., Gibson, C. E., & Rossell, R. S. (2006). Sediment loads and sources in heavily modified Irish catchments: A move towards informed management strategies. *Geomorphology*, 79, 93–113. https://doi.org/10.1016/j.geomorph.2005.09.018

Fitzgerald, E. P., & Godfrey, L. C. (2008). *Upper Mad River corridor plan*. Waitsfield, VT: Friends of the Mad River.

Flener, C., Vaaja, M., Jaakkola, A., Krooks, A., Kaartinen, H., Kukko, A., … Alho, P. (2013). Seamless mapping of river channels at high resolution using mobile LiDAR and UAV-photography. *Remote Sensing*, 5, 6382–6407. https://doi.org/10.3390/rs5126382

Fox, G. A., Wilson, G. V., Simon, A., Langendoen, E. J., Akay, O., & Fuchs, J. W. (2007). Measuring streambank erosion due to ground water seepage: Correlation to bank pore water pressure, precipitation and stream stage. *Earth Surface Processes and Landforms*, 32, 1558–1573. https://doi.org/10.1002/esp.1490

Garvey, K. M. (2012). Quantifying erosion and deposition due to stream planform change using high spatial resolution digital orthophotography and lidar data, M.S. Thesis, University of Vermont: Burlington, VT.

Heritage, G. L., & Milan, D. J. (2009). Terrestrial laser scanning of grain roughness in a gravel-bed river. *Geomorphology*, 113, 4–11. https:// doi.org/10.1016/j.geomorph.2009.03.021

Hodge, R., Brasington, J., & Richards, K. (2009). Analysing laser-scanned digital terrain models of gravel bed surfaces: Linking morphology to sediment transport processes and hydraulics. *Sedimentology*, 56, 2024–2043. https://doi.org/10.1111/j.1365-3091.2009.01068.x

Hohenthal, J., Alho, P., Hyyppa, J., & Hyyppa, H. (2011). Laser scanning applications in fluvial studies. *Progress in Physical Geography*, 35, 782–809. https://doi.org/10.1177/0309133311414605

Hugenholtz, C., Brown, O., Walker, J., Barchyn, T., Nesbit, P., Kucharczyk, M., & Myshak, S. (2016). Spatial accuracy of UAV-derived Orthoimagery and topography: Comparing photogrammetric models processed with direct geo-referencing and ground control points. *GEOMATICA*, 70, 21–30. https://doi.org/10.5623/cig2016-102

Hugenholtz, C. H., Whitehead, K., Brown, O. W., Barchyn, T. E., Moorman, B. J., LeClair, A., … Hamilton, T. (2013). Geomorphological mapping with a small unmanned aircraft system (sUAS): Feature detection and accuracy assessment of a photogrammetrically-derived digital terrain model. *Geomorphology*, 194, 16–24. https://doi.org/10.1016/j.geomorph.2013.03.023

Hughes, M. L., McDowell, P. F., & Marcus, W. A. (2006). Accuracy assessment of georectified aerial photographs: Implications for measuring lateral channel movement in a GIS. *Geomorphology*, 74, 1–16. https://doi.org/10.1016/j.geomorph.2005.07.001

Jaboyedoff, M., Demers, D., Locat, J., Locat, A., Locat, P., Oppikofer, T., … Turmel, D. (2009). Use of terrestrial laser scanning for the characterization of retrogressive landslides in sensitive clay and rotational landslides in river banks. *Canadian Geotechnical Journal*, 46, 1379–1390. https://doi.org/10.1139/T09-073

James, M. R., & Robson, S. (2012). Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application. *Journal of Geophysical Research: Earth Surface*, 117. F03017.DOI:https://doi.org/10.1029/2011JF002289

Javernick, L., Brasington, J., & Caruso, B. (2014). Modeling the topography of shallow braided rivers using structure-from-motion photogrammetry. *Geomorphology*, 213, 166–182. https://doi.org/10.1016/j.geomorph.2014.01.006

Lague, D., Brodu, N., & Leroux, J. (2013). Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (N-Z). *ISPRS Journal of Photogrammetry and Remote Sensing*, 82, 10–26. https://doi.org/10.1016/j.isprsjprs.2013.04.009

Lai, Y. G., Thomas, R. E., Ozeren, Y., Simon, A., Greimann, B. P., & Wu, K. (2015). Modeling of multilayer cohesive bank erosion with a coupled bank stability and mobile-bed model. *Geomorphology*, 243, 116–129. https://doi.org/10.1016/j.geomorph.2014.07.017

Langendoen, E. J., Simon, A., Klimetz, L., Natasha, B., & Ursic, M. E. (2012). *Quantifying sediment loadings from streambank erosion in selected agricultural watersheds draining to Lake Champlain*. Technical Report. Grand Isle, VT: Lake Champlain Basin Program.

Lawler, D. M. (1993). The measurement of river bank erosion and lateral channel change: A review. *Earth Surface Processes and Landforms*, 18, 777–821. https://doi.org/10.1002/esp.3290180905

Lejot, J., Delacourt, C., Piégay, H., Fournier, T., Trémélo, M.-L., & Allemand, P. (2007). Very high spatial resolution imagery for channel bathymetry and topography from an unmanned mapping controlled platform. *Earth Surface Processes and Landforms*, 32, 1705–1725. https://doi.org/10.1002/esp.1595

Micheletti, N., Chandler, J. H., & Lane, S. N. (2015). Investigating the geomorphological potential of freely available and accessible structure-from-motion photogrammetry using a smartphone. *Earth Surface Processes and Landforms*, 40, 473–486. https://doi.org/10.1002/esp.3648

Milan, D. J., Heritage, G. L., Large, A. R. G., & Entwistle, N. S. (2010). Mapping hydraulic biotopes using terrestrial laser scan data of water surface properties. *Earth Surface Processes and Landforms*, 35, 918–931. https://doi.org/10.1002/esp.1948

Milan, D. J., Heritage, G. L., & Hetherington, D. (2007). Application of a 3D laser scanner in the assessment of erosion and deposition volumes and channel change in a proglacial river. *Earth Surface Processes and Landforms*, 32, 1657–1674. https://doi.org/10.1002/esp.1592

Miřijovsky , J., & Langhammer, J. (2015). Multitemporal monitoring of the morphodynamics of a mid-mountain stream using UAS photogrammetry. *Remote Sensing*, 7, 8586–8609. https://doi.org/10.3390/rs70708586

Miřijovsky , J., Michalková, M. Š., Petyniak, O., Máčka, Z., & Trizna, M. (2015). Spatiotemporal evolution of a unique preserved meandering system in Central Europe—The Morava River near Litovel. *CATENA*, 127, 300–311. https://doi.org/10.1016/j.catena.2014.12.006

Molina, J.-L., Rodríguez-Gonzálvez, P., Molina, M. C., González-Aguilera, D., & Espejo, F. (2014). Geomatic methods at the service of water resources modelling. *Journal of Hydrology*, 509, 150–162. https://doi.org/10.1016/j.jhydrol.2013.11.034

Neugirg, F., Stark, M., Kaiser, A., Vlacilova, M., Della Seta, M., Vergari, F., … Haas, F. (2016). Erosion processes in calanchi in the Upper Orcia Valley, Southern Tuscany, Italy based on multitemporal high-resolution terrestrial LiDAR and UAV surveys. *Geomorphology*, 269, 8–22. https://doi.org/10.1016/j.geomorph.2016.06.027

O'Neal, M. A., & Pizzuto, J. E. (2011). The rates and spatial patterns of annual riverbank erosion revealed through terrestrial laser-scanner surveys of the South River, Virginia. *Earth Surface Processes and Landforms*, 36, 695–701. https://doi.org/10.1002/esp.2098

Osman, A. M., & Thorne, C. R. (1988). Riverbank stability analysis. I: Theory. *Journal of Hydraulic Engineering*, 114, 134–150. https://doi.org/10.1061/(ASCE)0733-9429(1988)114:2(134)

Perroy, R. L., Bookhagen, B., Asner, G. P., & Chadwick, O. A. (2010). Comparison of gully erosion estimates using airborne and ground-based LiDAR on Santa Cruz Island, California. *Geomorphology*, 118, 288–300. https://doi.org/10.1016/j.geomorph.2010.01.009

Reinfelds, I. (1997). Reconstruction of changes in bankfull width: A comparison of surveyed cross-sections and aerial photography. *Applied Geography*, 17, 203–213. https://doi.org/10.1016/S0143-6228(97)00005-2

Resop, J. P., & Hession, W. C. (2010). Terrestrial laser scanning for monitoring streambank retreat: Comparison with traditional surveying techniques. *Journal of Hydraulic Engineering*, 136, 794–798. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000233

Rinaldi, M., & Darby, S. E. (2007). 9 Modelling river-bank-erosion processes and mass failure mechanisms: Progress towards fully coupled simulations. In *Developments in earth surface processes*, Helmut Habersack, Hervé Piégay, & Massimo Rinaldi (Eds.) (pp. 213–239. [online] Available from: http://www.sciencedirect.com/science/article/pii/S0928202507111263) Elsevier: Amsterdam, Netherlands. Accessed 28 June 2016

Roze, A., Zufferey, J. C., Beyeler, A., & McClellan, A. (2014). *eBee RTK accuracy assessment. White paper sense fly* [online] Available from: https://www.sensefly.com/fileadmin/user_upload/sensefly/documents/eBee-RTK-Accuracy-Assessment.pdf (Accessed 20 June 2017)

Shahbazi, M., Sohn, G., Théau, J., & Menard, P. (2015). Development and evaluation of a UAV-photogrammetry system for precise 3D environmental modeling. *Sensors*, 15, 27493–27524. https://doi.org/10.3390/s151127493

Simon, A., Curini, A., Darby, S. E., & Langendoen, E. J. (2000). Bank and near-bank processes in an incised channel. *Geomorphology*, 35, 193–217. https://doi.org/10.1016/S0169-555X(00)00036-2

Simon, A., & Rinaldi, M. (2006). Disturbance, stream incision, and channel evolution: The roles of excess transport capacity and boundary materials in controlling channel response. *Geomorphology*, 79, 361–383. https://doi.org/10.1016/j.geomorph.2006.06.037

Smith, M. W., & Vericat, D. (2015). From experimental plots to experimental landscapes: Topography, erosion and deposition in sub-humid badlands from structure-from-motion photogrammetry. *Earth Surface Processes and Landforms*, 40, 1656–1671. https://doi.org/10.1002/esp.3747

Tamminga, A. D., Eaton, B. C., & Hugenholtz, C. H. (2015). UAS-based remote sensing of fluvial change following an extreme flood event. *Earth Surface Processes and Landforms*, 40, 1464–1476. https://doi.org/10.1002/esp.3728

Tamminga, A., Hugenholtz, C., Eaton, B., & Lapointe, M. (2015). Hyperspatial remote sensing of channel reach morphology and hydraulic fish habitat using an unmanned aerial Vehicle (UAV): A first assessment in the context of river research and management. *River Research and Applications*, 31, 379–391. https://doi.org/10.1002/rra.2743

Tseng, C.-M., Lin, C.-W., Stark, C. P., Liu, J.-K., Fei, L.-Y., & Hsieh, Y.-C. (2013). Application of a multi-temporal, LiDAR-derived, digital terrain model in a landslide-volume estimation: multi-temporal LiDAR DTM in landslide volume estimation. *Earth Surface Processes and Landforms*, 1587–1601. https://doi.org/10.1002/esp.3454

Underwood, K. L. (2004). *Phase 2 stream geomorphic assessment New Haven River Watershed Addison County, Vermont*. Addison County Regional Planning Commission: Middlebury, Vermont.

Watts, A. C., Ambrosia, V. G., & Hinkley, E. A. (2012). Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use. *Remote Sensing*, 4, 1671–1692. https://doi.org/10.3390/rs4061671

Wheaton, J. M., Brasington, J., Darby, S. E., & Sear, D. A. (2010). Accounting for uncertainty in DEMs from repeat topographic surveys: Improved sediment budgets. *Earth Surface Processes and Landforms*, 35, 136–156. https://doi.org/10.1002/esp.1886

Whitehead, K., & Hugenholtz, C. H. (2014). Remote sensing of the environment with small unmanned aircraft systems (UASs), part 1: A review of progress and challenges. *Journal of Unmanned Vehicle Systems*, 2, 69–85. https://doi.org/10.1139/juvs-2014-0006

Whitehead, K., Hugenholtz, C. H., Myshak, S., Brown, O., LeClair, A., Tamminga, A., … Eaton, B. (2014). Remote sensing of the environment with small unmanned aircraft systems (UASs), part 2: Scientific and commercial applications. *Journal of Unmanned Vehicle Systems*, 2, 86–102. https://doi.org/10.1139/juvs-2014-0007

Williams, R., Brasington, J., Vericat, D., Hicks, M., Labrosse, F., & Neal, M. (2011). Chapter Twenty - Monitoring Braided River Change Using Terrestrial Laser Scanning and Optical Bathymetric Mapping. In *Geomorphological mapping methods and applications* (pp. 507–532. [online] Available from: http://www.sciencedirect.com/science/article/pii/B9780444534460000203) Elsevier: Oxford, UK. Accessed 6 July 2012

Woodget, A. S., Carbonneau, P. E., Visser, F., & Maddock, I. P. (2015). Quantifying submerged fluvial topography using hyperspatial resolution UAS imagery and structure from motion photogrammetry. *Earth Surface Processes and Landforms*, 40, 47–64. https://doi.org/10.1002/esp.3613

CHAPTER 3. APPLICATION OF UNMANNED AIRCRAFT SYSTEM (UAS) FOR MONITORING BANK EROSION ALONG A RIVER CORRIDOR

Abstract

Excessive streambank erosion is a significant source of fine sediments and associated nutrients in many river systems as well as poses risk to infrastructure. Geomorphic change detection using high-resolution topographic data is a useful method for monitoring the extent of bank erosion along river corridors. Advances in unmanned aircraft system (UAS) and structure from motion (SfM) photogrammetry techniques have provided a powerful new tool for capturing high resolution topographic data. To evaluate the effectiveness of UAS-based photogrammetry for monitoring bank erosion, a fixed-wing UAS was deployed to survey 20 km of river corridors in central Vermont, in the north-eastern United States multiple times over a two-year period. Digital elevation models (DEMs) and DEMs of difference were utilized to quantify the volumetric change along selected portions of the survey area where notable erosion occurred. Longer term estimates of change were made by comparison of UAS surveys to previously collected airborne lidar surveys. Results showed that UAS was capable of collecting high quality topographic data at fine resolutions even along vegetated river corridors provided that the survey timing and conditions were optimal. UAS survey data compared well to existing airborne lidar surveys and allowed robust quantification of significant geomorphic changes along rivers.

Introduction

Monitoring geomorphic change along river corridors is typically an essential component of developing watershed and surface water management strategies. Rate of streambank erosion is one of the direct measurements of fluvial geomorphic change used to analyse river corridors. Streambank erosion can represent a large portion of overall sediment and nutrient (e.g. phosphorus) loading to river systems (Bauer et al. 2002; Walling et al. 2008; Langendoen et al. 2012; Foucher et al. 2017) and is therefore important to quantify as part of comprehensive catchment water quality studies. Measurement of bank erosion and channel change is also a critical part of understanding the geomorphic condition of river systems (Piégay et al. 2005; Kline & Cahoon 2010). Additionally, monitoring bank erosion provides an understanding of the risk to infrastructure and stream habitat posed by fluvial erosion (Kline & Dolan 2008; Thakur et al. 2012).

Several methods exist to measure and monitor streambank erosion and retreat. Traditional direct measurement methods include cross-sectional surveys and bank pins (Lawler 1993; Lawler et al. 1999). Lidar (laser scanning) from both airborne and terrestrial platforms has resulted in more comprehensive and detailed measurement of bank movement (Thoma et al. 2005; Resop & Hession 2010; O'Neal & Pizzuto 2011; Grove et al. 2013) and hillslope and gully erosion (Perroy et al. 2010; Tseng et al. 2013; Pirasteh & Li 2016; Cavalli et al. 2017). Determination of longer term (multiple years or decades) estimates of streambank erosion rates has been successful through combining airborne lidar and historical aerial photos (Rhoades et al. 2009; De Rose & Basher 2011; Garvey 2012) and by applying digital photogrammetry to historical imagery (Bakker & Lane 2017).

A common approach for quantifying geomorphological change involves the creation of digital elevation models (DEMs) from sequential surveys and then subtracting the later

DEM from the earlier DEM; the resulting difference represents land elevation change between the two survey dates. The dataset from the differencing of sequential DEMs is often referred to as a DEM of Difference (DoD). This approach has been utilized with survey data collected using photogrammetry, airborne lidar, and TLS (Milan et al. 2007; Perroy et al. 2010; O'Neal & Pizzuto 2011; Bremer & Sass 2012; Tseng et al. 2013; Grove et al. 2013; Cavalli et al. 2017). Recently, advances in the development of digital photogrammetry methods and unmanned aircraft systems (UAS) platforms have resulted in a resurgence of photogrammetry being used to generate topographic data and DEMs to detect geomorphic change (Westoby et al. 2012; Miřijovský & Langhammer 2015; Eltner et al. 2017; James et al. 2017; Cook 2017).

Advancements in UAS technology, also known as unmanned aerial vehicles (UAVs) or drones, have given rise to a flexible and affordable system for collecting topographic data. UAS-based surveying can overcome some of the existing data collection shortcomings of ground surveys and manned aircraft systems, such as being limited to specific sites, high costs or requiring long data collection lead-times. While DEMs and contours from aerial photography using photogrammetric methods have been available for decades, recent advances in image processing software, driven in part by innovations in computer vision and structure from motion (SfM) and multi-view stereo (MVS) photogrammetric algorithms, have rapidly advanced the resolution of UAS topographic data. In contrast to historical photogrammetry surveying, UAV SfM photogrammetry typically uses only basic camera technology and an automated processing workflow resulting in far lower costs (Westoby et al. 2012; Carbonneau & Dietrich 2017). SfM is ideally suited for processing photos with a high degree of overlap taken from a wide variety of positions (i.e. a moving sensor) (Westoby et al. 2012). Originally developed by the computer vision field during the 1990s, SfM and variations have become widely available in desktop software packages such as Agisoft PhotoScan, Pix4D,

and Microsoft Photosynth. Digital photogrammetric methods such as SfM are applicable to imagery collected using any platform, including handheld smartphone cameras (Micheletti et al. 2015), but have been more widely adopted to process imagery collected using UAS (Hugenholtz et al. 2013).

UAS-based photogrammetric surveying have seen many applications within recent years; reviews by Colomina and Molina (2014), Watts (2012), and Whitehead et al. (2014) highlight UAS characteristics and applications in photogrammetry and remote sensing. Fluvial study applications include mapping bathymetry (Lejot et al. 2007), channel topography (Woodget et al. 2015; A. Tamminga et al. 2015; Miřijovský et al. 2015) and production of very high resolution DEMs (Whitehead & Hugenholtz 2014; Micheletti et al. 2015; Neugirg et al. 2016). In addition, UAS-derived data have shown potential in quantifying bank erosion and monitoring volumetric change in fluvial settings due to flooding (Miřijovský et al. 2015; Miřijovský & Langhammer 2015; A.D. Tamminga et al. 2015; Cook 2017; Hamshaw et al. 2017). However, to date, UAS investigations of river channels have utilizes surveys over a single river reach (i.e. short sections less than a kilometre in length) typically with multi-copter UAS. In addition, applications of UAS for geomorphic change detection along rivers have been limited to areas largely clear of obstructing vegetation. There remains need for evaluation of UAS-based photogrammetry applied over longer sections of river corridor encompassing more varied areas including those with areas of dense vegetation.

In this study, we present an application of UAS-based photogrammetry for monitoring long (approximately 20 km) lengths of river corridors for quantifying streambank erosion rates along multiple rivers in the north-eastern US. Calculations of bank erosion from selected sites are used to illustrate the performance of the system. In addition, we discuss some of the limitations of UAS and recommendations for application in a watershed management setting.

Methods

Study Area

Data collection took place along four rivers (Shepard Brook, Mad River, Winooski River, and New Haven River) located in central Vermont (Figure 3.1). This area of Vermont drains the western portion of the Green Mountain Range and is part of the Lake Champlain basin. The study area features a humid continental climate with mean annual precipitation of 40-60 mm. Soils range from fine sandy loams derived from glacial till deposits in the uplands to silty loams derived from glacial lacustrine deposits in the lowlands. Streambanks in the study area on average are approximately 2 m high, ranging from 1.3 m to 3.8 m high. Vegetation is highly varied and ranges from bare soils to tall grass/brush and tree cover.

All four rivers have a significant history of flooding and resulting channel erosion that dates back to early settlements along the river corridor when historical deforestation of the watershed resulted in river channel destabilization (Underwood 2004; Fitzgerald & Godfrey 2008). During the last two decades, multiple flood events in each of the catchments have resulted in significant river channel erosion causing damage to infrastructure and impacts to water quality. The northeast US is experiencing increase in magnitude and frequency of rainfall events, a trend expected to continue (Betts 2012) making the study regionally relevant. Similar climate changes are predicted elsewhere in the world. As rivers in such regions continue to adjust to a changing hydrological regime, data collected using UAS methods regarding the current state of the river and its geomorphology are useful for broader watershed studies. An affordable method such as UAS could prove to be very useful in tracking rates of streambank erosion if UAS surveys can be done every few years.

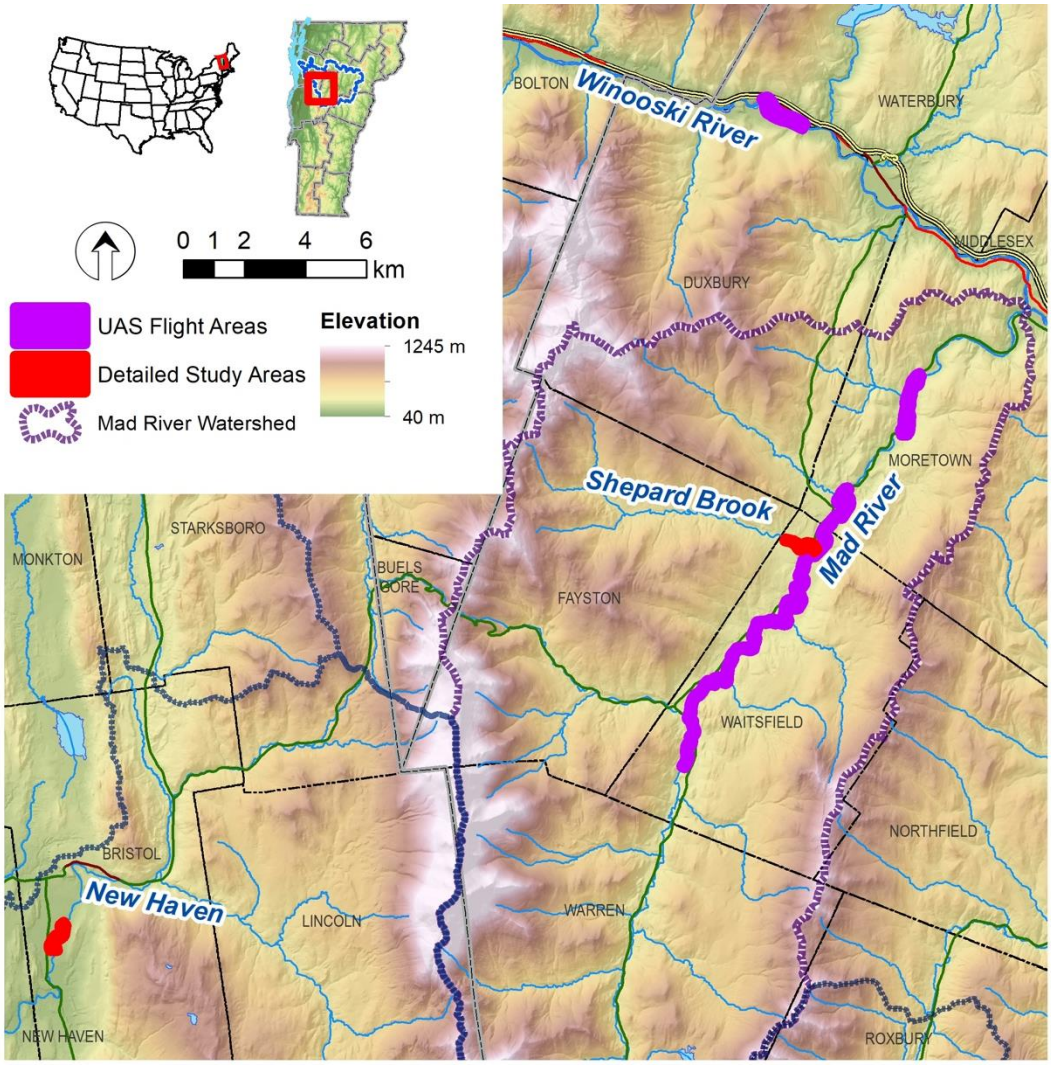


Figure 3.1. Map of study area and portions of river corridor surveyed with UAS

Table 3.1. UAS used in survey campaigns

UAS Model	Survey Campaigns	Camera model	Weight	Wingspan
eBee	Spring 2015	Canon S110 (12 MP)	0.69 kg	96 cm
eBee RTK	Summer/Fall 2015, Spring/Summer 2016	Sony WX (18.2 MP)	0.7 kg	96 cm
eBee Plus	Spring 2017	senseFly S.O.D.A. (20 MP)	1.1 kg	110 cm

Data Collection

UAS surveys took place during a two-year period between spring 2015 and spring 2017. The greatest number of flights were performed in spring months (April-May) when vegetation growth is at a minimum in Vermont and snow has melted. Additional survey campaigns were conducted along portions of the study rivers during summer (August) and late autumn (November-December) months. Topographic data obtained from airborne laser scanning (ALS) surveys were also used to calculate longer-term amounts of erosion along portions of the New Haven River and Shepard Brook. ALS surveys were collected in May 2014 along Mad River and Shepard Brook and in November 2012 for the New Haven River. The 2014 ALS survey was collected at an average point spacing of 0.7 m and the 2012 ALS survey at 1.6 m spacing.

During the study period, a few large storm events resulted in high river flows and caused channel erosion. These include an early spring rainfall event on February 26, 2015 which caused significant bank erosion along the New Haven River and a mid-summer flash flood event on August 17, 2016 which caused moderate bank erosion along Shepard Brook. Additionally, a number of large river flows occurred in the New Haven River in between the 2012 ALS survey and the 2015 UAS survey which resulted in assumed periodic bank erosion. However, between the 2014 ALS survey and 2015 UAS surveys along the Mad River and Shepard Brook, no major storm events occurred, and therefore, stream bank erosion was assumed to be minor.

UAS surveys were performed using a senseFly eBee fixed-wing UAS equipped with an RGB true-color camera (Figure 3.2a). Three models of eBee were used during the study with spring 2015 flights performed using the original eBee model and subsequent campaigns utilizing the eBee RTK or eBee Plus model (Table 3.1). The RTK and plus models distinguish

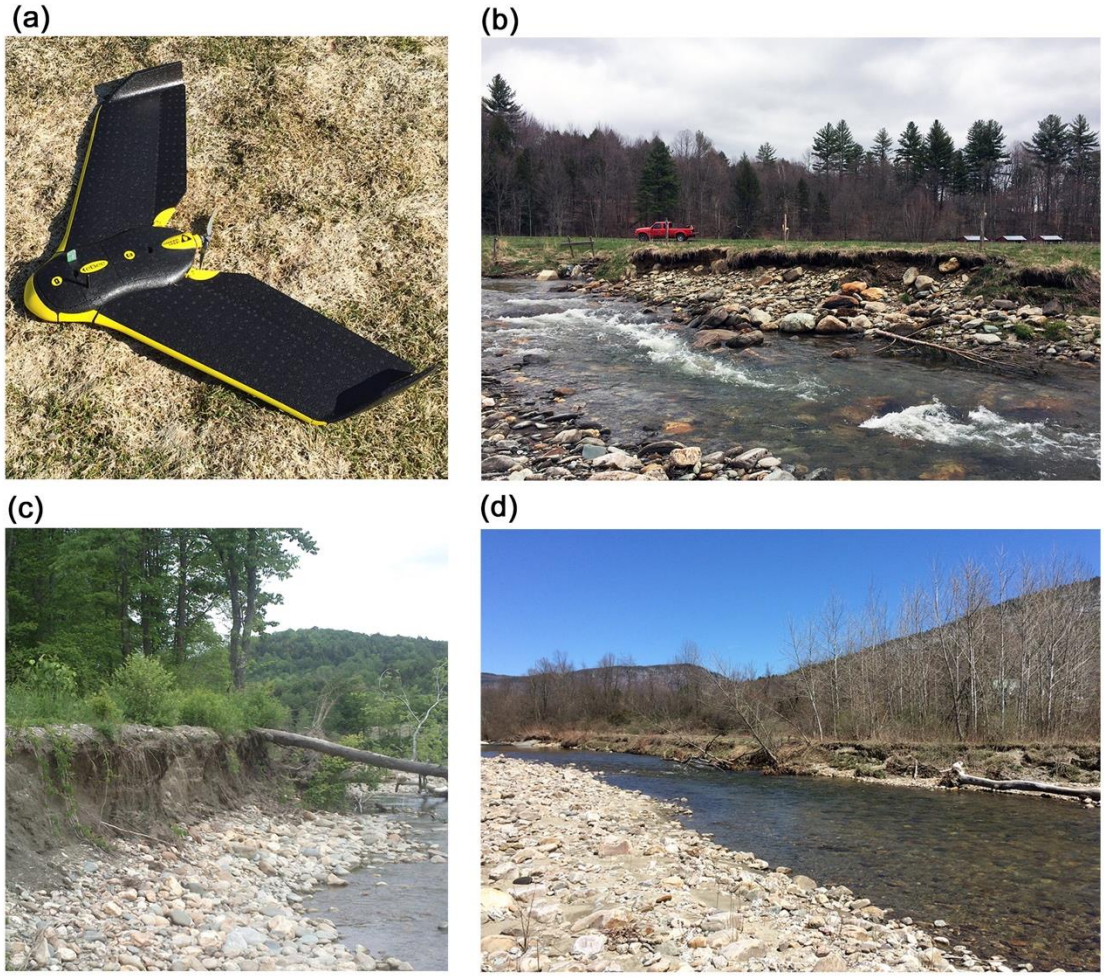


Figure 3.2. (a) senseFly eBee UAS; (b) section of streambank along Shepard Brook in November 2015; (c) example of eroding streambank along Mad River in July 2015 with presence of summer vegetation growth; (d) section of streambank along New Haven River experience erosion in April 2016.

themselves from the standard model by incorporating a survey-grade RTK GPS receiver in order to directly georeference the data with submetre accuracy. For UAS data collected with the standard eBee model, ground control points (GCPs) were utilized to accurately georeference the data. All UAS flights were collected with a target ground sample distance (GSD) of 3.6 cm with a resulting typical altitude above ground level of approximately 100 m. Lateral and longitudinal image overlap were both set to 70%. During spring and late autumn survey campaigns, UAS flights occurred in "leaf-off" conditions when vegetation was minimal.

DEM Analysis

UAS imagery was first post-processed using senseFly's eMotion Version 3.3.4 software package and then passed to the Pix4D Version 4.0.21 (Pix4D, Inc.) software package for photogrammetric processing. Like other digital photogrammetric UAS solutions, Pix4D has a seamless workflow that ingests UAS imagery, generates a 3D point cloud from the overlapping images, and uses the point cloud to produce an orthorectified image mosaic and raster digital surface model. Pix4D also has the capability to automatically generate a DEM (Figure 3.3b), also referred to as a digital terrain model (DTM), from the DSM (Figure 3.3a) and point cloud using a proprietary, machine-learning based algorithm. The UAS-based DEMs had a cell size of 0.15 m compared to 1.0 m and 0.7 cm for the 2012 ALS and 2014 ALS surveys respectively. We note that various methods and software packages are available that enable generation of DSMs and DEMs from point cloud data. We selected to use the automated method of DEM generation provided by the senseFly and Pix4D system as it is representative of an automated and efficient workflow that is practical for generation of topographic data over large areas and many survey campaigns.

The accuracies of DEMs were evaluated through the use of GCPs collected along selected river reaches. GCPs were surveyed using a TopCon HiperLite+ differential GNSS receiver. The GCP positions were collected with the GNSS rover in a semikinematic ("stop-and-go") mode, and the GNSS base station positions were corrected using the Online Position User Service (OPUS) provided by the National Geodetic Survey of the National Oceanic and Atmospheric Administration. DEMs from the airborne lidar surveys (Figure 3.3c) utilized in this study were publicly available from the State of Vermont and are considered a hydro-flattened DEM.

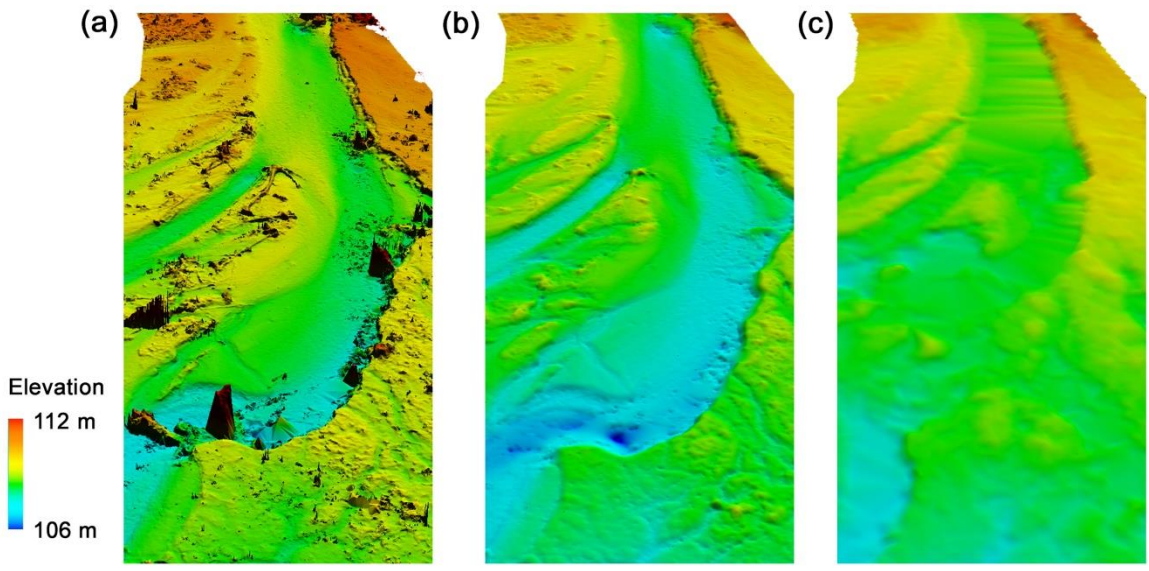


Figure 3.3. View of portion of New Haven River as seen in (a) DSM from April 27, 2016 UAS Survey, (b) DEM generated from April 27, 2016 UAS survey, (c) DEM from 2012 ALS survey

To compare DEMs generated from multiple survey dates, DEMs of difference (DoDs) were generated in Quick Terrain Modeler Version 8.0.4 (Applied Imagery). DoDs were calculated between successive UAS surveys as well as between UAS DEMs and the ALS DEM. DoDs were consistently calculated by subtracting the later date survey from the earlier date resulting in negative values indicating erosion.

Streambank Erosion Calculation

During the study period, one section of the New Haven River experienced significant channel movement and bank erosion (horizontal bank movement > 10 m). This river reach is at high risk for channel erosion and has been subject to previous river channel stabilization efforts. Other sections of the Shepard Brook, Mad River, and Winooski River, that were within the study area, had localized areas of minor to moderate erosion (horizontal bank retreats ~ 1 m). In this paper, we selected two river reaches to highlight analysis of channel change and measurement of streambank erosion: a 1.2 kilometre section of the New Haven River site with

significant channel movement and a 1.5 kilometre section of Shepard Brook with minor bank erosion (Figure 3.1).

Measurements of streambank erosion were determined by calculating a volumetric change along the river corridor. Volume change was determined from the DoD models within a pre-defined river corridor area. The river corridor area was delineated to represent the approximate area subject to river flows during high water level or where potential bank erosion could occur. Both a total negative (erosion) and positive (aggradation) elevation change along the river corridor can be determined as well as a net change.

Results and Discussion

Data acquisition and accuracy

Over the course of a two-year monitoring period, we conducted UAS surveys that covered nearly 50 km of river length. An overview of survey coverage and fieldwork effort is shown in Table 3.2. The greatest number of flights (55) were completed in 2015 where surveys were performed in early spring, mid-summer and late autumn in contrast to 2016 and 2017 where surveys occurred primarily only in early spring. The average length of river surveyed in single flight was 553 m, although longer distances were achieved in 2016 and 2017 surveys where average length of river per flight averaged 760 m and 843 m respectively. This greater efficiency was likely due to a few factors including better optimization of flight lines, fewer equipment issues, and for 2017 surveys, use of the eBee Plus UAV which features greater battery capacity.

The river corridor surveys required a total of 21 full-days in the field to collect. With a total of 49.7 km of river corridor surveyed, the average length per day was 2.37 km collected in approximately four flights (average of 4.3/field day). Rainy and excessively windy weather conditions resulted in requiring rescheduling some field days or cutting them short. Out of 21

survey days, 9 (43%) required rescheduling or shortening. Difficulties in having appropriate weather conditions varied from year-to-year with spring 2016 being especially challenging where all 5 survey days had to be rescheduled. All days that needed to be rescheduled occurred in spring (April and May) when rainfall is more frequent in the north-eastern U.S. than other months of the year.

Table 3.2. Summary of UAS flights and survey coverage

Year	Number of flights	Total Length of River Surveyed (km)	Mean Length of River per Flight (m)	Total days** in field for surveying	Number of days impacted* by weather
2015	55	21.7	395	12	3
2016	18	13.7	760	5	5
2017	17	14.3	843	4	1

* Impacted survey days refer to those that were either cancelled and rescheduled due to rain or wind or those days cut short due to wind or rain.
** A field day was considered 8 hours in the field, with approximately 6 hours available for survey efforts given 2 hours for travel accommodation.

Results of the comparison of DEM values to a set of GPS surveyed GCPs at the two areas is presented in Table 3.3. Mean errors were lowest for the ALS survey at both areas with -0.02 m for New Haven River survey and 0.04 m for Shepard Brook survey. UAS survey performance was highest with the April 2017 surveys. Errors for both ALS and UAS surveys were higher at the Shepard Brook site compared to the New Haven River area. Across all UAS surveys we found an average median error of 0.09 m. This compares well to a previous study that found median vertical errors in UAS-derived topographic data of 0.11 m (Hamshaw et al. 2017).

We found the utilization of a sparse network of GCPs (i.e. 3-4 GCPs per survey area) was helpful to adjust for any overall bias/datum shift and as error check. The use of direct georeferenced topographic data in combination of a small number of GCPs has been found effective also by Carbonneau and Dietrich (2017).

Table 3.3. Assessment of accuracy of DEMs based on comparison to GCPs

New Haven River Site (n = 16)				
	2012 ALS	**2015 UAS**	**2016 UAS**	**2017 UAS**
Mean Error (m)	0.02	0.25	0.12	0.04
Median Error (m)	0.02	0.23	0.08	0.02
Standard Deviation Error (m)	0.09	0.09	0.15	0.12
RMSE (m)	0.09	0.26	0.19	0.12

Shepard Brook Site (n = 10)		
	2014 ALS	**2017 UAS**
Mean Error (m)	0.04	-0.09
Median Error (m)	0.00	0.03
Standard Deviation Error (m)	0.20	0.36
RMSE (m)	0.19	0.35

Calculation of Streambank Erosion

Application to New Haven River

We surveyed along a 1.2 km long river section of the New Haven River that experienced significant bank erosion and river channel movement during the study period. Between the November 2012 ALS survey and December 2015 UAS survey, extensive river channel movement was evident (Figure 3.4) as the result of a number of storm events. Continued erosion along portions of the streambank was evident from subsequent UAS surveys in April 2016 and April 2017. A large amount of erosion was attributed to a February 2016 rain storm that caused high river flows. All UAS surveys were able to be completed during what would be considered "leaf-off" conditions when vegetation growth is minimal and deciduous trees have dropped their leaves. During summer, vegetation and tree cover along this section of the New Haven were fairly extensive (Figure 3.4d). The December 2015

UAS survey ended up covering a smaller area than 2016 and 2017 surveys due to flights being cut short by rainfall.

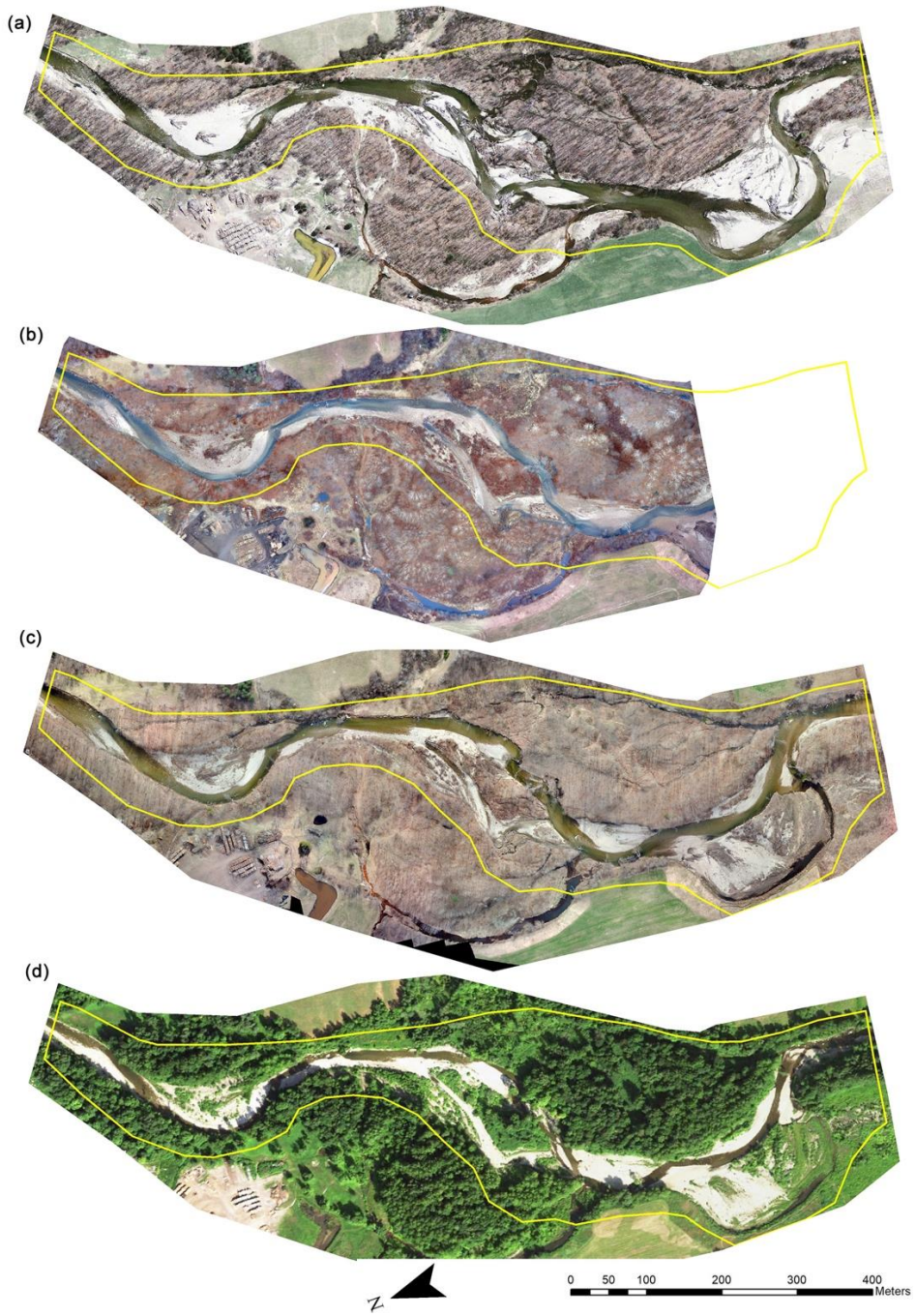


Figure 3.4. Section of the New Haven River as seen in (a) aerial imagery from April 2012, (b) UAS orthomosaic imagery from December 2015, (c) UAS orthomosaic imagery form April 2017, and (d) aerial imagery from July 2016. Area indicated by yellow boundary represents area of river corridor used in analysis of DEMs.

Automated DEM generation from 2016 and 2017 UAS surveys produced high quality topographic data with few obvious vegetation errors and little missing data (Figure 3.5). At the time of spring UAS surveys, vegetation was noticeably less dense than during the December 2015 UAS survey. The presence of areas of denser vegetation along the river during fall can be seen in the December 2015 UAS orthomosaic imagery (Figure 3.4b) in the dark brown areas. We observed, in spring, vegetation was matted down form snowpack resulting in greater visibility of the ground surface.

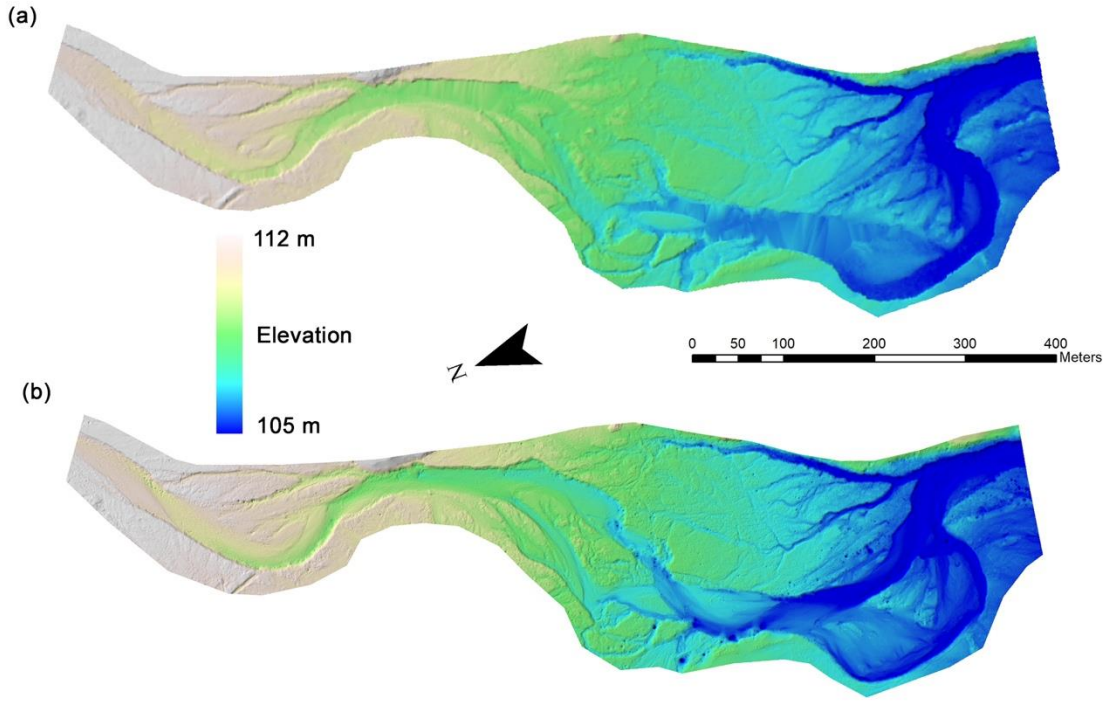


Figure 3.5. Digital elevation model (DEM) of New Haven River produced from (a) 2012 ALS survey and (b) 2017 UAS survey

DoDs generated from multiple date DEMs allowed for spatio-temporal analysis of topographic change within the river corridor area. Between the April 2017 UAS and November ALS 2012 surveys, a net volumetric change of -19,920 m^3 occurred over the 15.2 ha area. The changes included isolated areas of both deposition and erosion (Figure 3.6). In all, an estimated 31,509 m^3 of erosion occurred and 11,589 m^3 of deposition or aggradation

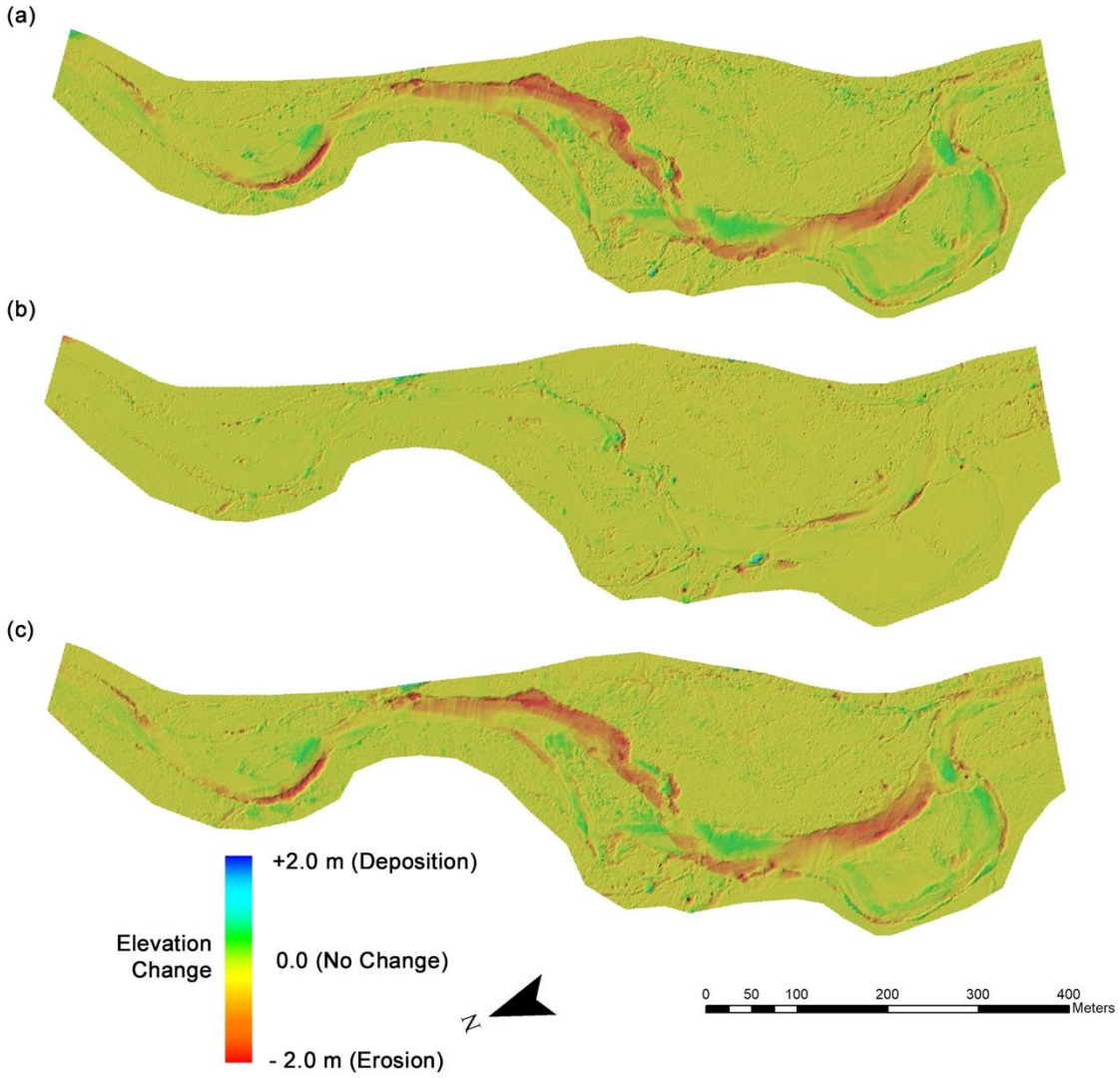


Figure 3.6. Elevation change between surveys along a section of the New Haven River as visualized by DEMs of difference (DoDs) between (a) 2012 ALS survey and 2016 UAS survey, (b) 2016 UAS survey and 2017 UAS survey, and (d) 2012 ALS survey and 2016 UAS survey.

was evident over the nearly five-year period. We also evaluated the geomorphic change at the intermediate survey date of April 2016, which confirmed that the majority of erosion occurred between 2012 and 2016, rather than between 2016 and 2017 (Table 3.4). Of note, the net change calculated from 2012 to 2016 was -14056 m^3 and from 2016 to 2017, -5,866 m^3, giving

a total net change of -19,922 m^3, which is consistent with the direct measurement of 2017-2012 change above. The average annual rate of volumetric erosion was ~6,300 m^3/year. If an average stream bank height of 1.9 m (based on field measurements) is assumed over the entire 1,200 m long river reach, the average annual rate of bank retreat was 1.4 m/yr/m.

Table 3.4. Summary of volumetric change of surface within river corridor area

DoD model	Positive Change (Deposition) m^3	Negative Change (Erosion) m^3	Net Change m^3
2017 UAS – 2012 ALS	11,589	31,509	-19,920
2017 UAS – 2016 UAS	5,503	11,369	-5,866
2016 UAS – 2012 ALS	13,848	27,904	-14,056

We utilized the automated DTM (DEM) generation capability integrated into Pix4D Mapper (Version 4.0.5). Version 4 of Pix4D was released during the study period and highlights that the processing of UAS imagery and SfM photogrammetric techniques, is a rapidly evolving area characterized by frequent software releases. With a number of proprietary algorithms used in various software packaged we note that it would be expected that different software packages produce slightly different DEMs. While we did not study the impact of different software packages, Ouédraogo et al. (2014) found differences in DEM generation from two different software packages, Agisoft PhotoScan and MicMac, resulted in DEMs with difference in root mean square error (RMSE) of 4.9 cm. Vallet et al. (2012) found a similar scale difference in mean error of 6.3 cm between DTMs generated by Pix4D and by a different photogrammetric process using SocetSet NGATE. These scale differences were minor compared to the scale of geomorphic changes we intended to quantify in our study, and therefore, we do not believe would impact our conclusions. Topographic change detection

and volume calculations have also been made through comparison of photogrammetric point clouds instead of derived DEMs with promising results (Cook 2017). However, we elected to use a more conventional DEM analysis since raster datasets are readily compatible in common spatial analysis software packages such as ESRI ArcGIS that offer many tools for analysing raster datasets.

The presence of differences in water surface and vegetation growth were potential sources of additional error we identified in the DEMs. Errors due to vegetation in the 2016 and 2017 UAS DEMs were not significant as evidenced by little observed elevation change in areas that had significant vegetation. Based on a comparison of water surface across a stable portion of the river we found differences in the DEM of ~0.2 m between the UAS surveys and the ALS DEM. Negligible differences were observed between the 2016 and 2017 UAS surveys. While it is possible that river bed lowering occurred during the study period, we did not collect simultaneous field measurements of bathymetry, and given the reliability of SfM techniques for measuring bathymetry (Cook 2017), our analysis did not provide conclusive evidence of bathymetric changes. Studies have shown that bathymetric UAS measurements can be improved through refraction correction (Lejot et al. 2007; Dietrich 2017) to reduce errors.

The DEM quality from the December 2015 UAS survey was poorer in contrast to the April 2016 and 2017 surveys. Observation of the DoDs (Figure 3.7) revealed significant areas of measured deposition in places where no observed deposition occurred. In referring to the aerial imagery (Figure 3.4), these areas correspond to denser vegetation areas and show significant interpolation and smoothing in the DEMs. Errors in the 2015 DEM due to vegetation are also evident in the volumetric change between the 2017 – 2015 DoD (Figure 3.7b), which showed a net change of 1,401 m^3 when known erosion occurred and a negative

net change was expected. The December 2015 UAS flight was also conducted in light rain conditions, which resulted in poorer image contrast when compared to the spring 2016 and 2017 images. The combination of greater density of vegetation in late Autumn and possibly other factors made DoD calculations unreliable using the 2015 UAS survey. However, areas of significant erosion can be clearly identified in the data set, and therefore, measurement of erosion at specific individual areas would be required as opposed to measurement over the entire river corridor area.

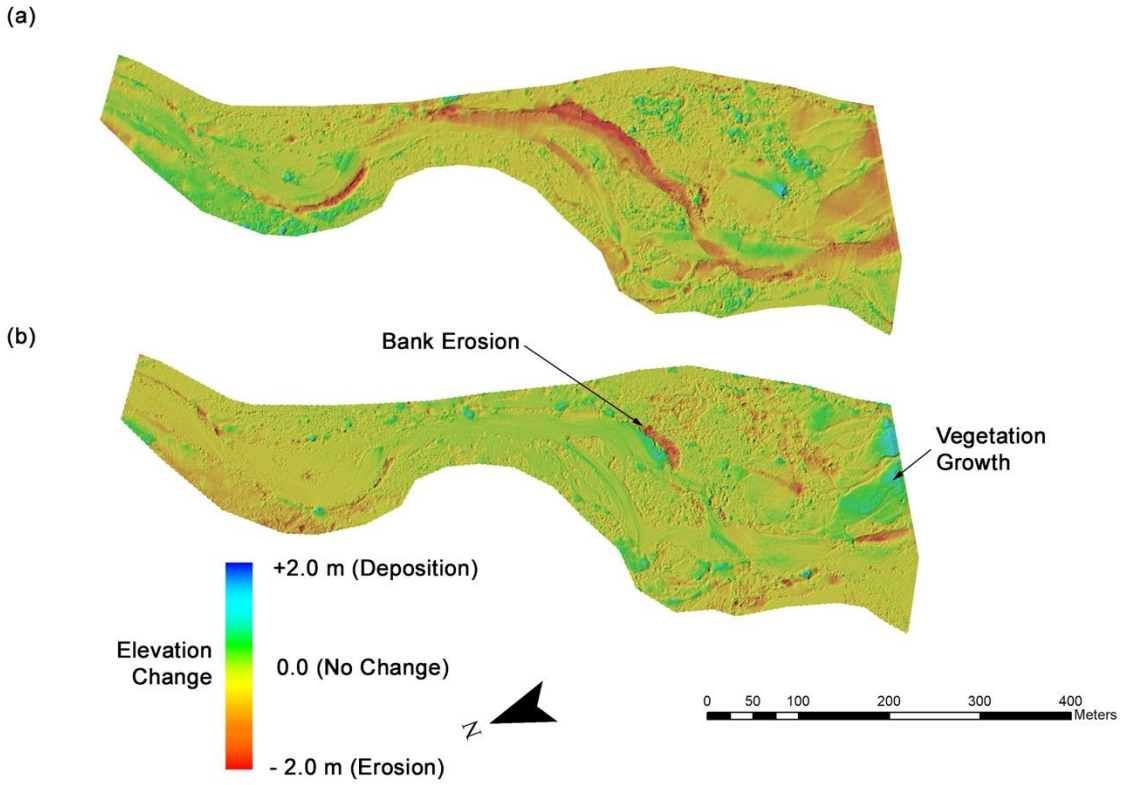


Figure 3.7. DoD for New Haven River as calculated from (a) 2012 ALS survey and 2015 UAS survey and (b) 2015 UAS survey and 2017 UAS survey

Application to Shepard Brook

UAS surveys collected along a portion of Shepard Brook (Figure 3.1) were also used to quantify channel movement over a river reach with observed erosion, but with different

characteristics than the New Haven River. In contrast to the New Haven River area, Shepard Brook has denser vegetation and greater tree cover in the river corridor; it is a smaller river with shorter streambanks (~1.2 m high), and it is less susceptible to channel movement and bank erosion. Between the May 2014 ALS survey and April 2017 UAS survey, several medium size storm events caused minor observable erosion in isolated locations. A short duration flash flood event in summer 2016 caused the greatest amount of bank erosion, but still on only short sections (such as site shown in Figure 3.8) and with less than 1 m of retreat over 3 years.

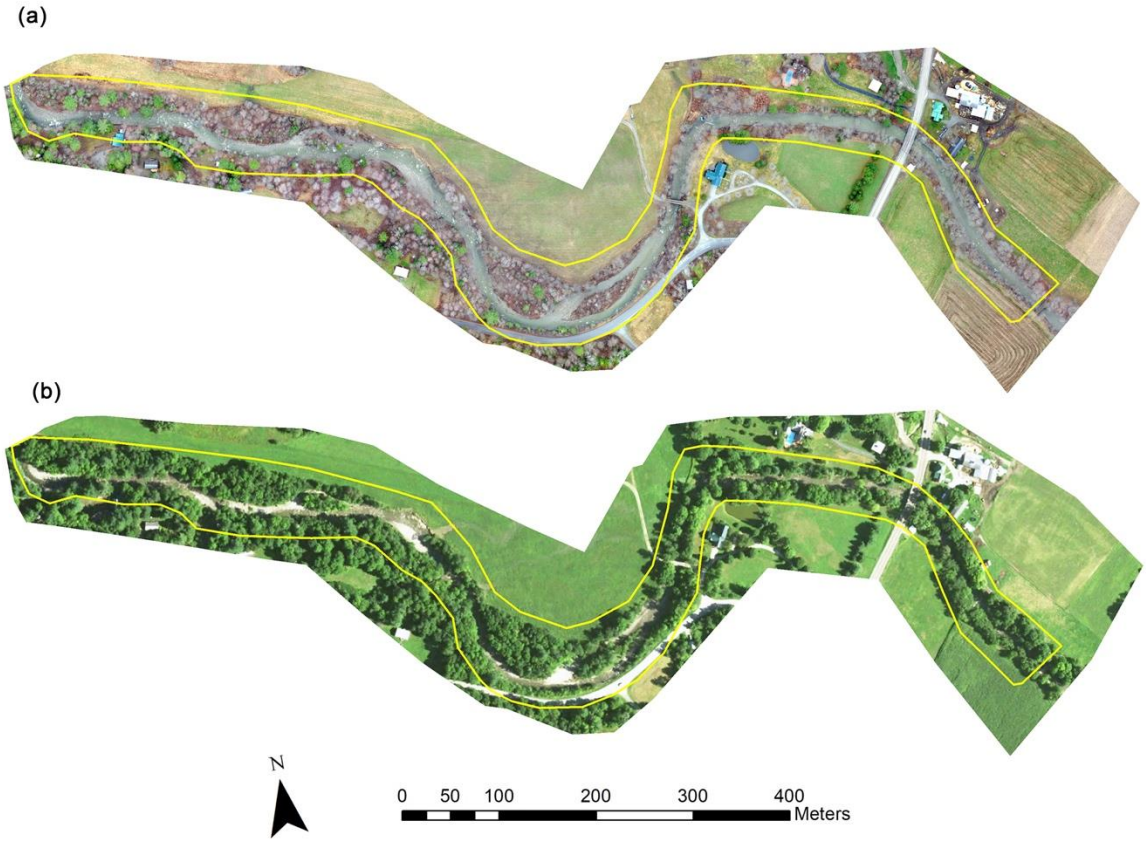


Figure 3.8. Section of Shepard Brook as seen in (a) UAS orthomosaic imagery from April 2017 and (b) aerial imagery from July 2016. Area indicated by yellow boundary represents area of river corridor used in analysis of DEMs.

Intermediate UAS surveys were also completed in April 2015, August 2015, November 2015, May 2016, and August 2016. In analysing geomorphic change, we only considered in detail the

April 2017 UAS survey in comparison to the 2015 ALS survey in order compute the greatest amount of change represented in the two DEMs used to generate the DoD.

Errors in the UAS DEMs were more prevalent at the Shepard Brook site than at the NHR site. Large areas of smoothed/interpolated data were present in areas of thick trees cover where the UAS imagery could not reliably observe the ground surface (Figure 3.9). Similar missing data resulting from smoothing can be observed on much of the streambank area. In comparison to the New Haven River, Shepard Brook has greater tree cover along the streambanks, which may explain the poorer performance. This can be observed in the DoD between 2017 UAS and 2014 ALS survey (Figure 3.9c). The large areas of vegetation along the banks resulted over-estimated erosion values along many portions of the river channel. This is also evident in the measurement of volumetric change over the river corridor which was showed a likely innacurate net change of 13,372 m^3, with respective measurements of total positive change (deposition) of 21,76 6 m^3 and total negative change of 8,034 m^3. However, an area with active erosion (shown in Figure 3.9b) within an area with less vegetation is easily visible in the DoD showing as bank erosion (Figure 3.9c). The results of DEM generation from Shepard Brook indicates that in densely vegetated river corridors, including those with a number of evergreen trees, a greater erosion threshold is necessary for UAS survey to be reliable. Additionally, measurements of erosion may be most successful if performed over specific smaller areas where DEMs are known to be representative of the actual ground surface.

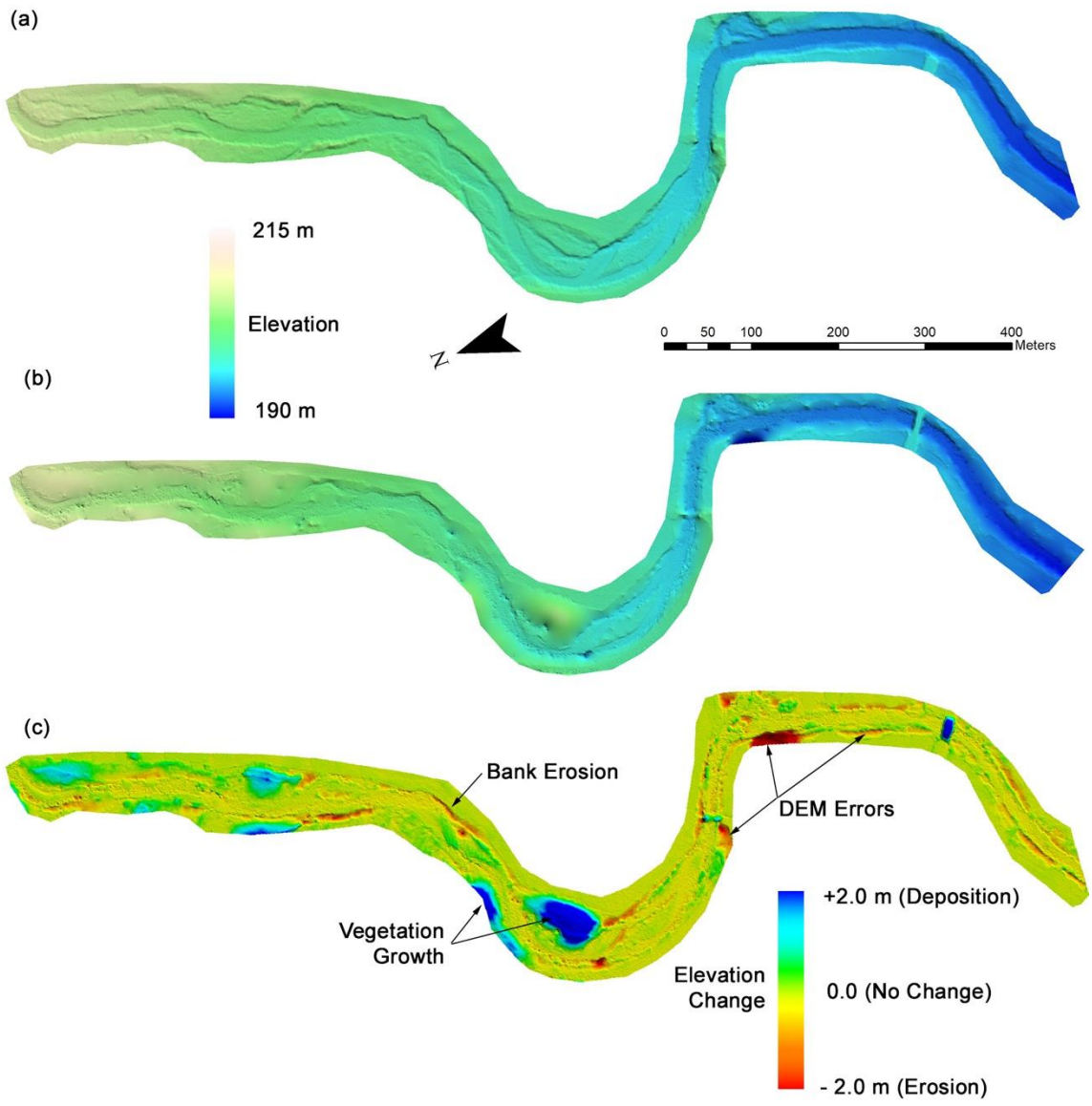


Figure 3.9. Digital elevation models (DEMs) of New Haven River produced from (a) 2014 ALS survey and (b) 2017 UAS survey and (c) DEM of difference (DoD) calculated from 2017 UAS survey – 2014 ALS survey.

Characteristics of river corridor and relation to bank erosion measurement

The ability to detect geomorphic change the river corridor using topographic data is a function of the magnitude of change, resolution of the topographic data, and amount of error and noise in the data. We observed the primary source of noise in the topographic data was due to the presence of heavy vegetation. Given that photogrammetric methods such as UAS-

based SfM are line-of-sight survey methods that are dependent on being able to observe the surface of interest, the effect of vegetation that obscures or partially obscures the data is expected. This is consistent with previous findings that dense vegetation can cause large errors (Cook 2017; Hamshaw et al. 2017). As previously noted by Cook (2017), SfM techniques are capable of filtering out sparse vegetation effectively. We observed similar results at the New Haven River site, where spring UAS DEMs reliably captured the ground surface, filtering out the presence of sparse ground vegetation and trees. Therefore, we found that the usefulness of UAS-based photogrammetry for capturing streambank topography is driven more by the effective density of vegetation than the absolute presence.

The automated DEM generation method used in our study is surprisingly robust in filtering out noise due to sparse vegetation and trees. In climates similar to the north-eastern U.S., the timing of survey is critical as river corridors with deciduous tree cover and grass/brush are best surveyed in early spring after snowmelt but prior to summer vegetation growth. This confirms previous findings that assessed the ability to capture streambank topography using UAS by Hamshaw et al. (2017) and found early spring conditions had much lower errors than summer and autumn conditions. River corridors that feature year-round vegetation (i.e. tropical and subtropical climates) or the dominance of evergreen vegetation will offer limited opportunity for photogrammetric methods such as SfM.

Challenges and Recommendations for UAS river corridor monitoring

This study utilizes the rapidly advancing technology of UAS and digital photogrammetry in surveying river corridors for the monitoring of streambank erosion. Many previous studies focused on acquisition and assessment of UAS-based topographic data along a single, short river reach. In contrast, we collected survey data over approximately 20 km over a varied set of river reaches. Studies seeking to evaluate geomorphic change are necessarily

dependent on the timing of surveys capturing the land surface pre- and post- significant storm events. While our various study areas were selected in part because of a known occurrences of bank erosion and susceptibility to continued erosion, only very limited areas of significant bank erosion occurred as no large flood events took place between survey dates. Therefore, while we have highlighted the application of UAS-based photogrammetry along two river sections, the evaluation of UAS-based bank erosion quantification along many river reaches remains to be evaluated.

We also note there are differences in topography and land cover between our study area and many demonstrated applications of UAS for geomorphic change detection. In the northeast U.S., many river corridors are purposefully protected to preserve vegetation and tree cover, which presents a challenge to remote sensing-based survey methods such as photogrammetry. However, because of the flexibility in survey timing offered by UAS, we were able to wait for optimal survey conditions in order to acquire high quality topographic data along many river sections. In the process, we encountered several challenges in completing and processing data due in part to the use of an emerging technology, which in certain aspects is still in infancy, but at the same time rapidly advancing in some aspects. We make the following recommendations as lessons learned for future applications of UAS for surveying along river corridors including the application to geomorphic change detection and streambank erosion measurement.

1. For applications in continental and temperate climates, we recommend surveys be performed in spring conditions after snowmelt and prior to "leaf-out" to minimize errors caused by vegetation. Late-autumn conditions may also be appropriate provided there is not significant dead standing vegetation/brush still present.

2. The ability of the UAS-based photogrammetric method to capture the ground surface is dependent on the density of vegetation, not just the presence of vegetation. We recommend confirming through site visits or from historical

imagery whether at any times of the year, the area of interest is relatively free of dense, obscuring vegetation rather than relying on basic presence/absence of trees or vegetation in planning surveys.

3. For climates similar to the north-eastern U.S., we recommend planning on accommodating one third of planned survey days to be rescheduled due to weather conditions. In our study, we found that 43% of survey days were impacted by excessive wind or rain.

4. We found that a UAS with accurate, direct georeferencing capabilities such as the eBee RTK and RTK-enabled eBee Plus greatly, simplified field data collection because they eliminated the need for GCPs. However, to achieve maximal accuracy or to accommodate a workflow utilizing lower cost UAVs, we recommend the collection of at least a sparse network of GCPs encompassing the entire survey area.

Conclusions and Future Work

The UAS application to monitoring of river corridors for streambank erosion presented here provides a cost-effective and efficient way to obtain high-resolution topography data on river corridors. While accuracy depended heavily on the density of vegetation present, we were able to capture high quality DEMs along river corridors with significant tree canopy and vegetation provided surveys were conducted in early spring when optimal ground conditions occur. We utilized an automated workflow for georeferencing UAS-derived topography and generating DEMs that then allowed the direct comparison of multiple survey dates or to airborne lidar surveys by using a differencing of DEMs approach. The ability to calculate the volume of erosion and deposition along the entire river corridor provides a better understanding of the rate and pattern of bank erosion.

Given sufficient planning and selection of survey dates to achieve optimal vegetation and weather conditions, UAS-based photogrammetry provides topographic data that improves upon the resolution of currently available airborne lidar survey data. UAS technology is a rapid growth area and new camera sensor technology, improvements in photogrammetric

software and processing algorithms, and the direct georeferencing capability of GPS equipped UAVs should both improve the utility and performance of future systems.

Acknowledgements

This research was supported by the Vermont Water Resources and Lakes Studies Center. Additional support provided by Vermont EPSCoR with funds from the National Science Foundation (NSF) Grant EPS-1101317 and EPS-1556770, NSF Grant CMMI-1229045, NSF Graduate Research Fellowship under Grant DGE-0925179NSF, the Robert & Patricia Switzer Foundation, and Grant OASRTRS-14-H- UVM from the US Department of Transportation is also gratefully acknowledged. The views, opinions, findings, and conclusions reflected in this paper are solely those of the authors and do not represent the official policy or position of any funding sources or endorse any third-party products or services that may be included in this presentation or associated materials. The authors acknowledge the additional contributions of the UVM Spatial Analysis Lab UAS team and Kristen Underwood.

References

Bakker M, Lane SN. 2017. Archival photogrammetric analysis of river–floodplain systems using Structure from Motion (SfM) methods. Earth Surf Process Landf. 42:1274–1286.

Bauer DW, Mulla DJ, Sekely AC. 2002. Streambank slumping and its contribution to the phosphorus and suspended sediment loads of the Blue Earth River, Minnesota. J Soil Water Conserv. 57:243+.

Betts AK. 2012. Historic Trends and Future Climatic Projections for Vermont.

Bremer M, Sass O. 2012. Combining airborne and terrestrial laser scanning for quantifying erosion and deposition by a debris flow event. Geomorphology. 138:49–60.

Carbonneau PE, Dietrich JT. 2017. Cost-effective non-metric photogrammetry from consumer-grade sUAS: implications for direct georeferencing of structure from motion photogrammetry. Earth Surf Process Landf. 42:473–486.

Cavalli M, Goldin B, Comiti F, Brardinoni F, Marchi L. 2017. Assessment of erosion and deposition in steep mountain basins by differencing sequential digital terrain models. Geomorphology. 291:4–16.

Colomina I, Molina P. 2014. Unmanned aerial systems for photogrammetry and remote sensing: A review. ISPRS J Photogramm Remote Sens. 92:79–97.

Cook KL. 2017. An evaluation of the effectiveness of low-cost UAVs and structure from motion for geomorphic change detection. Geomorphology. 278:195–208.

De Rose RC, Basher LR. 2011. Measurement of river bank and cliff erosion from sequential LIDAR and historical aerial photography. Geomorphology. 126:132–147.

Dietrich JT. 2017. Bathymetric Structure-from-Motion: extracting shallow stream bathymetry from multi-view stereo photogrammetry. Earth Surf Process Landf. 42:355–364.

Eltner A, Kaiser A, Abellan A, Schindewolf M. 2017. Time lapse structure from motion photogrammetry for continuous geomorphic monitoring. Earth Surf Process Landf [Internet]. [cited 2017 Jun 7]. Available from: http://onlinelibrary.wiley.com/doi/10.1002/esp.4178/abstract

Fitzgerald EP, Godfrey LC. 2008. Upper Mad River Corridor Plan. Waitsfield, VT: Friends of the Mad River.

Foucher A, Salvador-Blanes S, Vandromme R, Cerdan O, Desmet M. 2017. Quantification of bank erosion in a drained agricultural lowland catchment. Hydrol Process. 31:1424–1437.

Garvey KM. 2012. Quantifying Erosion and Deposition Due to Stream Planform Change Using High Spatial Resolution Digital Orthophotography and Lidar Data [M.S. Thesis]. Burlington, VT: University of Vermont.

Grove JR, Croke J, Thompson C. 2013. Quantifying different riverbank erosion processes during an extreme flood event. Earth Surf Process Landf. 38:1393–1406.

Hamshaw SD, Bryce T, Rizzo DM, O'Neil-Dunne J, Frolik J, Dewoolkar MM. 2017. Quantifying streambank movement and topography using unmanned aircraft system photogrammetry with comparison to terrestrial laser scanning. River Res Appl. 33:1354–1367.

Hugenholtz CH, Whitehead K, Brown OW, Barchyn TE, Moorman BJ, LeClair A, Riddell K, Hamilton T. 2013. Geomorphological mapping with a small unmanned aircraft system (sUAS): Feature detection and accuracy assessment of a photogrammetrically-derived digital terrain model. Geomorphology. 194:16–24.

James MR, Robson S, Smith MW. 2017. 3-D uncertainty-based topographic change detection with structure-from-motion photogrammetry: precision maps for ground control and directly georeferenced surveys. Earth Surf Process Landf [Internet]. [cited 2017 Mar 23]. Available from: http://onlinelibrary.wiley.com/doi/10.1002/esp.4125/abstract

Kline M, Cahoon B. 2010. Protecting River Corridors in Vermont1. JAWRA J Am Water Resour Assoc. 46:227–236.

Kline M, Dolan K. 2008. River Corridor Protection Guide: Fluvial Geomorphic-Based Methodology to Reduce Flood Hazards and Protect Water Quality.

Langendoen EJ, Simon A, Klimetz L, Natasha B, Ursic ME. 2012. Quantifying Sediment Loadings from Streambank Erosion in Selected Agricultural Watersheds Draining to Lake Champlain. Grand Isle, VT: Lake Champlain Basin Program.

Lawler DM. 1993. The measurement of river bank erosion and lateral channel change: A review. Earth Surf Process Landf. 18:777–821.

Lawler DM, Grove JR, Couperthwaite JS, Leeks GJL. 1999. Downstream change in river bank erosion rates in the Swale-Ouse system, northern England. Hydrol Process. 13:977–992.

Lejot J, Delacourt C, Piégay H, Fournier T, Trémélo M-L, Allemand P. 2007. Very high spatial resolution imagery for channel bathymetry and topography from an unmanned mapping controlled platform. Earth Surf Process Landf. 32:1705–1725.

Micheletti N, Chandler JH, Lane SN. 2015. Investigating the geomorphological potential of freely available and accessible structure-from-motion photogrammetry using a smartphone. Earth Surf Process Landf. 40:473–486.

Milan DJ, Heritage GL, Hetherington D. 2007. Application of a 3D laser scanner in the assessment of erosion and deposition volumes and channel change in a proglacial river. Earth Surf Process Landf. 32:1657–1674.

Miřijovský J, Langhammer J. 2015. Multitemporal Monitoring of the Morphodynamics of a Mid-Mountain Stream Using UAS Photogrammetry. Remote Sens. 7:8586–8609.

Miřijovský J, Michalková MŠ, Petyniak O, Máčka Z, Trizna M. 2015. Spatiotemporal evolution of a unique preserved meandering system in Central Europe — The Morava River near Litovel. CATENA. 127:300–311.

Neugirg F, Stark M, Kaiser A, Vlacilova M, Della Seta M, Vergari F, Schmidt J, Becht M, Haas F. 2016. Erosion processes in calanchi in the Upper Orcia Valley, Southern Tuscany, Italy based on multitemporal high-resolution terrestrial LiDAR and UAV surveys. Geomorphology. 269:8–22.

O'Neal MA, Pizzuto JE. 2011. The rates and spatial patterns of annual riverbank erosion revealed through terrestrial laser-scanner surveys of the South River, Virginia. Earth Surf Process Landf. 36:695–701.

Ouédraogo MM, Degré A, Debouche C, Lisein J. 2014. The evaluation of unmanned aerial system-based photogrammetry and terrestrial laser scanning to generate DEMs of agricultural watersheds. Geomorphology. 214:339–355.

Perroy RL, Bookhagen B, Asner GP, Chadwick OA. 2010. Comparison of gully erosion estimates using airborne and ground-based LiDAR on Santa Cruz Island, California. Geomorphology. 118:288–300.

Piégay H, Darby SE, Mosselman E, Surian N. 2005. A review of techniques available for delimiting the erodible river corridor: a sustainable approach to managing bank erosion. River Res Appl. 21:773–789.

Pirasteh S, Li J. 2016. Landslides investigations from geoinformatics perspective: quality, challenges, and recommendations. Geomat Nat Hazards Risk.:1–18.

Resop JP, Hession WC. 2010. Terrestrial laser scanning for monitoring streambank retreat: comparison with traditional surveying techniques. J Hydraul Eng. 136:794–798.

Rhoades EL, O'Neal MA, Pizzuto JE. 2009. Quantifying bank erosion on the South River from 1937 to 2005, and its importance in assessing Hg contamination. Appl Geogr. 29:125–134.

Tamminga A, Hugenholtz C, Eaton B, Lapointe M. 2015. Hyperspatial Remote Sensing of Channel Reach Morphology and Hydraulic Fish Habitat Using an Unmanned Aerial Vehicle (UAV): A First Assessment in the Context of River Research and Management. River Res Appl. 31:379–391.

Tamminga AD, Eaton BC, Hugenholtz CH. 2015. UAS-based remote sensing of fluvial change following an extreme flood event. Earth Surf Process Landf. 40:1464–1476.

Thakur PK, Laha C, Aggarwal SP. 2012. River bank erosion hazard study of river Ganga, upstream of Farakka barrage using remote sensing and GIS. Nat Hazards. 61:967–987.

Thoma DP, Gupta SC, Bauer ME, Kirchoff CE. 2005. Airborne laser scanning for riverbank erosion assessment. Remote Sens Environ. 95:493–501.

Tseng C-M, Lin C-W, Stark CP, Liu J-K, Fei L-Y, Hsieh Y-C. 2013. Application of a multi-temporal, LiDAR-derived, digital terrain model in a landslide-volume estimation: multi-temporal LiDAR DTM in landslide volume estimation. Earth Surf Process Landf.:1587–1601.

Underwood KL. 2004. Phase 2 Stream Geomorphic Assessment New Haven River Watershed Addison County, Vermont. Middlebury, Vermont: Addison County Regional Planning Commission.

Vallet J, Panissod F, Strecha C, Tracol M. 2012. Photogrammetric performance of an ultra light weight swinglet UAV. ISPRS - Int Arch Photogramm Remote Sens Spat Inf Sci. XXXVIII-1/C22:253–258.

Walling DE, Collins AL, Stroud RW. 2008. Tracing suspended sediment and particulate phosphorus sources in catchments. J Hydrol. 350:274–289.

Watts AC, Ambrosia VG, Hinkley EA. 2012. Unmanned Aircraft Systems in Remote Sensing and Scientific Research: Classification and Considerations of Use. Remote Sens. 4:1671–1692.

Westoby MJ, Brasington J, Glasser NF, Hambrey MJ, Reynolds JM. 2012. "Structure-from-Motion" photogrammetry: A low-cost, effective tool for geoscience applications. Geomorphology. 179:300–314.

Whitehead K, Hugenholtz CH. 2014. Remote sensing of the environment with small unmanned aircraft systems (UASs), part 1: a review of progress and challenges. J Unmanned Veh Syst. 2:69–85.

Whitehead K, Hugenholtz CH, Myshak S, Brown O, LeClair A, Tamminga A, Barchyn TE, Moorman B, Eaton B. 2014. Remote sensing of the environment with small unmanned aircraft systems (UASs), part 2: scientific and commercial applications. J Unmanned Veh Syst. 2:86–102.

Woodget AS, Carbonneau PE, Visser F, Maddock IP. 2015. Quantifying submerged fluvial topography using hyperspatial resolution UAS imagery and structure from motion photogrammetry. Earth Surf Process Landf. 40:47–64.

CHAPTER 4. A NEW MACHINE-LEARNING APPROACH FOR CLASSIFYING HYSTERESIS IN SUSPENDED-SEDIMENT DISCHARGE RELATIONSHIPS USING HIGH-FREQUENCY MONITORING DATA

Abstract

Studying the hysteretic relationships embedded in high-frequency suspended sediment concentration and river discharge data over individual storm events provides insight into the drivers and sources of riverine sediment during events. However, the literature remains limited to analyses using simple visual classifications (linear, clockwise, counter-clockwise, and figure-eight patterns) or the collapse of these patterns to a hysteretic index. This study helps automate the assessment of event sediment dynamics through the use of machine learning and three years of high-frequency suspended sediment and discharge data collected from a medium-sized watershed and five of its tributaries. Across all sites, 600 events were captured and fourteen different types of hysteresis were identified. Event classification was automated by training a restricted Boltzmann machine (RBM), a type of artificial neural network, on images of the suspended sediment-discharge plots. The expanded classification allowed for new insight into drivers of hysteresis types including spatial scale, antecedent conditions, hydrology and rainfall. The probabilistic RBM classification network predicted the correct or next most similar class 71% of the time. With increased availability of high-frequency suspended sediment data, the hysteretic classification approach presented here can be used to inform watershed management efforts to identify sediment sources and reduce fine sediment export.

Introduction

Quantifying the relationship between riverine sediment export and discharge provides important information for understanding the state of hydrologic systems, ecosystem disturbances/stressors, with implications for downstream water quality. In particular, export of suspended sediment plays a critical role in sediment pollution, water-quality degradation, and ecosystem impairment.[1] The association of suspended sediment and sediment-bound nutrients such as particulate phosphorus motivates better characterization and understanding of watershed sediment dynamics, nutrient loading, and potential risks (e.g., eutrophication) to aquatic ecosystems.[2]

Processes associated with suspended sediment transport during hydrological events can be referred to as event sediment dynamics. Information on these dynamics as well as the sediment source may be inferred from the storm-runoff response and corresponding sediment concentration response.[3–5] Given the underlying complexity and non-linear processes controlling sediment transport, the linkage between suspended sediment and discharge over single storm events often cannot be described by simple linear or univariate relationships.[6] This has given rise to numerous studies on suspended sediment concentration-discharge (SSC-Q) relationships, frequently observed as hysteretic in nature.[4,5,7–10]

Williams[10] appears to be the first to systematically describe single event SSC-Q hysteretic behavior (e.g., linear, clockwise, counter-clockwise, figure-eight loops and a few variations) using shapes and timing of the hydrograph and corresponding sedigraph, and offers examples of physical watershed processes that cause such patterns. Subsequent studies (1) identified these types of hysteresis in the SSC-Q relationships,[11,12] (2) validated the physical processes that give rise to these patterns and then,[7,8,13,14] (3) inferred sediment dynamics occurring in the study watersheds.[9,15–17] Studies also demonstrated broad applicability of the

hysteresis patterns beyond the SSC-Q relationship; including, but not limited to, the relationship between discharge and soil moisture,[18] discharge to other solute concentrations,[19,20] and suspended sediment and turbidity.[21]

SSC-Q hysteresis has also been used to identify (1) the relative contribution of in-stream sediment sources and more distant hillslope sources to overall sediment yields,[9,16,17,22–24] (2) whether individual watersheds are supply-limited or transport-limited,[3,4,7,19] and (3) the predominant sediment source, such as bank erosion.[5,17] Studies on the temporal effects on the predominant watershed hysteresis types have inferred seasonal dynamics of sediment supply and transport,[4,5,9,16,17,23,24] and although less extensively, the effect of watershed size,[3,17,22] to understand how tributary sediment delivery differs from that of the downstream, main channel outlet.

Classification of hysteretic SSC-Q patterns is important when making inferences about sediment dynamics. To date, researchers have utilized either a qualitative visual classification or hysteresis indices (HI)[25] to quantitatively measure differences in the rising and falling limbs of the SSC-Q relationship. Various metrics have been proposed to automate and objectively classify hysteretic behavior.[17,18,21,22,25–27] Some indices facilitate compression of information on the shape and pattern of the SSC-Q relationship into a single metric that helps infer event sediment dynamics without the need for classification. However, HI values are not unique (i.e., individual storm events with different hysteresis patterns can have the same index value) and therefore, often require additional metrics such as loop area or direction to preserve information lost during data compression.[18,25]

Machine learning methods can help identify patterns in hydrological data. For example, feed-forward backpropagation algorithms have long been used in rainfall-runoff modeling and streamflow prediction.[28,29] More recently, a new variety of pattern recognition

networks called deep belief neural networks (DBNNs) excel at classification applications such as hand-written character recognition,[30] sparking extensive research into deep learning. One building block of the DBNN is the restricted Boltzmann machine (RBM), which acts as a feature extractor for pattern recognition and classification.[31] The suite of RBM algorithms, now readily available in a variety of computer languages that run on a desktop PC, make them attractive in balancing state-of-the art performance with ease of implementation.[32]

This study leverages three years of high-resolution riverine suspended sediment time series from multiple sites to show proof-of-concept of (1) expanding the existing visual classification system of storm-event suspended sediment hysteresis patterns, and (2) automating the classification of event hysteresis using a novel machine-learning technology designed for high-frequency environmental monitoring data. We then illustrate the utility of the technology for understanding the environmental drivers of suspended sediment dynamics during storms and suspended sediment provenance. The discussion includes implications and opportunities for watershed management communities, future applications, and modifications of this approach.

Methods

Study Area

The Mad River watershed, located in the Green Mountains of Vermont within the Lake Champlain Basin (Figure 4.S1), was selected as a study site based on available long-term stream gauge records and ongoing geomorphic and sediment dynamics studies.[33,34] Elevation ranges from 132 m to 1,245 m above sea level, with forests (83% of watershed area) dominating all but the valley floors, which are occupied by agriculture (8%) and village centers and other developed lands (8%) (Table 4.S1). Soils range from fine sandy loams derived from glacial till deposits in the uplands to silty loams derived from glacial lacustrine deposits in the

lowlands. Erosional watershed processes include bank erosion, agricultural runoff, unpaved road erosion, urban stormwater, and hillslope erosion. The Mad River main stem has been subjected to channel management activities (e.g. straightening, dredging, and armoring) as recent as the mid-20th century contributing to present-day decreased access to flood plains and increased erosion hazards.[35] Mean annual precipitation in the watershed ranges from 43 mm along the valley floor to ~60 mm along the upper watershed slopes.[36]

Data Collection

We selected six study sites for monitoring, one along the main stem and one on each of five tributaries (Figure 4.S1). The Mad River is a fifth-order stream and the monitored tributaries are all fourth-order except for High Bridge Brook, which is a third-order stream. In-situ digital turbidity sensors (Forest Technology Systems model DTS-12) and automated samplers (Teledyne Isco 6712) with stage sensors collected river level and water quality data for analyzing event sediment dynamics. Both turbidity and discharge data were collected at 15-min intervals. Suspended sediment concentration (SSC) samples were collected over storm events for laboratory analysis. Samples were processed using the standard gravimetric method (EPA Method 160.2) for measuring total suspended solids; but we discuss using the term SSC for generalizability. To estimate a continuous (15-min interval) SSC time series, relationships between turbidity and SSC were developed for each monitoring station using a rating curve (Figure 4.S2). At all sites, turbidity was highly correlated with SSC (Table 4.S2). Tipping bucket rain gauges (Onset HOBO) collected precipitation data at seven locations within the watershed (Figure 4.S1). A meteorological station, located in the central portion of the watershed, provided soil moisture measured as volumetric water content of the soil at multiple depths.

Stage-discharge relations were obtained from the existing USGS (Mad River - Station ID 04288000) rating curve or rating curves developed from discharge measured on three of the tributaries (Mill, Folsom, and Shepard Brook). At the High Bridge Brook and Freeman Brook monitoring sites, discharge was estimated by measuring stage and using an approximated stage-discharge rating curve based on the Mad River gauge discharge, scaled on watershed area. Study sites were instrumented in the spring after ice breakup and maintained until the start of ice formation in December from 2013 through 2015. Freeman Brook and High Bridge Brook had unstable channels making stage-discharge relationships impractical to develop; thus, they were monitored only during the 2013 data collection period.

The identification of onset and end of individual storm events was semi-automated. The onset was set as the first positive rate of change between consecutive 15-min Q measurements; while termination was based on manual identification of an end point based on the graphical sloped line approximation.[37] When multiple, proximate, discharge peaks could be attributed to distinct rain bands, they were divided into separate events.[9] Rainfall for each event was assumed to be equivalent to the nearest rain gauge for each of the five subwatersheds, and calculated using a Thiessen polygon weighted average of all rain gauges for the main stem.

A comprehensive range of hydrological conditions characterized the monitoring period. May to June 2013 was the wettest consecutive two-month period on record and culminated in a large flooding event on July 3, 2013; whereas, late summer 2015 featured very low flows and drought conditions. For both the Mad River and all monitored tributaries, bankfull flow events occurred during the monitoring period. When comparing our three-year monitoring to a flow duration curve generated from the Mad River USGS stream record shows the monitoring period adequately encompassed the variety of flow conditions (Figure 4.S4).

Across all monitoring sites, 145 storm events were captured resulting in a total of 600 unique events (Figure 4.S4 inset). The highest number of events (35) were recorded in July and the fewest (7) in December. It should be noted that stations were not always online in May or December of a given year because of sensor deployment limitations.

SSC-Q Plot Image Processing

For each storm event, SSC-Q plots were generated from the time series. Because the visual hysteresis patterns may be affected by noise in the SSC or Q data, the event time series for SSC and Q were processed using a Savitsky-Golay smoothing filter[38] prior to generating the hysteresis plots (Figure 4.S5). A 3rd-order, 21-step filter for the Mad River site and a 3rd-order, 13-step filter for the tributary sites provided the best reduction of noise, while preserving the peaks and shape. Because smoothing was performed solely to simplify visual classification, it was applied after calculating the storm metrics. The SSC-Q hysteretic loop was shaded to preserve time (i.e., loop direction) (Figure 4.1b). Images used an 8-bit grayscale color palette, where white indicates the onset of the storm, and dark gray the end of the event, plotted on a black background (Figure 4.1c). To reduce computational time during classification, the SSC-Q plots were converted to 28 by 28-pixel resolution; both SSC and Q were normalized on a per-event basis. SSC-Q plot images were used to train and test an automated classifier using a supervised machine learning approach.

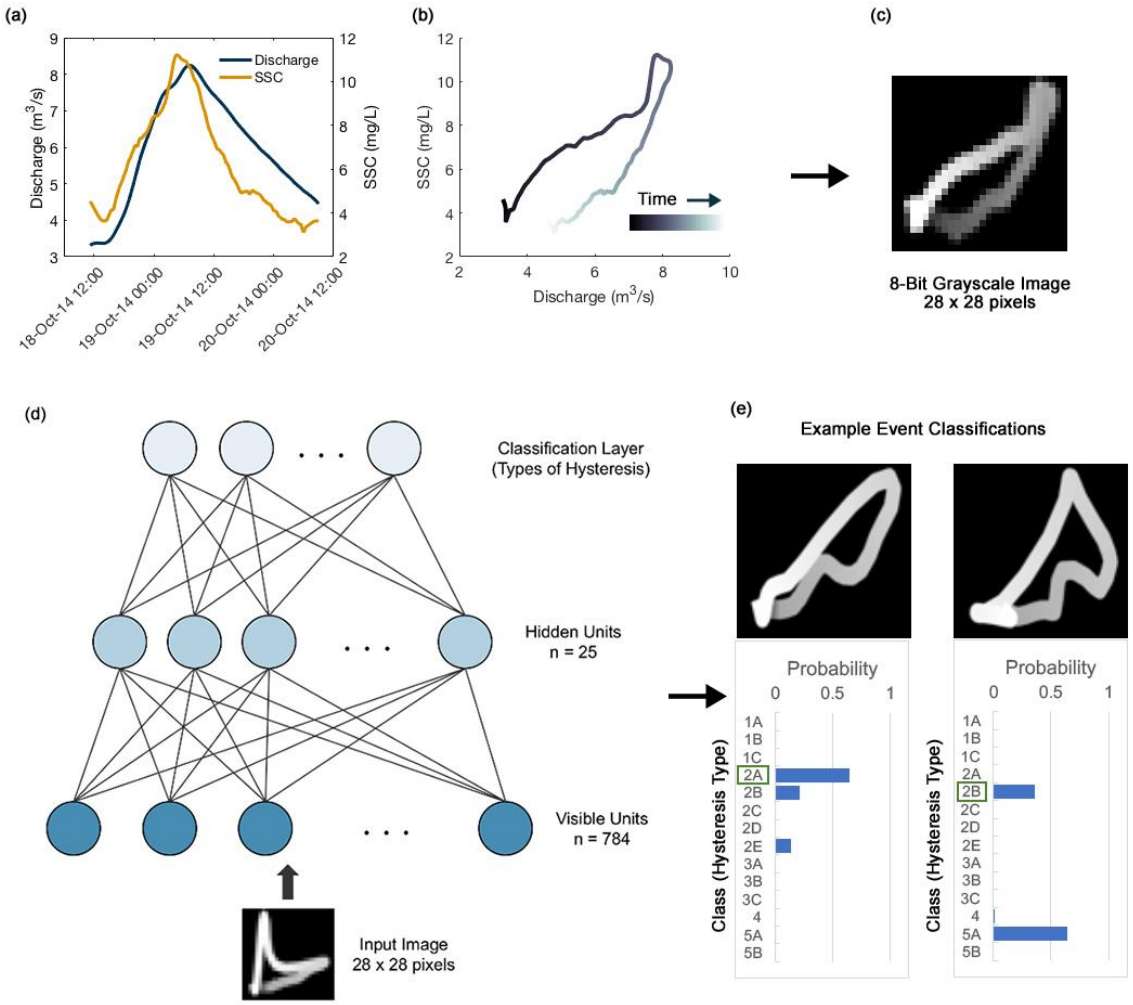


Figure 4.1. (a) hydrograph and sedigraph of October 18, 2014 event at Mad River site; (b) SSC-Q event plot with time represented by shading; (c) 8-bit grayscale image of SSC-Q plot at reduced resolution of 28 x 28 pixels used as input to classification tool where time is represented as going from white to gray; (d) restricted Bolztmann machine (RBM) classifier network; and (e) example of classification output from RBM network for two SSC-Q images where manually labeled correct class indicated by green box.

Automated SSC hysteresis classification of storm events

A set of frequently occurring patterns embedded in the 600 SSC-Q plots was identified; and the SSC-Q plots were manually labeled with the best-matching type of hysteresis. To automate the storm-event classification, a restricted Boltzmann machine (RBM) classifier (Figure 4.1d and Appendix 4.S2) was trained on a portion of the data and tested on the remaining data. Two examples of each hysteretic type were selected for the training data

to ensure each type was represented; the training data were then supplemented with a random selection of events until one-third of the events (210) were included for training. The remaining 361 events were withheld for testing. The "complex" hysteresis types were excluded from the training and testing data.

The RBM classifier performance was compared to the manually assigned hysteresis type of each storm. Because some hysteresis patterns may be a transition between two types, we also evaluated the network using "correct" classification or its next most similar type. The number of hidden nodes, mini-batches, and amount of training data all influence the prediction results; therefore, we varied the number of each as part of evaluating the RBM performance. The results were averaged over 25 model runs, which comprised five network simulations each using five different realizations of training data.

Analysis of storm event variables

We analyzed a suite of additional hydrological and meteorological variables (see Figure 4.5) on a per event basis in conjunction with the raw sedigraph and hydrograph data to determine whether particular conditions are more conducive to producing a given hysteresis type. To determine if differences in the typical storm event were more likely associated with a particular hysteresis type, we compared the variable mean between each hysteresis type and all other types, repeating for each variable. Means were compared using between group t-tests on normally-distributed variables, and Wilcoxon rank tests when normality could not be assumed. In addition, a HI was calculated following procedures in Lloyd *et al.*[25]

Two additional event variables were calculated for each event at the Mad River site. The coefficient of variation (CV) of the total event rainfall from all rain gauges within the watershed was computed to assess the spatial variability of rainfall. Secondly, the total stormflows from all events were fit to a lognormal distribution, repeated for each site. This

distribution was then used to estimate the quantile of stormflow for each event providing a measure of the size of hydrological response associated with each storm event.

Results

Hysteresis in SSC-Q event relationships

Events fell into fourteen SSC-Q hysteresis types that could be grouped into five main categories (Class I – V) corresponding to those originally proposed by Williams[10]; however, Classes II, III, and V are now further subdivided into newly proposed types based on patterns observed repeatedly at the study sites (Figure 4.2). Class I, consists of Types 1A, 1B, and 1C and represents variations on the SSC-Q relationships that do not exhibit any hysteretic behavior. Class II behavior (clockwise loops) consists of types 2A, 2B, 2C, 2D, and 2E with differences in the timing of the peak SSC and peak Q influencing the shape of the hysteresis. Type 2A is indicative of the peak SSC occurring just prior to the peak Q; therefore, resulting in a SSC-Q plot with minor amount of hysteresis. In contrast, when the SSC peak occurs well before peak Q (i.e. Type 2D), it results in an "L" shaped loop. Type 2E is a variation where the peak SSC occurs well before peak Q but has a secondary peak SSC occurring near the peak Q. The Class III SSC-Q relationships (counter-clockwise loops) were similarly subdivided into Types 3A, 3B, and 3C reflecting separation differences in the timing between the SSC and Q peaks. An SSC-Q plot with a linear relationship followed by clockwise loop is indicative of Class IV behavior. The figure-eight shaped SSC-Q loops are represented as Class V with subcategories Type 5A and Type 5B discriminating between the loop direction.

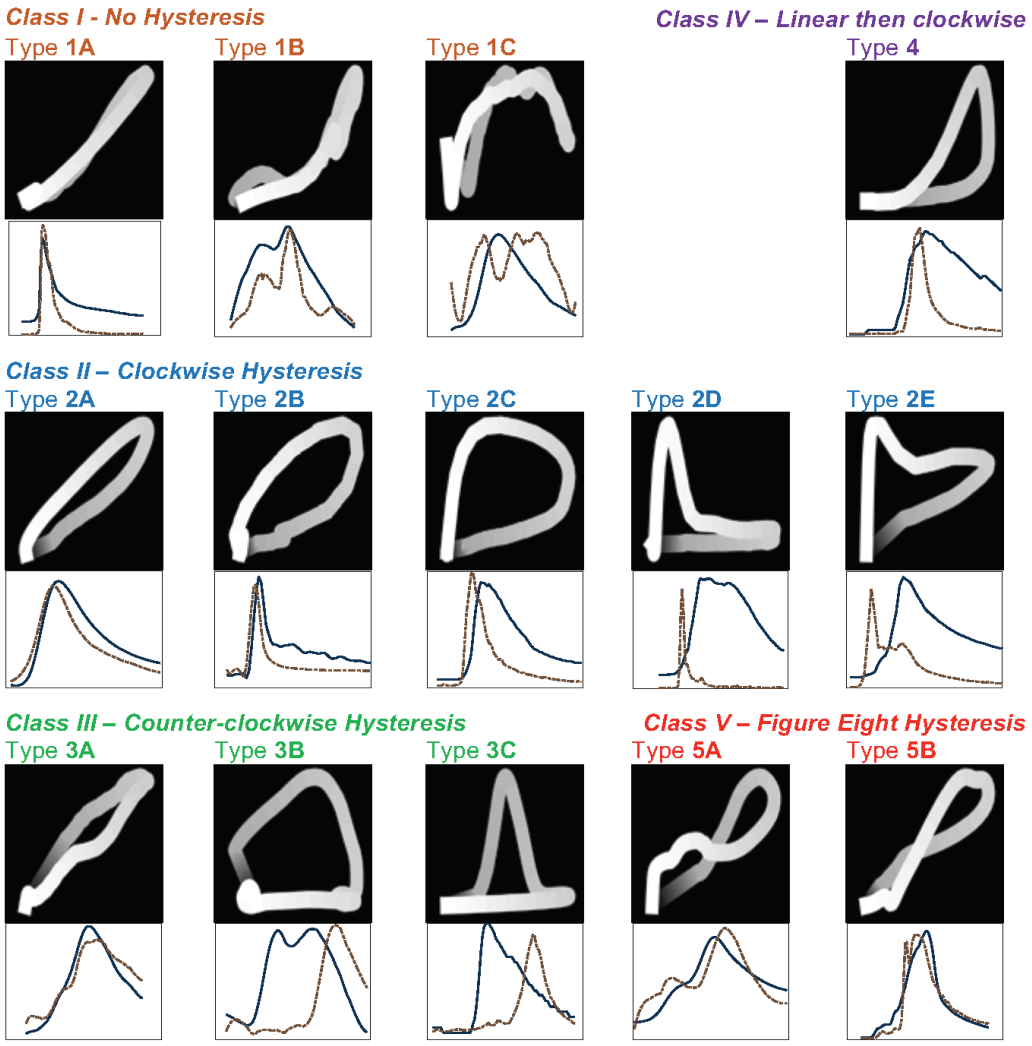


Figure 4.2. Classes of hysteresis in SSC-Q plots from events observed in the Mad River watershed. Solid line indicates hydrograph and dashed line indicates sedigraph. (Note: scales are not consistent between storms)

The majority (90.0%) of the monitored SSC-Q event relationships displayed hysteretic behavior. Across all study sites, the most commonly observed hysteresis types (63.8%, or 388 of 600 events) were Class II (clockwise) patterns, indicative of a peak SSC occurring prior to the peak discharge. Class III (counter-clockwise) types occurred less frequently (8.5%). We were unable to identify a clear hysteresis type in 5.1% of the events and labeled those "complex" type hysteresis.

The frequency of hysteresis types varied among sites (Figure 4.3a). The Mad River (main stem) site featured the most varied hysteresis types including more frequent occurrence of counter-clockwise and figure-eight patterns compared to the tributaries. The tributaries predominantly featured Class II (clockwise) patterns (73%); however, the distribution of Class II subcategories varied among sites. The most common type observed at Freeman, Folsom, and Mill Brook was Type 2B; whereas Type 2C and 2D were the most common pattern at High Bridge Brook and Shepard Brook, respectively. There was variability in median HI by hysteresis type (Figure 4.4). Types 1A, 1B, and 1C had median values of HI near zero as well as the figure eight patterns (Types 5A and 5B). As expected, Class II (clockwise) types had positive median HI and Class II (counterclockwise) had negative. However, with the exception of two types (2C and 3B) median values of HI between types were not statistically different from other types (Figure 4.4).

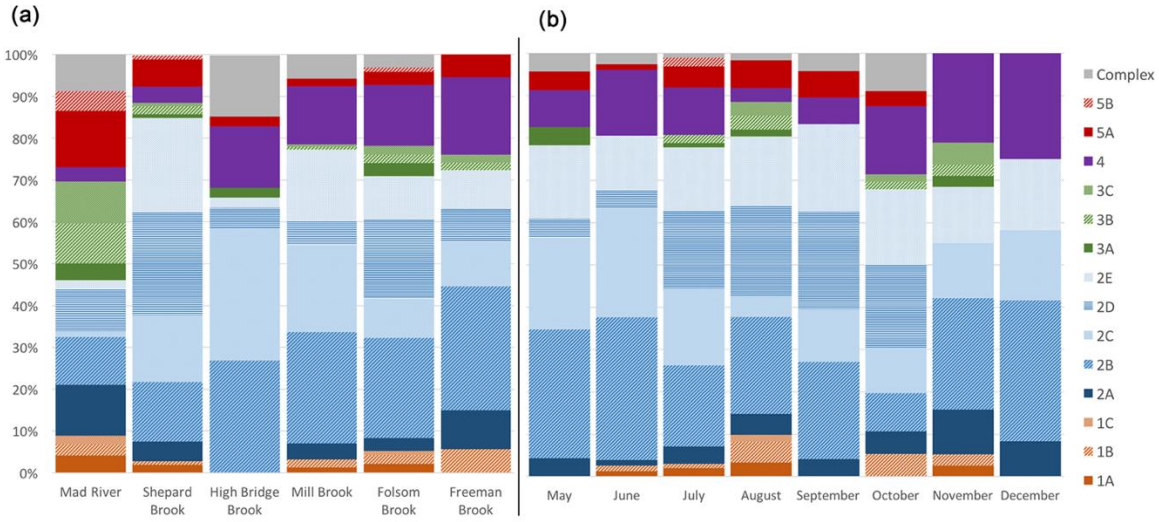


Figure 4.3. (a) Distribution of hysteresis types by site (b) Distribution of hysteresis types by month for only 4th order tributaries (Shepard, Mill, Folsom, and Freeman Brook) combined.

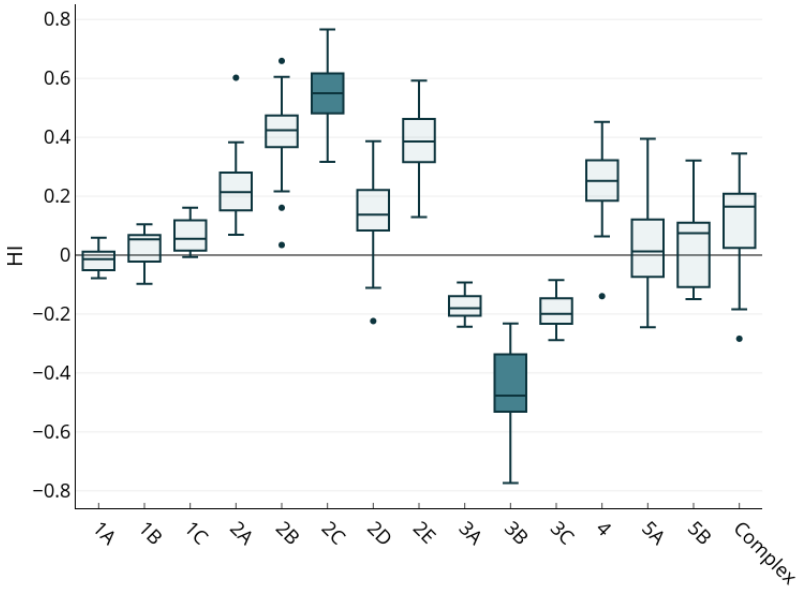


Figure 4.4. Box plots of Hysteresis Index plotted by hysteresis type. Shaded box plots indicate statistical difference from other types.

Automated classification of hysteresis

An RBM configuration with 25 hidden nodes and a mini-batch size of 14 provided sufficiently good prediction. Tests using smaller numbers of hidden nodes and mini-batch size degraded performance; while more hidden nodes resulted in negligible improvement and overfitting. This was consistent with the RBM training guidelines of Hinton[39], which recommend a mini-batch size equal to the number of classes.

The accuracy (total percentage of correct predictions) was 96% for the training data and 52% for the testing data. For the training data, 97% of events were classified correctly or into similar classes, and for the testing data, 71% (Figures 4.S7 and 4.S8). Results of some model runs showed difficulty in discriminating the hysteresis direction (i.e. classified as counterclockwise Type 3B instead of clockwise Type 2B) for some storms indicating that the

learning algorithm places a higher weighting on the image shape rather than the direction of hysteresis (Figure 4.S9).

Storm event characteristics of hysteresis types

Using only a single watershed scale (the fourth-order tributaries) allowed the analysis between typical event characteristics and hysteresis types to be performed at a consistent spatial scale (Figure 4.5). Patterns exhibiting no hysteresis (Types 1A and 1B) or minor hysteresis (Type 2A) occurred relatively infrequently (36 out of 414 events) across the four sites. These events were characterized as having low, 14-day antecedent rainfall, low deep soil moisture, and a lower and shorter event rainfall duration. In addition, despite having little separation between the peaks of the sedigraph and hydrograph, the time between the sedigraph peak and rainfall pulse was longer on average than other events.

Type 2B hysteresis (moderate clockwise loop) was the most commonly observed (96 events). Compared to all events, larger storms on average were classified as Type 2B, featuring greater 3-day and 14-day antecedent rainfall, higher pre-storm baseflows, greater rainfall total and maximum rainfall rate. Hydrological metrics also trended higher with large flood intensity, stormflow, sediment load, and peak sediment concentration. Similarities in Type 2C and Type 2B events were observed in terms of larger flow metrics and antecedent rainfall. However, Type 2C (pronounced clockwise loops) events differed in that they have higher antecedent soil moisture conditions, shorter time span between the rain and sedigraph peaks, and less intense and shorter rainfall than Type 2B. Type 2C events would therefore be characterized by very wet antecedent conditions in which a non-intense rainfall results in a rapid sediment response.

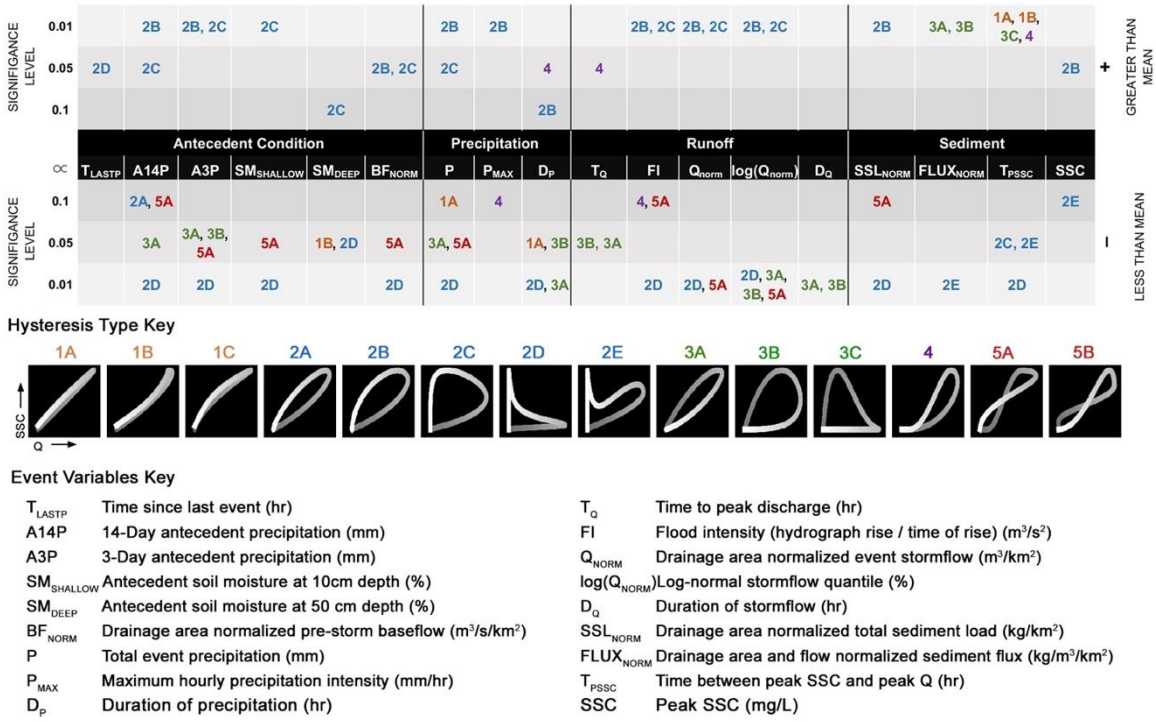


Figure 4.5. Hysteresis types with significant difference within type mean compared to overall mean for storm event variables from all 4th order tributaries (Shepard, Mill, Freeman, and Folsom brook). Normally distributed variables tested with t-test and non-normally distributed variables with the Wilcoxon test.

The Type 2D events had very different characteristics than Types 2B or 2C events. Compared to Types 2B and 2C events, Type 2D events were characterized by drier antecedent conditions with on average lower baseflow, less 3-day and 14-day antecedent rainfall, and lower soil moisture. In addition, Type 2D events tended to feature lower and shorter rainfall and correspondingly less stormflow and less sediment yield. These events were also characterized as having the shortest time span between the rainfall peak and sedigraph peak. Type 2E clockwise events are characterized by both a rapid sediment response and a delayed, second sediment pulse closer to the hydrograph peak. It therefore may be considered a combination of Type 2D and Type 2A or 2B patterns. Perhaps owing to its "mixing" of two event types, Type 2E events are not distinguished statistically from the overall data for the

majority of variables. Of note, Type 2E events feature little separation in the timing of the rainfall peak and sedigraph peak, which is consistent with the very early sediment pulse, and the tendency for low sediment concentrations and low flow normalized sediment flux.

Although Class III, IV, and V patterns occurred less frequently (83 out of 414 events) in the four tributaries, some patterns emerged between the event metrics and hysteresis types classes. The Type 3A and 3B counterclockwise patterns were observed in events that featured on average less antecedent rainfall and lower rainfall amount and duration. The flows were also characterized as being smaller in magnitude with longer durations and time to peak discharge. Interestingly, they also tended to have higher flow normalized sediment fluxes, which is likely a result of the sediment peak occurring during a portion of the hydrograph with higher flows compared to Type 2D. Type 4 hysteresis, which features a more linear SSC-Q relationship followed by a clockwise loop, occurred most frequently during events with a few unique characteristics. These events tended to have low rainfall rates that were long in duration with corresponding low flood intensity and longer times to peak discharge. These storms also occurred most frequently in late spring and late autumn when more typical climate patterns comprise slower moving, less intense frontal storms in spring and autumn, as opposed to shorter, more intense convective storms typical of the summer. Events displaying Type 5A hysteresis were on average small events that occurred with dry antecedent conditions.

Discussion

Expanded classification of hysteresis types and comparison to HI

Expansion of the hysteresis classes helped identify differences among our sites that would not be possible using an analysis based solely on the five original classes proposed by Williams.[10] For example, three tributaries (Shepard, Folsom, and Freeman Brook) were essentially dominated by the same percentage (65 – 67%) clockwise (Class II) hysteresis

patterns across the three sites. However, classification using this one category obscures the fact that in Shepard Brook the most common Type 2D hysteresis differs from that observed in Folsom and Freeman Brook where Type 2B is the most frequent. In addition, at the High Bridge Brook tributary site a different type of clockwise pattern was most frequent (Type 2C). Thus, our expanded classification helps further distinguish in-storm sediment export regimes that could provide more insight into dominant drivers of suspended sediment export during storm events across different catchments.

While other studies have suggested the need for an additional new class, we believe this study is the first to systematically expand the types of hysteresis observed in the SSC-Q relationship. Asselman[3] differentiated between two types of clockwise loops (*typical* and *pronounced*), which would be most similar to our Types 2B and 2C, respectively. Similarly, a weak clockwise loop was identified by Gao and Josefson[4], which corresponds to Type 2A in our classification. Previous studies have also identified a double clockwise loop.[11,17] As noted by Zuecco *et al.*[18] hydrographs that feature multiple peaks can create complex patterns such as the double clockwise loop; but generally, they can be avoided by separating the peaks into individual events. The limitation posed by the singular clockwise loop classification was also identified by Duvert *et al.*[7], where flash flood type events produced hysteresis patterns that could not be easily classified. Our expanded classifications comprise patterns commonly observed in our study; they also comprise types considered or proposed in the above studies.

Hysteresis indices have been adopted widely as a way to expand the analysis of hysteresis loop in terms of direction and prominence. The non-uniqueness of HI across types as identified by Lloyd *et al.*[25] was evident in our data set. For example, Types 2A and 2D have similar HI values as do Types 3A and 3C. However, both pairs of classes differ significantly in the distance between peak SSC and Q. Various enhancements to hysteresis indices have

been proposed to overcome the issue of non-uniqueness. Lloyd *et al.*[25] proposed also computing a loop area; and Zuecco *et al.*[18] proposed using the minimum and maximum differences between the rising and falling limbs of the hysteresis plot. However, all the hysteresis indices, by design, lose information about the shape of the loop during the compression to an index value. Using the entire SSC-Q hysteretic image as proposed here avoids such loss of potentially valuable information, as illustrated by the following discussion.

Drivers of hysteresis type

We were able to interpret different drivers of hysteresis type (Figure 4.5). Antecedent watershed conditions related to soil moisture levels at the start of events, and prior rainfall amounts had a clear impact on event SSC-Q hysteresis type classification. Previously, wet antecedent conditions in the watershed have been connected to clockwise patterns,[23] and in general, our findings were consistent. However, our results indicate a more nuanced analysis is warranted because different types of clockwise patterns were observed to have different event drivers including antecedent conditions and storm event characteristics.

Moderate clockwise hysteresis (i.e., Type 2B) events were on average larger storm events that occurred after significant rainfall and with higher pre-storm baseflows. Because these events featured higher than average sediment load and peak concentrations, this suggests that large rainfall events occurring on relatively wet antecedent watershed conditions cause widespread connection and mobilization of sediment sources; but with limited supply. This is in line with previous studies that found clockwise patterns were indicative of sediment supply in areas nearby the channel,[23] and ample sediment supply from channel and gully erosion.[17] Bank erosion and gully erosion are present in all four tributary watersheds. Conversely, smaller rainfall events occurring with wet antecedent conditions in the watershed were likely to exhibit stronger hysteresis (Type 2C) than observed with larger events. Therefore, sediment sources

are still quickly mobilized, but are more limited in supply, suggesting less lateral connectivity with sediment sources. Thus, in these systems, small storms with high antecedent landscape moisture can quickly mobilize sediment from available nearby sources, yet have limited supply given the rapid decline in sediment concentration around the hydrograph peak.

Events with dry antecedent conditions likely have limited availability or connection to sediment sources, thereby limiting loading. Interestingly, a varied set of hysteresis patterns (primarily Types 2D, 3A, 3B, and 5A) were associated with various measures of dry antecedent conditions. Type 2D events occurred most frequently; and the very short time from rainfall to peak SSC suggests the sediment supply is limited to nearby, overland sediment sources. Thus, once rainfall ceases, the sediment supply decreases very rapidly; and typically dry antecedent conditions suggest that Type 2D and similar patterns would not have significant overland flow or connection to remote sediment sources. This conclusion is consistent with Asselman's[3] observation that pronounced clockwise hysteresis was attributable to erosion and sediment supply just upstream of the measurement location.

Counter-clockwise (Class III) patterns are typically attributed to sediment sources being more distant from the channel[15] and may be attributed to storm events where erosion is primarily occurring in the far upstream or upslope portions of the watershed. These events feature long time delay between rainfall and sediment response. That these events had similar characteristics (dry antecedent conditions and smaller storms) to Type 2D events where sediment is proximal highlights the spatial complexity likely in the sediment response. In the larger Mad River watershed, rainfall events, particularly those during summer months, are often spatially isolated. Therefore, events occurring under similar conditions but with very different sediment pulse timing, may be the result of spatially isolated areas within the watershed mobilizing sediment. Figure eight Type 5A events also were associated with dry

antecedent conditions. This is similar to findings of Seeger[8] but in contrast to findings reported by Buendia et al.,[23] who observed figure-eight patterns associated with large events and wet antecedent conditions. Figure-eight patterns have also been attributed to a combination of streambank and streambed sources with a delayed sediment contribution from distinct upstream sources;[8,11,40] and therefore, may be considered the result of fairly complex processes with drivers that may be more watershed specific. We also note, that our analysis indicates dry antecedent conditions promote more spatially and temporally heterogeneous hysteresis types.

Seasonal changes in hysteresis types suggest shifting drivers and sources of sediment to streams. We observed Type 2C and 2B events more frequently during the late spring (May/June) and late autumn (November/December) months (Figure 4.3b). In the Northeastern U.S., streambanks, fields and hillslopes have less ground cover and are more susceptible to erosion during these months. Other studies observed a similar seasonal trend with clockwise patterns more prevalent in spring or late autumn/winter conditions.[3,9,16] Type 2D events occurred most frequently during summer and early autumn months (July to October). Presentation of hysteresis types were consistent with typical climate patterns observed in the Northeastern U.S. (i.e., more frequent rainfall and wetter antecedent conditions in June/July and drier conditions in August/September followed by increased rainfall and soil moisture again in October/November). During late autumn conditions, Type 4 events were more prevalent. This type of hysteresis has not been studied as extensively as the more common clockwise, counterclockwise, and figure-eight classes; but Williams[10] alluded to a threshold effect at higher discharges. This may suggest sediment sources for Type 4 events are primarily in-channel or bank sources activated by some discharge threshold. Since this event is more common during late autumn and lacks the early sediment pulse present in

Type 2D and 2E events, it also suggests possible occurrence of seasonal depletion of proximal sediment sources.

Effect of watershed scale

The timing of sediment delivery from sources within the watershed to the outlet affects the type of hysteresis observed; and therefore, watershed scale needs to be considered. We observed a scale effect in our study that manifest in a significantly different distribution of hysteresis patterns observed at the Mad River outlet (watershed scale) compared to the tributary scale. When comparing the distributions of hysteresis types at Mill, Shepard, and Folsom Brook to the distribution at the Mad River outlet, we observed a notable shift from predominantly clockwise (Class II) hysteretic types to a more uniform distribution (Figure S10). Asselman[3] noted a similar shift from clockwise loops in upstream tributaries to counter-clockwise loops in downstream locations. Analysis also suggests that the river flood wave can propagate downstream faster than the sediment pulse and that tributaries of this size often produce clockwise SSC-Q loops, both consistent with previous work.[10,16,17,23].

The effect of upstream sediment source delivery on downstream hysteresis type was also observed in our rain gauge data. The Mad River subwatersheds are on the spatial scale of 10 – 50 km^2; as a result, they are more uniformly impacted by rainfall events than the entire Mad River watershed, which frequently experiences rainfall limited to portions of the watershed. Events with large separation between sedigraph and hydrograph peaks (Types 2D, 3B, 3C) tend to have both smaller total stormflow and higher variability among rain gauge totals (Figures 4.S12 and 4.S13). This suggests that at the Mad River scale, hysteresis types with the sediment pulse well before or after the hydrograph peak are indicative of rainfall localized in only a portion of the watershed. These findings are consistent with those of Smith and Dragovich[17] who observed small events were dominated by flow and sediment delivery

from a localized portion of the upstream watershed. Gao[4] noted that hysteresis analysis is most often performed for watersheds smaller than 100 km^2; and that application to larger watersheds is largely affected by sediment delivery from upstream tributaries and, therefore, more difficult to isolate the specific source. Our results support this conclusion; however, our simultaneous monitoring of upstream and downstream locations as well as rainfall variability within watersheds allowed more robust analysis at the larger watershed scale.

Implications for watershed management

Hysteresis analysis has been used to infer sediment dynamics worldwide with the goal of reducing soil loss and sediment export.[5,9,19] Our study offers an expanded classification scheme of hysteresis patterns that leverages high-frequency sediment data to better understand sediment dynamics within a watershed. With SSC-Q hysteresis being monitored in watersheds around the world, we believe the pattern types observed in our study are likely transferable to other locations. However, we do not presume this classification as an exhaustive set of patterns.

Hysteresis analysis relies solely on the timing and shape of the sedigraph and hydrograph, relatively low-cost monitoring equipment such as stage and turbidity sensors may be used without necessarily creating rating curves for discharge and SSC. Aich *et al.*[22] demonstrated a rapid assessment survey, in which only a small number of storms are monitored, can provide insight into sediment transport processes within a watershed. However, as this study demonstrates, longer term monitoring may uncover important seasonal trends.

We noted differences among the subwatersheds in terms of the most frequently observed hysteresis types, which help provide insights into sediment sources. In Shepard Brook, for example, Type 2D and 2E hysteresis were observed more frequently; and Type 4

were less frequent than in other watersheds suggesting the availability of nearby, rainfall-activated sediment sources and the deficiency of distant, upslope sediment sources. Sediment export suppression efforts could, therefore, target sediment sources near the outlet such as gravel road ditches. In Mill Brook and Freeman Brook, Type 4 hysteresis was observed frequently, which is indicative of sediment sources activated above some discharge threshold. Thus, in this watershed, watershed managers might focus on sediment control efforts in areas with soils subject to erosion under higher discharges (e.g., streambanks and gullies), and monitor for shifts in hysteresis patterns to assess changes in sediment production.

Challenges and Opportunities for Expansion

The tools for automating visual pattern recognition from hydrological data presented here represent a novel application of artificial neural networks and an advancement beyond existing time series forecasting[28,29] and clustering applications[41,42]. Our proof-of-concept using an RBM is readily scalable to deep learning algorithms that offer the potential for analyzing even larger data sets. The RBM-based classification of SSC-Q relationships also highlights opportunities for expansion of the methodology beyond analysis of event sediment dynamics. In this study, we combined high frequency monitoring and detailed pattern recognition to identify differences in SSC-Q event relationships that would be obscured if using only the overall hysteretic behavior (e.g. clockwise vs. counterclockwise behavior) of events. Concentration-discharge relationships using other constituents have found hysteresis patterns not observed in SSC-Q relationships.[18,43] Therefore, potential for broader application of this approach to the analysis of event dynamics of other solutes or constituents exists provided high frequency monitoring data are available.

A challenge encountered in our study was the identification of distinct hysteresis types used in the classification scheme. A possible solution lies with the nature of the RBM classifier

model. Because the RBM is a probabilistic network, the probability of correctly classifying each input pattern is computed (i.e. the marginal distribution across classes of hysteresis) along with the suggested classification (Figure 4.1e). If a group of patterns are routinely unable to be classified with a significant level of confidence, it may suggest that additional hysteresis types exist and that the RBM should be re-trained on a larger set of hysteresis types. Complex hysteresis patterns that arise from multiple peak hydrographs (Figure 4.S11a) present an additional challenge for visual classification and highlights the need for careful consideration of event separation techniques. To minimize the effect of multi-peak hydrographs, we followed an approach similar to Sheriff et al.[9] and accordingly, whenever possible, split back-to-back hydrograph rises into individual rainfall events.

As more sediment data become available, particularly from other watershed locations, classification performance should improve through training RBM models on greater numbers of events. We observed differences in the distribution of hysteresis types between relatively similar watersheds. Therefore, having a greater number of events from a variety of watersheds with different land use, climate, geology, topography, and drainage area offer an opportunity for building a greater understanding of drivers of sediment loading during storms across both time and space. Building this knowledge and meaning behind an expanded set of hysteresis types in SSC-Q relationships offers a practical tool for applications focused on sediment connectivity and sources in watersheds. Furthermore, as tools for interpreting big data evolve, environmental researchers should continue their application in interpreting large, high-frequency datasets to better understand the complex environmental processes and their drivers.

Acknowledgements

Support provided by Vermont EPSCoR with funds from the National Science Foundation (NSF) Grants EPS-1101317 and EPS-1556770. Additional support provided by NSF under Grant DGE-0925179NSF, the Vermont Water Resources and Lake Studies Center, and the Robert & Patricia Switzer Foundation. We thank Kristen Underwood for fieldwork contributions and acknowledge landowners of study sites for their support.

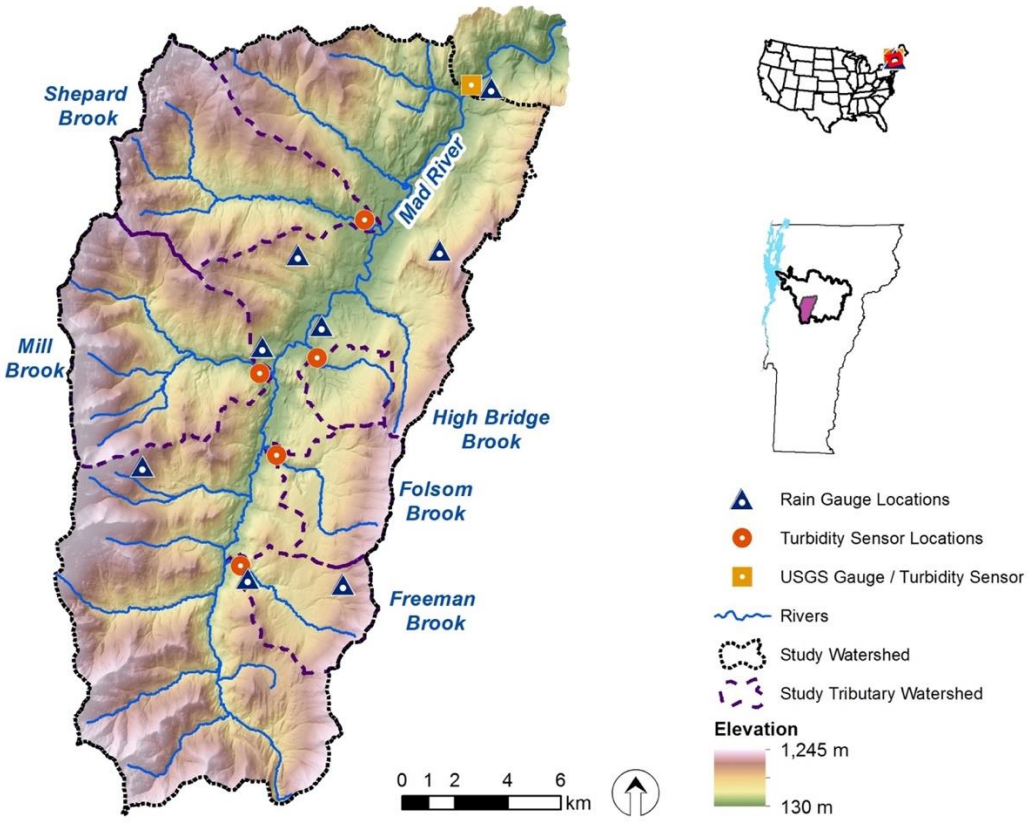


Figure 4.S1. The Mad River watershed and monitoring site locations

Table 4.S1. Key characteristics of study watersheds

	Shepard Brook	High Bridge Brook	Mill Brook	Folsom Brook	Freeman Brook	Mad River
Area (km²)	44.6	8.6	49.8	18.2	17.0	344.0
Minimum Elevation (m)	195	225	216	229	266	140
Maximum Elevation (m)	1117	796	1114	886	860	1245
Elevation Range (m)	923	571	898	657	594	1105
Stream Order	4th	3rd	4th	4th	4th	5th
Drainage Density (km/km²)	2.38	2.45	2.16	1.77	1.95	0.97
% Forested Land	92.2	66.7	89.2	77.6	76.2	85.5
% Agricultural Land	1.0	16.6	1.5	12.7	8.3	4.7
% Developed Land	5.6	15.5	7.0	8.8	14.6	8.0
% Other Land	1.1	2.1	0.8	0.7	1.7	1.1
Road density (km/km²)	0.83	2.30	1.19	1.07	1.80	1.32

Appendix 4.S1 Development of TSS-Turbidity relationships

Total suspended solids (*TSS*) samples and companion measurements of turbidity (*Turb*) were used to generate regression models (rating curves) capable of estimating TSS from turbidity. Models were fit using ordinary linear least square regression of the log transformed variables. Models are of the form

$$\log_{10}(TSS) = b_0 + b_1 \log_{10}(Turb) \ ,$$

where b_1 and b_0 are the slope and intercept, respectively, of the linear regression fit to the log transformed variables. Appropriateness of the model was confirmed by there being no trend in residuals using the regression model. Regression models were converted to normal unit space to give a power curve model of the form: $TSS = 10^{b_0} Turb^{b_1}$ or more generally, $TSS = aTurb^{b_1}$, where $a = 10^{b_0}$. Retransformation of the data introduces bias in the model requiring the application of a bias correction factor (BCF) to the estimates of TSS.[44] The bias correction factor for base-10 logarithmic transformed data is calculated as:

$$BCF = \frac{\sum_{i=1}^{n} 10^{e_i}}{n}$$

where n is the number of samples and e_i is the residual of each measurement in log units. The final model then becomes:

$$TSS = BCF \times aTurb^{b_1}.$$

Rather than using a single TSS – turbidity relationship for all six sites, a TSS – turbidity regression model was fit to each individual site (Figure 4.S2). This was confirmed with an ANCOVA test that identified significant differences in the slopes of the regression models

across all six sites ($p < 0.001$). This is further justified by the recognition that individual watersheds feature differing soil types suggesting that higher TSS prediction accuracy might be provided by TSS-turbidity relationships developed for individual monitoring sites. The individual TSS-turbidity regression models show very good model-fit with adj-R^2 ranging from 0.726 to 0.895 (Table 4.S2).

Table 4.S2. Parameters of fit power law rating curve for monitoring sites.

Site	*a*	b_1	*BCF*	n	R^2 - adjusted
Mad River	2.8222	0.8292	1.4257	163	0.726
Shepard Brook	4.2248	0.7034	1.1688	49	0.752
High Bridge Brook	0.5304	1.1949	1.0532	41	0.895
Mill Brook	1.2179	1.0022	1.0838	83	0.882
Folsom Brook	0.6104	1.1415	1.0717	75	0.888
Freeman Brook	1.2290	0.9670	1.0580	39	0.884

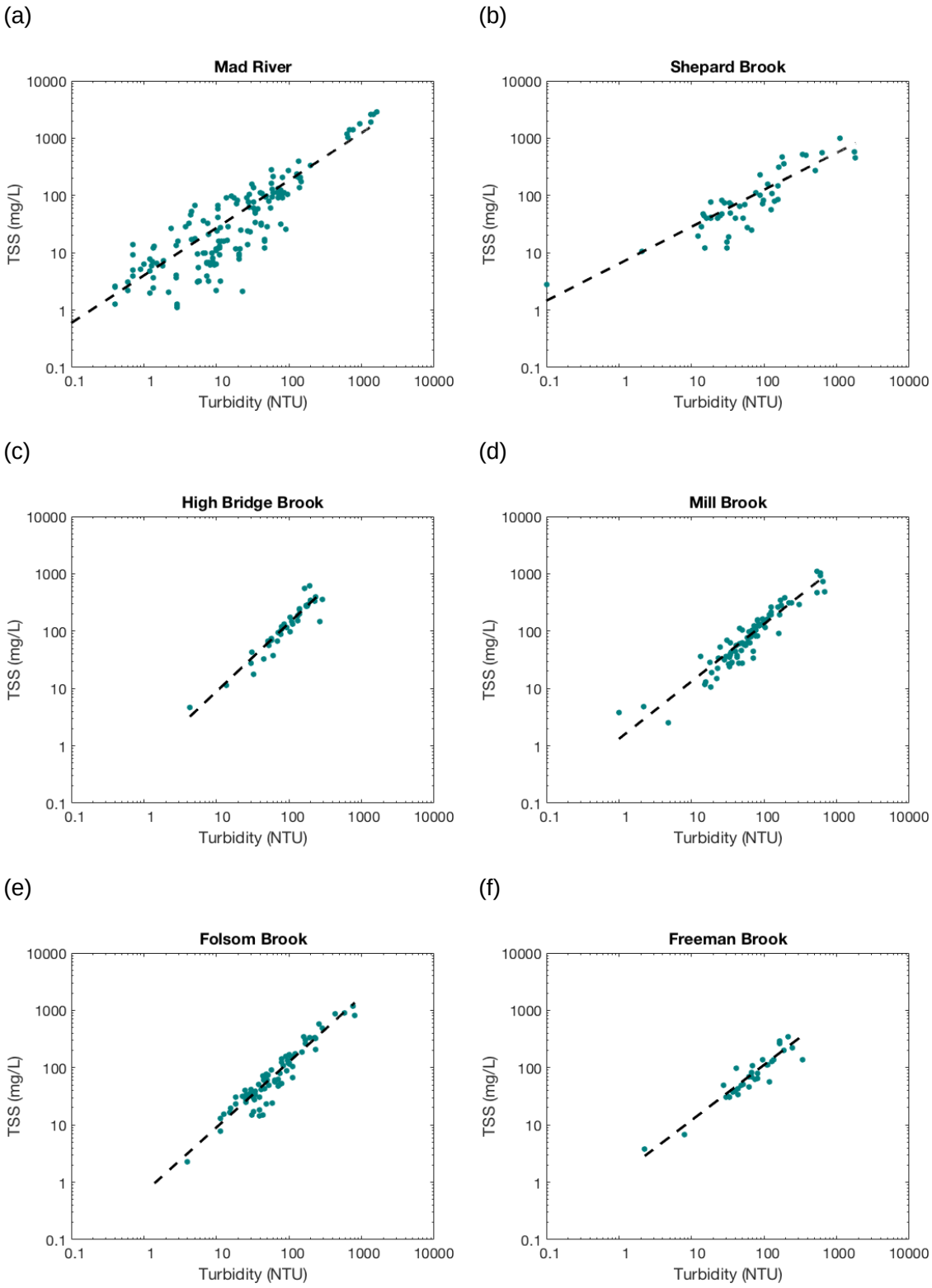


Figure 4.S2. TSS – turbidity rating curves for the six monitoring sites. Dashed line indicates power law fit.

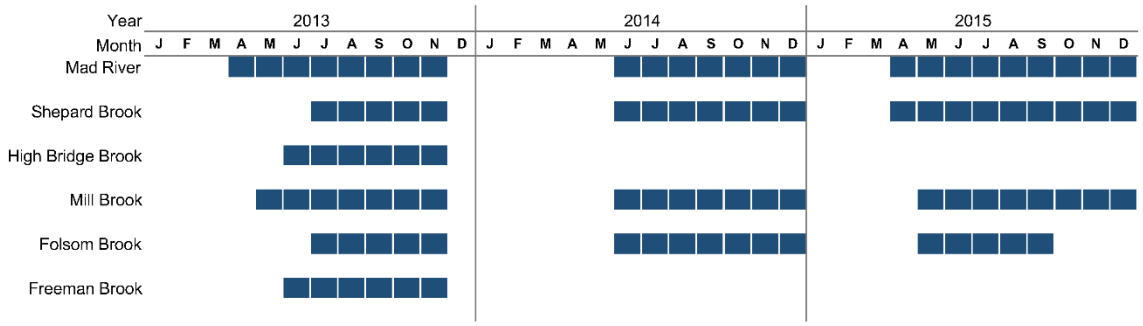


Figure 4.S3. Summary of monitoring record at each study site indicated by months when station was online

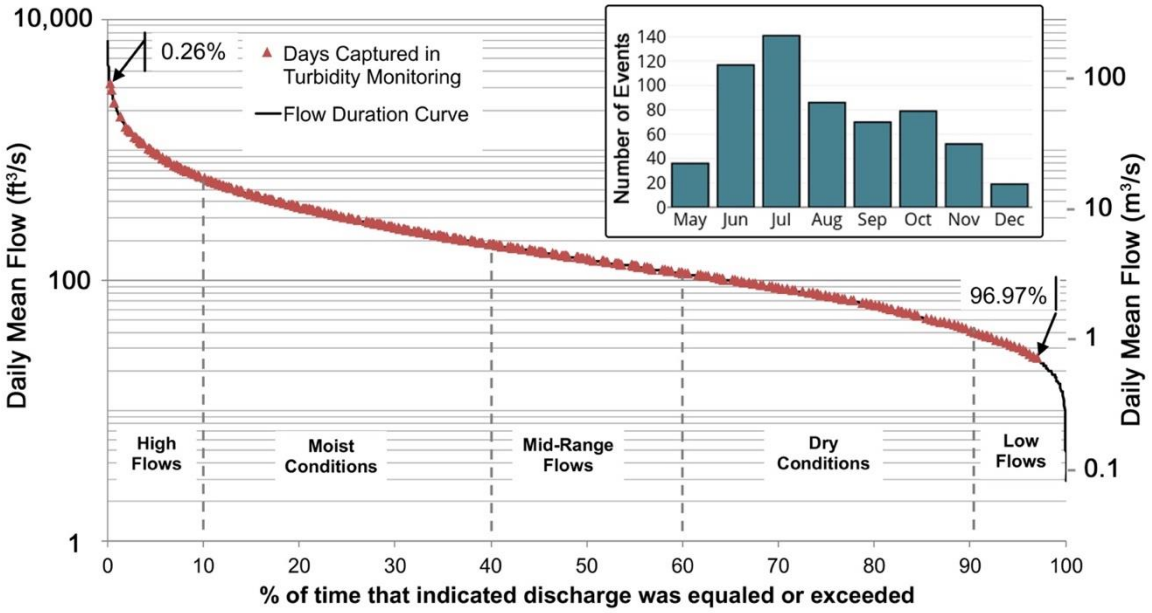


Figure 4.S4. Flow duration curve for the Mad River monitoring site showing days when suspended sediment monitoring occurred. Inset shows count of storm events captured by month across all sites combined.

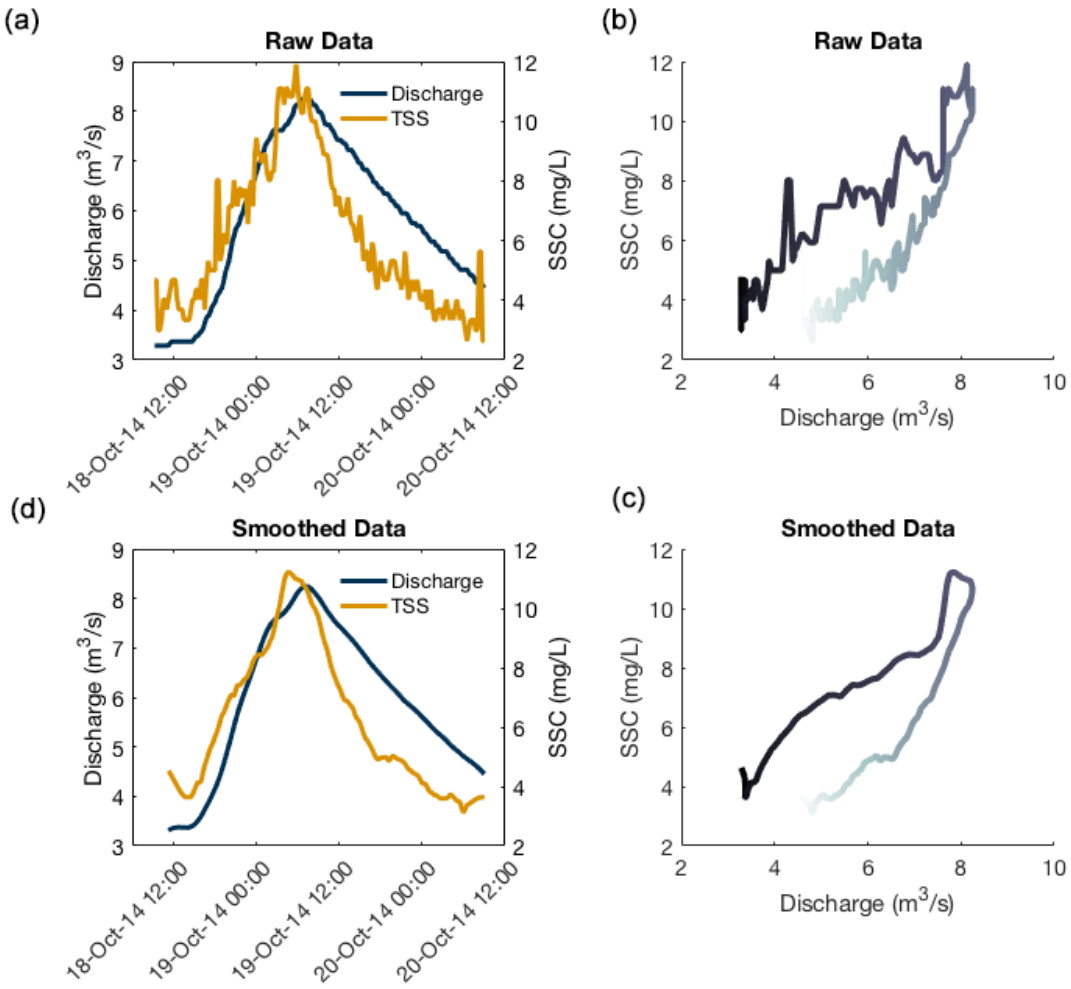


Figure 4.S5 – Example of smoothing and image processing applied to discharge and suspended sediment data from October 18, 2014 storm event at the Mad River. (a) and (c) depict the raw 15-min data and (b) and (d) show data after both time series are smoothed.

Appendix 4.S2 Restricted Boltzmann Machine (RBM)

The restricted Boltzmann machine (RBM)[45], is a type of artificial neural network used for unsupervised learning that has experienced an increase in popularity for its extended use as a classification model[46] and more recent use in deep learning applications.[47] The RBM is a stochastic, generative model that consists of two layers (Figure 4.S6a): a layer of visible nodes (input data) and a layer of hidden nodes. After training, the RBM provides a representation of the training data and its hidden layer may be considered a type of feature extractor.

Learning in a RBM is typically unsupervised using contrastive divergence,[48] which is a form of gradient descent. The goal of training a RBM is to adjust the parameters (weights) of the network such that the probability distribution extracted by the RBM becomes maximally similar to the training data (i.e. maximizing the log-likelihood of generating the observed data). Essentially, there are two phases in the RBM learning algorithm, (1) the *positive phase* where a training pattern is clamped to the visible units and hidden unit activations are computed and (2) the *negative phase* that consists of computing or modeling the reconstruction of the training data given the hidden unit states. Computation of the positive phase is fairly straightforward; but the negative phase requires inference methods using Markov Chain Monte Carlo sampling methods. Different sampling methods have been proposed for the contrastive divergence approach, with block Gibbs sampling over a fixed number of iterations[31] being one of the most common. To improve the learning speed and convergence, the dataset is typically split into mini-batches; and the gradient in each learning step is averaged over the patterns of the mini-batch. Readers are referred to Fischer and Igel[49] and Hinton[39] for in-depth discussion of RBM training methods.

While RBMs are useful in themselves for feature extraction or image generation after initial training; in many applications, they are extended to a classifier model. When extended to a classifier network algorithm, the trained hidden features are used as input to an additional classifier layer (Figure 4.S6b). In this approach, the trained RBM is then connected to a subsequent classifier layer; and the entire network is "fine-tuned" by the backpropagation learning method.[50] RBMs can be used for classification by other methods including training a separate RBM for each class[39] or by incorporating a "class label" node in the visual input layer.[46] We selected the former approach as it has been demonstrated to be both practical and is similar to the approach implemented in DBNN applications [10].

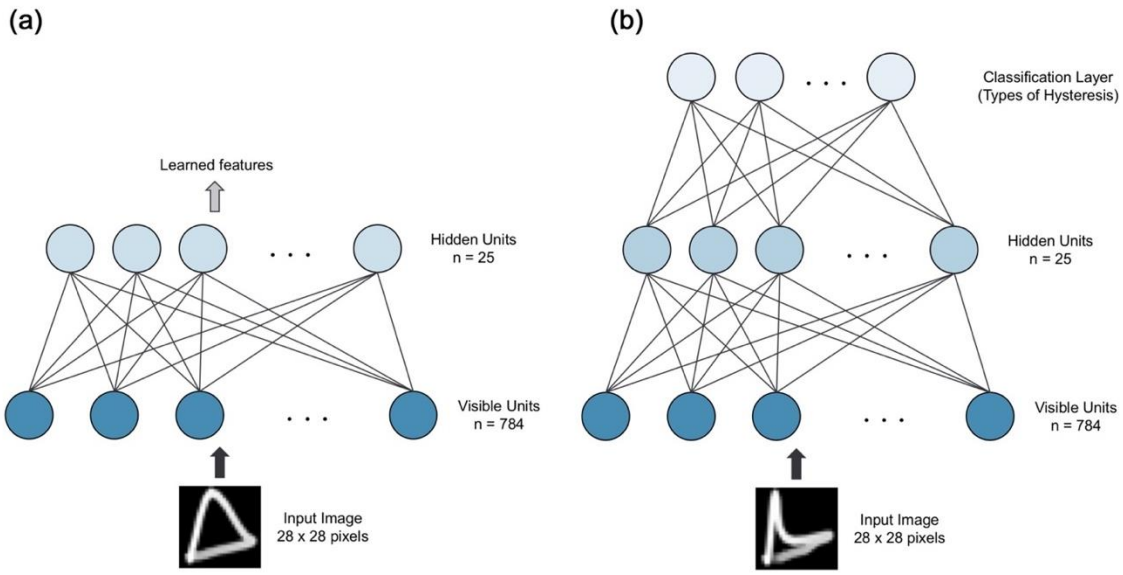


Figure 4.S6. (a) Network architecture of restricted Boltzmann machine (RBM) and (b) RBM classifier neural network.

RBM Model Classification Performance

TRAINING

						Actual									
Predicted	1A	1B	1C	2A	2B	2C	2E	2D	3A	3B	3C	4	5A	5B	
1A	5	0	0	0	0	0	0	0	0	0	0	0	0	0	
1B	0	2	0	0	0	0	0	0	0	0	0	0	0	0	
1C	0	0	2	0	0	0	0	0	0	0	0	0	0	0	
2A	0	0	0	16	0	0	0	0	0	0	0	0	0	0	
2B	0	0	0	0	30	0	0	0	0	0	0	0	1	0	
2C	0	0	0	1	0	29	0	0	0	0	0	0	0	0	
2E	0	0	0	0	0	1	28	1	0	0	0	0	1	0	
2D	0	0	0	0	0	0	1	27	0	0	0	0	0	0	
3A	0	0	0	0	0	0	0	0	3	0	0	0	0	1	
3B	0	0	0	0	0	0	1	0	0	8	0	0	0	0	
3C	0	0	0	0	0	0	0	0	0	0	12	0	1	0	
4	0	0	0	0	0	0	0	0	0	0	0	24	0	0	
5A	0	0	0	0	0	0	0	0	0	0	0	0	14	0	
5B	0	0	0	0	0	0	0	0	0	0	0	0	0	1	overall
% Correct Class	100%	100%	100%	94%	100%	97%	93%	96%	100%	100%	100%	100%	82%	50%	96%
% Correct & Similar Class	100%	100%	100%	94%	100%	97%	97%	100%	100%	100%	100%	100%	82%	50%	97%

Figure 4.S7. Confusion matrix for results of RBM classifier on training data set. Correct classes are indicated in green, shaded cells and what are considered visually similar classes in yellow, hatched cells.

TESTING

						Actual									
Predicted	1A	1B	1C	2A	2B	2C	2E	2D	3A	3B	3C	4	5A	5B	
1A	3	2	0	2	1	1	0	0	0	0	0	0	0	1	
1B	1	2	0	0	0	0	0	0	1	0	0	0	0	0	
1C	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
2A	1	2	0	7	2	0	0	0	2	0	0	1	3	0	
2B	1	0	2	5	59	7	3	0	0	3	0	8	2	1	
2C	0	0	0	0	11	30	8	2	0	0	0	0	0	0	
2E	0	0	0	1	11	6	22	7	0	3	0	0	6	2	
2D	0	0	0	0	1	0	5	31	0	1	2	0	1	0	
3A	0	0	0	1	0	0	0	0	1	0	0	0	0	0	
3B	0	1	0	1	2	3	1	0	1	3	0	3	2	1	
3C	0	0	0	0	2	0	0	6	0	1	5	3	0	1	
4	0	4	0	1	4	1	1	0	2	1	1	21	3	1	
5A	1	0	0	2	0	0	0	0	1	0	0	1	3	0	
5B	0	0	0	0	0	1	0	0	0	0	0	0	0	0	overall
% Correct Class	43%	18%	0%	35%	63%	60%	55%	67%	13%	25%	63%	57%	15%	0%	52%
% Correct & Similar Class	57%	73%	0%	70%	77%	74%	88%	83%	25%	33%	63%	78%	15%	14%	71%

Figure 4.S8. Confusion matrix for results of RBM classifier on test data set. Correct classes are indicated in green, shaded cells and what are considered visually similar classes in yellow, hatched cells.

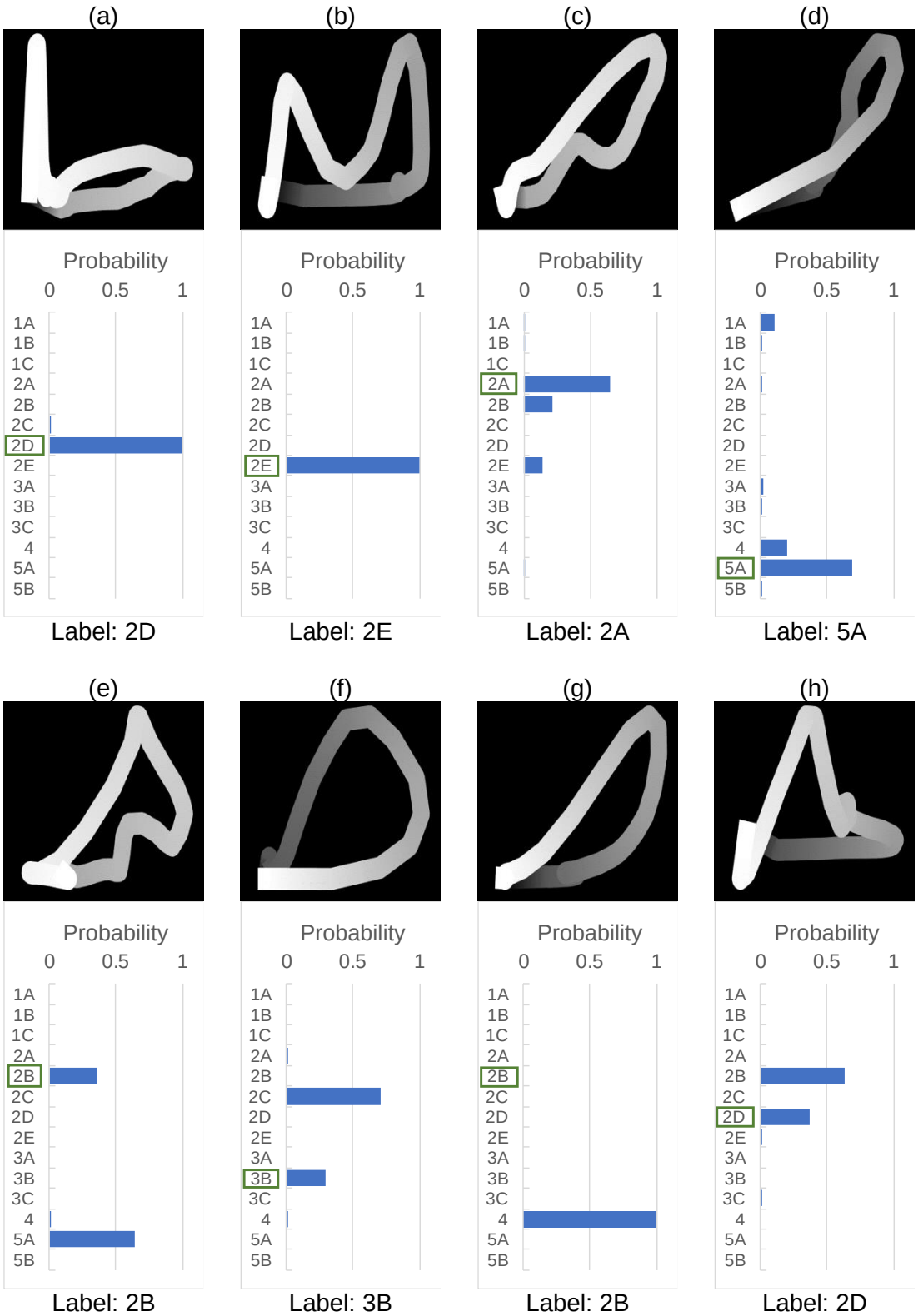


Figure 4.S9. Examples of storm events correctly (a,b,c,d) and incorrectly (e,f,g,h) classified by RBM model in terms of probability of each class being correct. Label applied to each event from manual classification is at bottom and indicated by boxed hysteresis type.

Table **4.S3**. Distribution of observed hysteresis patterns by site with most common type indicated in bold

	Mad River		Shepard Brook		High Bridge Brook		Mill Brook		Folsom Brook		Freeman Brook		All Sites	
Type	**%**	**n**	**%**	**n**	**%**	**n**	**%**	**n**	**%**	**n**	**%**	**n**	**%**	**n**
1A	4.1	6	1.9	2	0.0	0	1.3	2	2.1	2	0.0	0	2.0	12
1B	2.7	4	0.9	1	0.0	0	1.9	3	2.1	2	5.6	3	2.2	13
1C	2.0	3	0.0	0	0.0	0	0.0	0	1.0	1	0.0	0	0.7	4
2A	12.2	18	4.7	5	0.0	0	3.8	6	3.1	3	9.3	5	6.2	37
2B	11.5	16	14.2	15	26.8	11	**26.6**	**42**	**24.0**	**23**	**29.6**	**16**	**20.5**	**123**
2C	1.4	2	16.0	17	**31.7**	**13**	20.9	33	9.4	9	11.1	6	13.3	80
2D	10.1	15	**24.5**	**26**	4.9	2	5.7	9	18.8	18	7.4	4	12.3	74
2E	2.0	3	22.6	24	2.4	1	17.1	27	10.4	10	9.3	5	11.7	70
3A	4.1	6	0.9	1	2.4	1	0.0	0	3.1	3	0.0	0	1.8	11
3B	9.5	13	1.9	2	0.0	0	0.6	1	2.1	2	1.9	1	3.2	19
3C	10.1	15	0.9	1	0.0	0	0.6	1	2.1	2	1.9	1	3.3	20
4	3.4	4	3.8	4	14.6	6	13.9	22	14.6	14	18.5	10	10.0	60
5A	**13.5**	**20**	6.6	7	2.4	1	1.9	3	3.1	3	5.6	3	6.2	37
5B	4.7	7	0.9	1	0.0	0	0.0	0	1.0	1	0.0	0	1.5	9
Complex	8.8	13	0.0	0	14.6	6	5.7	9	3.1	3	0.0	0	5.2	31
		145		**106**		**41**		**158**		**96**		**54**		**600**

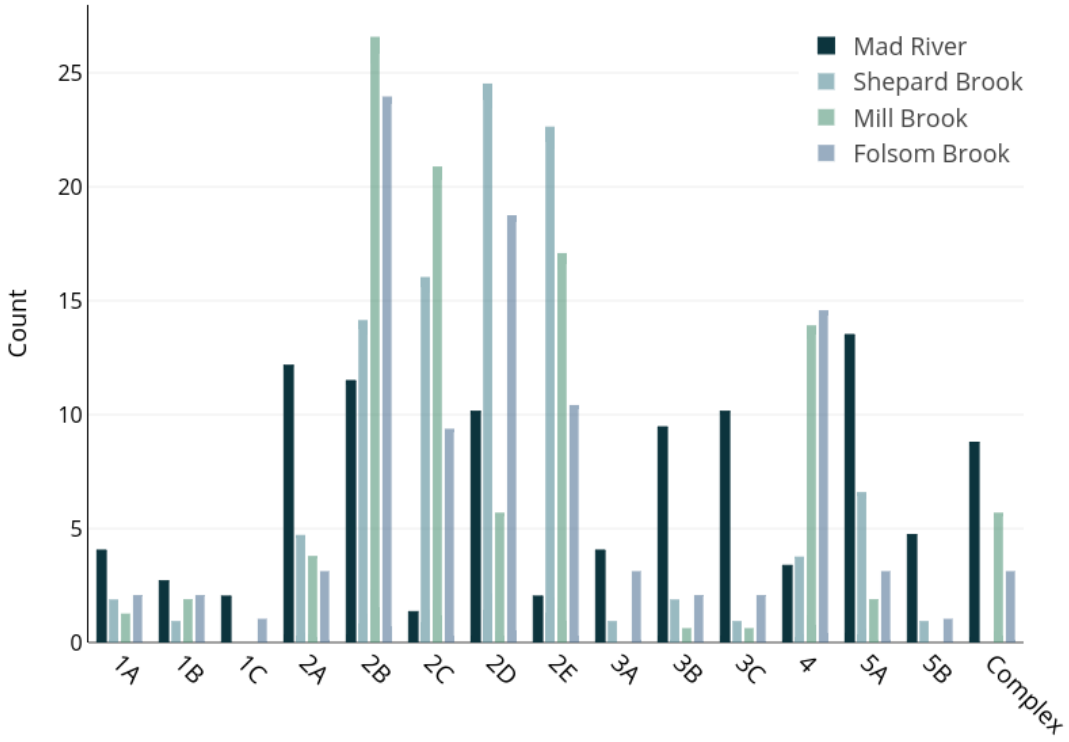


Figure 4.S10. Comparison of observed hysteresis types at Mad River watershed outlet and the Mill Brook tributary scale

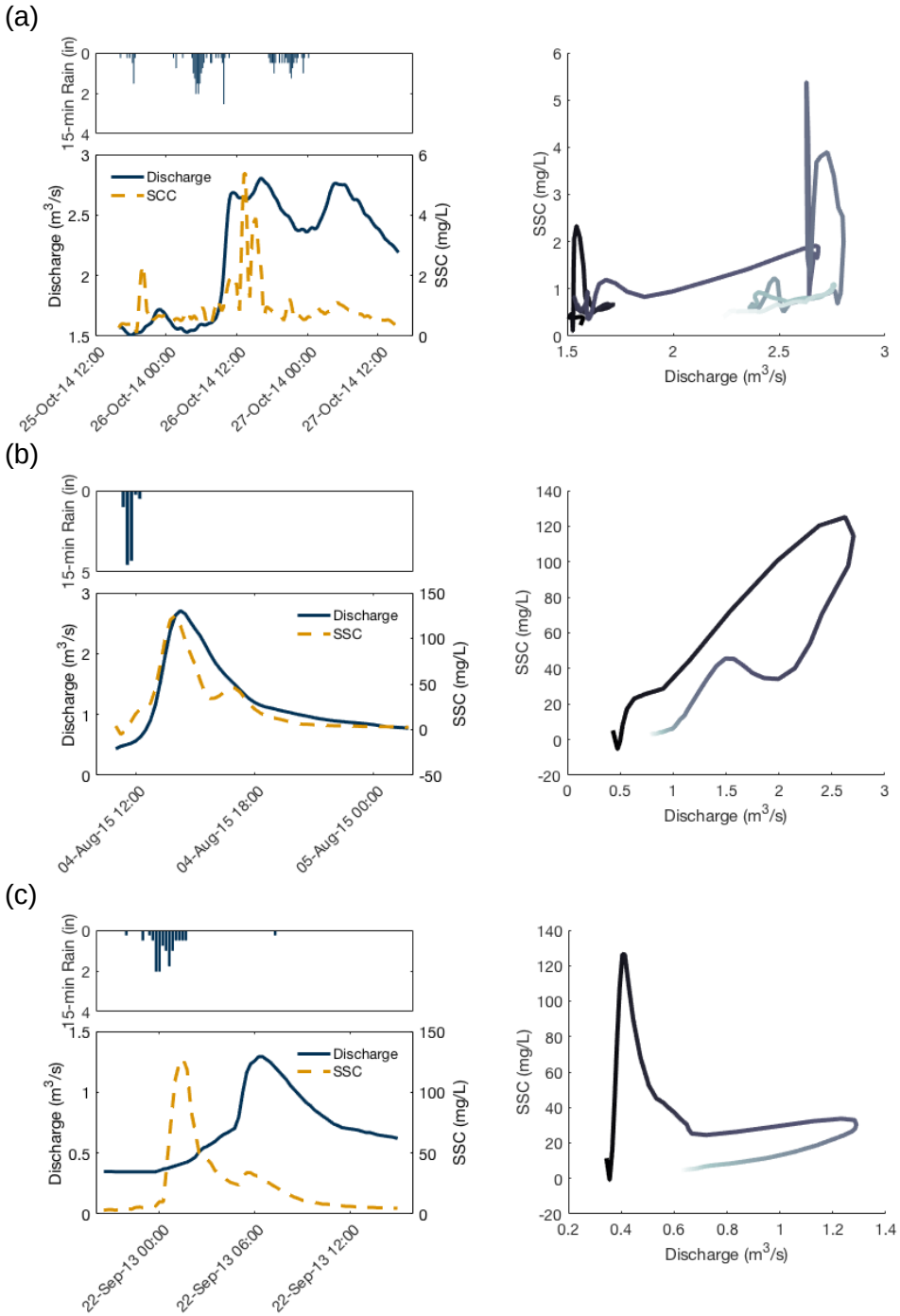


Figure 4.S11. Examples of (a) multipeak event and corresponding SSC-Q plot with a complex pattern of hysteresis, (b) storm event with Type 2A hysteresis showing peak SSC aligned closely with peak Q, and (c) storm event with Type 2D hysteresis showing peak SSC aligned closely with peak rainfall. All events are from Mill Brook.

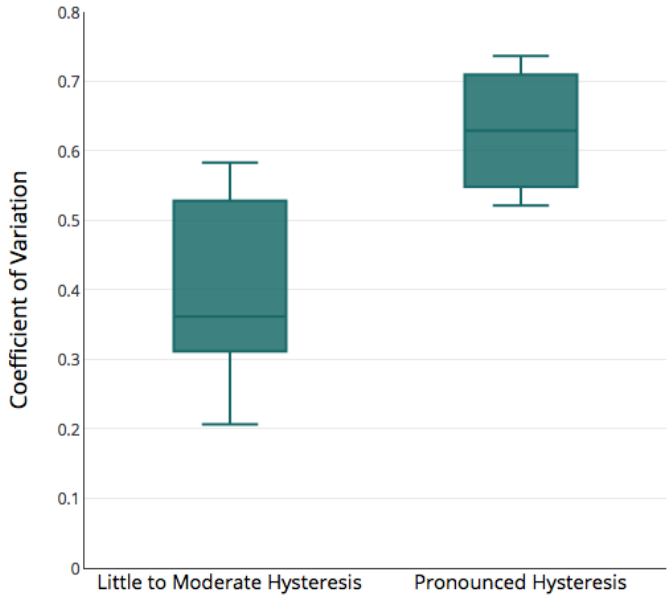


Figure 4.S12. Comparison of coefficient of variation in total event rainfall recorded by rainfall gauges between storms categorized as little to moderate hysteresis (Types 1A, 1B, 1C, 2A, 3A) and those with large separation of rainfall and sediment pulse (Types 2D, 3B, 3C).

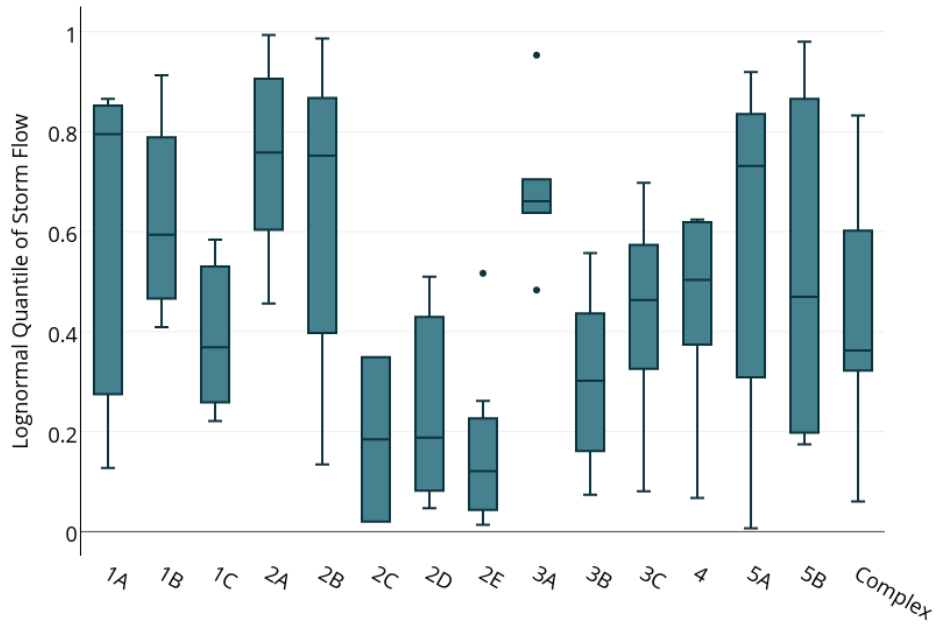


Figure 4.S13. Lognormal distribution quantile for stormflow of Mad River events by hysteresis type

References

(1) Gao, P. Understanding watershed suspended sediment transport. *Prog. Phys. Geogr.* **2008**, *32* (3), 243–263.

(2) Walling, D. E.; Collins, A. L.; Stroud, R. W. Tracing suspended sediment and particulate phosphorus sources in catchments. *J. Hydrol.* **2008**, *350* (3–4), 274–289.

(3) Asselman, N. E. M. Suspended sediment dynamics in a large drainage basin: the River Rhine. *Hydrol. Process.* **1999**, *13* (10), 1437–1450.

(4) Gao, P.; Josefson, M. Event-based suspended sediment dynamics in a central New York watershed. *Geomorphology* **2012**, *139–140*, 425–437.

(5) Lefrançois, J.; Grimaldi, C.; Gascuel-Odoux, C.; Gilliet, N. Suspended sediment and discharge relationships to identify bank degradation as a main sediment source on small agricultural catchments. *Hydrol. Process.* **2007**, *21* (21), 2923–2933.

(6) Onderka, M.; Krein, A.; Wrede, S.; Martínez-Carreras, N.; Hoffmann, L. Dynamics of storm-driven suspended sediments in a headwater catchment described by multivariable modeling. *J. Soils Sediments* **2012**, *12* (4), 620–635.

(7) Duvert, C.; Gratiot, N.; Evrard, O.; Navratil, O.; Némery, J.; Prat, C.; Esteves, M. Drivers of erosion and suspended sediment transport in three headwater catchments of the Mexican Central Highlands. *Geomorphology* **2010**, *123* (3–4), 243–256.

(8) Seeger, M.; Errea, M.-P.; Beguería, S.; Arnáez, J.; Martí, C.; García-Ruiz, J. . Catchment soil moisture and rainfall characteristics as determinant factors for discharge/suspended sediment hysteretic loops in a small headwater catchment in the Spanish pyrenees. *J. Hydrol.* **2004**, *288* (3–4), 299–311.

(9) Sherriff, S. C.; Rowan, J. S.; Fenton, O.; Jordan, P.; Melland, A. R.; Mellander, P.-E.; hUallacháin, D. Ó. Storm Event Suspended Sediment-Discharge Hysteresis and Controls in Agricultural Watersheds: Implications for Watershed Scale Sediment Management. *Environ. Sci. Technol.* **2016**, *50* (4), 1769–1778.

(10) Williams, G. P. Sediment concentration versus water discharge during single hydrologic events in rivers. *J. Hydrol.* **1989**, *111* (1), 89–106.

(11) Megnounif, A.; Terfous, A.; Ouillon, S. A graphical method to study suspended sediment dynamics during flood events in the Wadi Sebdou, NW Algeria (1973–2004). *J. Hydrol.* **2013**, *497*, 24–36.

(12) Soler, M.; Latron, J.; Gallart, F. Relationships between suspended sediment concentrations and discharge in two small research basins in a mountainous Mediterranean area (Vallcebre, Eastern Pyrenees). *Geomorphology* **2008**, *98* (1–2), 143–152.

(13) Pietroń, J.; Jarsjö, J.; Romanchenko, A. O.; Chalov, S. R. Model analyses of the contribution of in-channel processes to sediment concentration hysteresis loops. *J. Hydrol.* **2015**, *527*, 576–589.

(14) Cheraghi, M.; Jomaa, S.; Sander, G. C.; Barry, D. A. Hysteretic sediment fluxes in rainfall-driven soil erosion: Particle size effects. *Water Resour. Res.* **2016**, *52* (11), 8613–8629.

(15) Gellis, A. C. Factors influencing storm-generated suspended-sediment concentrations and loads in four basins of contrasting land use, humid-tropical Puerto Rico. *CATENA* **2013**, *104*, 39–57.

(16) Martin, S.; Conklin, M.; Bales, R. Seasonal Accumulation and Depletion of Local Sediment Stores of Four Headwater Catchments. *Water* **2014**, *6* (7), 2144–2163.

(17) Smith, H. G.; Dragovich, D. Interpreting sediment delivery processes using suspended sediment-discharge hysteresis patterns from nested upland catchments, south-eastern Australia. *Hydrol. Process.* **2009**, *23* (17), 2415–2426.

(18) Zuecco, G.; Penna, D.; Borga, M.; van Meerveld, H. J. A versatile index to characterize hysteresis between hydrological variables at the runoff event timescale. *Hydrol. Process.* **2016**, *30* (9), 1449–1466.

(19) Ramos, T. B.; Gonçalves, M. C.; Branco, M. A.; Brito, D.; Rodrigues, S.; Sánchez-Pérez, J.-M.; Sauvage, S.; Prazeres, Â.; Martins, J. C.; Fernandes, M. L.; et al. Sediment and nutrient dynamics during storm events in the Enxoé temporary river, southern Portugal. *CATENA* **2015**, *127*, 177–190.

(20) Vaughan, M. C. H.; Bowden, W. B.; Shanley, J. B.; Vermilyea, A.; Sleeper, R.; Gold, A. J.; Pradhanang, S. M.; Inamdar, S. P.; Levia, D. F.; Andres, A. S.; et al. High-frequency dissolved organic carbon and nitrate measurements reveal differences in storm hysteresis and loading in relation to land cover and seasonality: high-resolution doc and nitrate dynamics. *Water Resour. Res.* **2017**.

(21) Landers, M. N.; Sturm, T. Hysteresis in suspended sediment to turbidity relations due to changing particle size distributions. *Water Resour. Res.* **2013**, n/a-n/a.

(22) Aich, V.; Zimmermann, A.; Elsenbeer, H. Quantification and interpretation of suspended-sediment discharge hysteresis patterns: How much data do we need? *CATENA* **2014**, *122*, 120–129.

(23) Buendia, C.; Vericat, D.; Batalla, R. J.; Gibbins, C. N. Temporal Dynamics of Sediment Transport and Transient in-Channel Storage in a Highly Erodible Catchment. *Land Degrad. Dev.* **2015**, n/a-n/a.

(24) Yeshaneh, E.; Eder, A.; Blöschl, G. Temporal variation of suspended sediment transport in the Koga catchment, North Western Ethiopia and environmental implications. *Hydrol. Process.* **2014**, *28* (24), 5972–5984.

(25) Lloyd, C. E. M.; Freer, J. E.; Johnes, P. J.; Collins, A. L. Technical Note: Testing an improved index for analysing storm discharge-concentration hysteresis. *Hydrol. Earth Syst. Sci.* **2016**, *20* (2), 625–632.

(26) Langlois, J. L.; Johnson, D. W.; Mehuys, G. R. Suspended sediment dynamics associated with snowmelt runoff in a small mountain stream of Lake Tahoe (Nevada). *Hydrol. Process.* **2005**, *19* (18), 3569–3580.

(27) Lawler, D. M.; Petts, G. E.; Foster, I. D. L.; Harper, S. Turbidity dynamics during spring storm events in an urban headwater river system: The Upper Tame, West Midlands, UK. *Sci. Total Environ.* **2006**, *360* (1–3), 109–126.

(28) Abrahart, R. J.; Anctil, F.; Coulibaly, P.; Dawson, C. W.; Mount, N. J.; See, L. M.; Shamseldin, A. Y.; Solomatine, D. P.; Toth, E.; Wilby, R. L. Two decades of anarchy? Emerging themes and outstanding challenges for neural network river forecasting. *Prog. Phys. Geogr.* **2012**, *36* (4), 480–513.

(29) Maier, H. R.; Jain, A.; Dandy, G. C.; Sudheer, K. P. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environ. Model. Softw.* **2010**, *25* (8), 891–909.

(30) O'Connor, P.; Neil, D.; Liu, S.-C.; Delbruck, T.; Pfeiffer, M. Real-time classification and sensor fusion with a spiking deep belief network. *Front. Neurosci.* **2013**, *7*.

(31) Tieleman, T. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*; ACM, 2008; pp 1064–1071.

(32) Testolin, A.; Stoianov, I.; De Filippo De Grazia, M.; Zorzi, M. Deep Unsupervised Learning on a Desktop PC: A Primer for Cognitive Scientists. *Front. Psychol.* **2013**, *4*.

(33) Stryker, J.; Wemple, B.; Bomblies, A. Modeling sediment mobilization using a distributed hydrological model coupled with a bank stability model. *Water Resour. Res.* **2017**, *53* (3), 2051–2073.

(34) Wemple, B. C.; Clark, G. E.; Ross, D. S.; Rizzo, D. M. Identifying the spatial pattern and importance of hydro-geomorphic drainage impairments on unpaved roads in the northeastern USA. *Earth Surf. Process. Landf.* **2017**, n/a-n/a.

(35) Fitzgerald, E. P.; Godfrey, L. C. *Upper Mad River Corridor Plan*; Friends of the Mad River: Waitsfield, VT, 2008.

(36) PRISM Climate Group. 30-yr Normal Preipitation: Annual, Period: 1981-2010. Oregon State University 2015.

(37) Nathan, R. J.; McMahon, T. A. Evaluation of automated techniques for base flow and recession analyses. *Water Resour. Res.* **1990**, *26* (7), 1465–1473.

(38) Savitzky, A.; Golay, M. J. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.* **1964**, *36* (8), 1627–1639.

(39) Hinton, G. E. A Practical Guide to Training Restricted Boltzmann Machines. In *Neural Networks: Tricks of the Trade*; Montavon, G., Orr, G. B., Müller, K.-R., Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg, 2012; pp 599–619.

(40) Eder, A.; Strauss, P.; Krueger, T.; Quinton, J. N. Comparative calculation of suspended sediment loads with respect to hysteresis effects (in the Petzenkirchen catchment, Austria). *J. Hydrol.* **2010**, *389* (1–2), 168–176.

(41) Pearce, A. R.; Rizzo, D. M.; Watzin, M. C.; Druschel, G. K. Unraveling Associations between Cyanobacteria Blooms and In-Lake Environmental Conditions in Missisquoi Bay, Lake Champlain, USA, Using a Modified Self-Organizing Map. *Environ. Sci. Technol.* **2013**, *47* (24), 14267–14274.

(42) Chea, R.; Grenouillet, G.; Lek, S. Evidence of Water Quality Degradation in Lower Mekong Basin Revealed by Self-Organizing Map. *PLOS ONE* **2016**, *11* (1), e0145527.

(43) Lloyd, C. E. M.; Freer, J. E.; Johnes, P. J.; Collins, A. L. Using hysteresis analysis of high-resolution water quality monitoring data, including uncertainty, to infer controls on nutrient and sediment transfer in catchments. *Sci. Total Environ.* **2016**, *543, Part A*, 388–404.

(44) Rasmussen, P. P.; Gray, J. R.; Glysson, G. D.; Ziegler, A. C. Guidelines and procedures for computing time-series suspended-sediment concentrations and loads from in-stream turbidity-sensor and streamflow data. In *Applications of Hydraulics*; US Geological Survey: Reston, VA, 2011.

(45) Smolensky, P. *Information processing in dynamical systems: Foundations of harmony theory*; University of Colorado at Boulder Department of Computer Science: Boulder, CO, 1986.

(46) Larochelle, H.; Mandel, M.; Pascanu, R.; Bengio, Y. Learning algorithms for the classification restricted boltzmann machine. *J. Mach. Learn. Res.* **2012**, *13* (Mar), 643–669.

(47) Hinton, G. E.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A. r; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process. Mag.* **2012**, *29* (6), 82–97.

(48) Hinton, G. E.; Osindero, S.; Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18* (7), 1527–1554.

(49) Fischer, A.; Igel, C. Training restricted Boltzmann machines: An introduction. *Pattern Recognit.* **2014**, *47* (1), 25–39.

(50) Sarikaya, R.; Hinton, G. E.; Deoras, A. Application of Deep Belief Networks for Natural Language Understanding. *IEEEACM Trans. Audio Speech Lang. Process.* **2014**, *22* (4), 778–784.

CHAPTER 5. PREDICTING RIVER SUSPENDED SEDIMENT LOADINGS USING HYDROMETEOROLOGICAL VARIABLES AND COUNTERPROPAGATION NEURAL NETWORKS

Abstract

The estimation of suspended sediment discharge in river systems not being monitored for sediment or streamflow is essential for watershed monitoring and management needs in order to address the impacts of excessive fine sediment and associated nutrient loading. In this study, we develop and apply two recurrent counterpropagation networks (rCPN) used in tandem; one for predicting streamflow and another for suspended sediment load (SSL) using only hydrometeorological data as the input variables. Three years of suspended sediment concentration and streamflow data from three sites within a medium sized watershed in northeastern United States are used to train and test the rCPN. Predictions of SSL were also compared to those generated by a traditional sediment rating curve method. Results show the rCPN is capable of reliably estimating both streamflow and SSL using only measurements of precipitation and soil moisture as inputs. At all sites, the rCPN outperformed the sediment rating curve model. In addition, the rCPN estimation is demonstrated at an ungauged location by training on data from one watershed and testing in another similar watershed.

Introduction

Fine sediments have been recognized as an important, diffuse source of surface water pollution because of their role in the transport and fate of nutrients and contaminants such as phosphorous (Krueger et al., 2012; Perks et al., 2015; Walling, Collins, & Stroud, 2008). Additionally, suspended sediment and turbidity have been identified as the leading, direct cause of stream and river impairment in the United States (US EPA, 2013). Therefore, it is

desirable to quantify suspended sediment concentration in river systems when assessing the health of river systems, modeling sediment transport, or developing watershed management plans. Unfortunately, the direct measurement of suspended sediment concentration (SSC) frequently and over long periods is resource intensive; and as a result, models are typically utilized for estimating suspended sediment yields from watersheds.

Empirical methods, such as the creation of sediment rating curves (SRCs) using simple linear or power law regression models, have been used extensively for decades to predict sediment concentration from streamflow measurements and continue to be used in current practice. SRCs are most applicable to the estimation of daily average SSC and long-term (e.g. annual) sediment fluxes in medium to large river systems (Horowitz, 2003). When used to estimate SSC at finer time steps (e.g. hourly) or in very small rivers, they often give poor model fit resulting in inaccurate SSC estimates at both peak and/or base flows (Abrahart, See, Heppenstall, & White, 2008; Gao, 2008). Artificial neural networks (ANNs), a subset of the larger field of machine learning methods, are capable of identifying complex, non-linear patterns in large data. In general, ANNs are nonparametric, statistical tools whose structure is inspired by the design of the human nervous system. Because of the inherent complexity in sediment transport processes, there is large potential for ANN use in detailed sediment modeling (Abrahart et al., 2008).

A variety of other sediment transport models include physics-based, conceptual, and empirical models. Lumped and distributed physics-based models can simulate detailed erosion and sediment transport processes and can be advantageous for forecasting conditions outside the range of available observations (Mukundan et al., 2013; Pieri, Poggio, Vignudelli, & Bittelli, 2014; Stryker, Wemple, & Bomblies, 2017). However, physics-based models often require extensive data collection and calibration to simulate the complex physics associated with

erosion and sediment transport; as such, empirical or data-driven methods offer an alternative for sediment yield forecasting (Abrahart et al., 2008; Merritt, Letcher, & Jakeman, 2003). The non-linear relationships between sediment concentration, discharge and other variables, as well as the greater availability of highly-resolved, suspended sediment data have motivated the increased use of data-driven machine learning and genetic programming methods (Mount & Abrahart, 2011). Large amounts of suspended sediment data may now be generated using either automated samplers (Gettel et al., 2011) or surrogate (e.g. turbidity or acoustic backscatter) monitoring (Rasmussen, Gray, Glysson, & Ziegler, 2011; Tananaev & Debolskiy, 2014). The increased availability of reliable, relatively low-cost turbidity sensors, capable of recording at high frequencies (e.g. 5- or 15-minute intervals), offers new opportunities to investigate sediment dynamics at the individual storm event scale using ANN algorithms.

ANNs have found broad application in modeling of hydrology, rainfall-runoff, and water resource variables (Abrahart et al., 2012; ASCE, 2000a, 2000b; Firat, 2008; Maier, Jain, Dandy, & Sudheer, 2010). The most popular neural network is the feed forward back propagation (FFBP) algorithm due to its ability to form a non-linear mapping between input variables and a desired output and the widespread availability of software packages (e.g., Matlab Neural Network toolbox). However, the FFBP has been criticized given its high susceptibility to overfitting, relatively slow learning (training) algorithm, and its sensitivity to initial conditions (Abrahart et al., 2012; Cigizoglu & Alp, 2006; Mount & Abrahart, 2011). Thus, caution is needed to ensure users are knowledgeable of the training process and the algorithm's limitations. Despite these limitations, the positive results identified in many studies result in its continued use in an ever-expanding list of applications, including the prediction of SSC and suspended sediment load (SSL).

Similar to ANN applications for predicting streamflow, the most common ANN used in the prediction of SSC or SSL is the FFBP algorithm, or a variant thereof (Afan, El-shafie, Mohtar, & Yaseen, 2016). Variations on the traditional FFBP algorithm include coupling FFBP with adaptive neuro-fuzzy inference (e.g. Rajaee, Mirbagheri, Zounemat-Kermani, & Nourani, 2009) and wavelets (e.g. Liu, Shi, Fang, Zhu, & Ai, 2013). Other ANN and machine learning methods for estimating SSC or SSL include generalized regression neural networks (GRNN) (e.g. Cigizoglu & Alp, 2006), radial basis function networks (e.g. Alp & Cigizoglu, 2007), support vector machines (e.g. Kakaei Lafdani, Moghaddam Nia, & Ahmadi, 2013), and genetic programming (e.g. Kisi & Shiri, 2012). However, all of these applications used prior observations of suspended sediment as inputs to predict suspended sediment, limiting the model application to sites that have measured suspended sediment data. Even in recent years, the use of time-lagged *measured* suspended sediment data as input data continues to be the most common approach in ANN sediment prediction studies (Buyukyildiz & Kumcu, 2017; Joshi, Kumar, & Adhikari, 2016; Kumar, Pandey, Sharma, & Flügel, 2016; Olyaie, Banejad, Chau, & Melesse, 2015). Abrahart et al. (2008) are one of the few groups to acknowledge that this has little to no practical value in real-world applications. This raises a related concern involving clarity surrounding the intended real-world application of a number of the ANN SSC/SSL prediction studies. In an attempt to identify the optimal set of input variables for predicting SSC/SSL, modelers often make direct comparisons between models that utilize antecedent *measured* sediment data as an input variable to those that use only discharge and rainfall (Buyukyildiz & Kumcu, 2017; Cobaner, Unal, & Kisi, 2009; Joshi et al., 2016; Kisi, 2005; Liu et al., 2013; Shiri & Kişi, 2012) without full consideration that the two models are completely different from an operational standpoint (i.e. the former uses measured sediment data to forecast SSC/SSL at very short time frames into the future (e.g., 1-day or 1-hr) versus

an attempt to estimate SSC/SSL at locations without sediment monitoring). Given the resources needed for direct measurement of high-frequency SSC over long periods of time, we view the former as having very little real-world application.

The ANN studies that predict SSC or SSL using only discharge and meteorological data can be grouped into two broad approaches. The first uses only discharge as input data (Cigizoglu & Alp, 2006; Mustafa, Rezaur, Saiedi, & Isa, 2012; Nourani, Parhizkar, Vousoughi, & Amini, 2013; Zounemat-Kermani, Kişi, Adamowski, & Ramezani-Charmahineh, 2016). This approach acts as an alternative to fitting a SRC and requires measured discharge; but the mapping is not constrained to a linear or power-function relationship. The second approach combines measured discharge and rainfall data as inputs for predicting SSC or SSL (Alp & Cigizoglu, 2007; Heng & Suetsugi, 2013; Kakaei Lafdani et al., 2013; Kisi & Shiri, 2012; Kumar, Pandey, Sharma, & Flügel, 2015; Nourani & Kalantari, 2010; Nourani, Kalantari, & Baghanam, 2012; A. Singh, Imtiyaz, Isaac, & Denis, 2013; Zhu, Lu, & Zhou, 2007), and yields better performance compared to traditional statistical methods such as multiple linear regression (e.g. Cigizoglu & Kisi, 2006).

The development of ANN tools for predicting suspended sediment in catchments that are not gauged, with respect to both discharge and sediment data, is limited. Kamel et al. (2014) predicted SSL in ungauged catchments at the yearly average timescale using physical watershed characteristics and climate data as inputs. Alp and Cigizoglu (2007) evaluated a FFBN and GRNN for daily SSL prediction in a large, humid temperate watershed using only antecedent rainfall, but found the either model insufficient compared to models that included discharge as an input variable. In contrast, Zhu et al. (2007) predicted daily SSL using only rainfall and temperature data for a large watershed with a temperate monsoon climate and found good model performance. Raghouwanshi et al. (2006) also achieved good ANN performance when

predicting SSL in a watershed constrained to monsoon conditions. With the notable exception of these few studies, the vast majority of ANN sediment prediction relies on *measured* streamflow as an input variable, limiting the application to stream-gauged locations.

A more applied approach, which has found success in a few ANN streamflow prediction studies, modifies the ANN architecture to include time-delayed feedback (recurrent connections), where lagged *estimates* of streamflow are added as inputs at each prediction time step. To predict streamflow at ungauged locations, Chang et al. (2002) employed a recurrent FFBP network to predict hourly streamflow; and Besaw et al. (2010) used a recurrent counterpropagation network to predict hourly and daily streamflow. To our knowledge, this approach has not been utilized for sediment prediction, and offers the ability to leverage the temporal autocorrelation in hydrological time series without the need for antecedent *measured* values of suspended sediment.

The frequency with which current peer-reviewed suspended sediment prediction models (1) rely on *measured* streamflow and antecedent sediment data as model inputs, (2) use FFBP networks that overfit the training data, and (3) predict at limited temporal resolution (i.e., daily, monthly, and annual intervals) on large rivers, helped motivate our research. We are specifically interested in applications that target predictions in ungauged river systems. Turbidity data, available at high temporal resolution (hourly) and at multiple locations on small rivers, offered additional opportunity for training and validating ANNs to predict SSC or SSL. In this study, we use two recurrent counterpropagation networks in tandem to predict suspended sediment load using only meteorological data as the input variables. We compare the model results to the traditional SRC and test it for application at ungauged locations by training at one subwatershed and predicting in another with similar watershed properties.

Methods

Study area and data set

The Mad River and two of its tributaries, Shepard Brook and Mill Brook, were selected as the study area (Figure 5.1). Located in the Lake Champlain basin in central Vermont, the Mad River drains a 373 km^2 area in the Green Mountains and flows north into the Winooski River. The watershed area is approximately 83% forested with agricultural (8%) and developed (6%) land uses primarily located along the valley floor. The topography of the Mad River watershed is typical of the Northern Green Mountains area and ranges in elevation from 130 m to 1,245 feet above sea level. The Mad River main stem is a fifth-order river while the Mill Brook and Shepard Brook tributaries are fourth-order streams.

Discharge data were available from the USGS Geological Mad River gauging station (No. 04288000) for the Mad River sediment monitoring site, and by developing stage-discharge relationships for the Mill Brook and Shepard Brook sites. Suspended sediment data were estimated from *in situ* turbidity sensors (Forest Technology Systems [FTS] Model DTS-12) and automatic portable samplers (Isco Model 6712). Water temperature was also recorded by the turbidity sensors. The Waitsfield Weather station, located centrally in the watershed, provided 15-min precipitation measurements and soil moisture (measured as volumetric water content) at two soil depths. Seven additional meteorological stations with tipping bucket rain gauges (HOBO Model RG-2) were setup throughout the watershed for collection of rainfall data (Table 5.1).

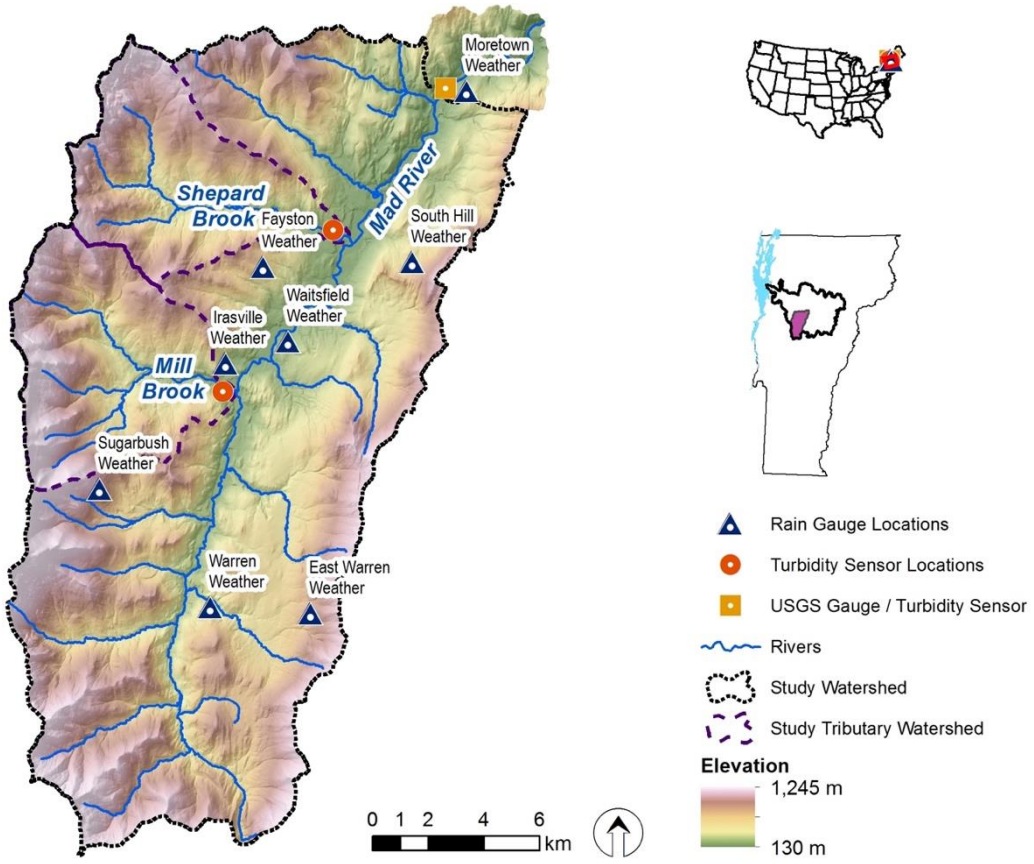


Figure 5.1. Map of study area identifying locations of monitoring stations

The Mad River watershed has a humid continental climate that features about 1200 mm of annual precipitation with distinct seasonal variation. Summer months feature warm temperatures and frequent, fast-moving, convective, rain storms that produce moderate rainfall. Winter can feature heavy snow, particularly on the slopes of the Green Mountains on the western portion of the watershed. Fall and spring typically feature slower moving frontal rain events. The Mad River and its tributaries ice over in the winter months; and spring snowmelt streamflows are typically some of the highest flows during the year. During the 2013-2015 monitoring period, the Mad River experienced an exceptionally wet period during early summer 2013 and moderate drought conditions in later summer 2015.

Table 5.1. Summary of spatial locations and data collected at monitoring stations

Monitoring Site	Latitude	Longitude	Elevation (m)	Data Collected
Meteorological Stations				
Moretown Weather	44.277362	-72.742031	166	Rainfall (mm)
South Hill Weather	44.220720	-72.756119	430	Rainfall (mm)
Fayston Weather	44.218954	-72.823194	448	Rainfall (mm)
Irasville Weather	44.179044	-72.836175	218	Rainfall (mm)
Sugarbush Weather	44.146823	-72.896246	661	Rainfall (mm)
Warren Weather	44.114678	-72.853027	273	Rainfall (mm)
East Warren Weather	44.107008	-72.801123	573	Rainfall (mm)
Waitsfield Weather	44.194960	-72.818398	207	Rainfall (mm) Soil Moisture (% VWC) at 10 cm and 50 cm depths
Sediment Monitoring Stations				
Mad River	44.2768039	-72.7420442	166	Turbidity (NTU)
Shepard Brook	44.2287270	-72.7870840	195	Turbidity (NTU) Stage (m)
Mill Brook	44.1787682	-72.8358413	218	Turbidity (NTU) Stage (m)

Methods

Estimation of suspended sediment

Suspended sediment was estimated for each site using rating curves developed for the *in-situ* turbidity sensors. Stream samples were collected during rain events using the autosamplers and analyzed for total suspended solids (TSS) by the standard gravimetric method (U.S. Environmental Protection Agency, 1999). Simultaneous measurements of turbidity were then log-transformed and fit to a linear regression model to generate the turbidity rating curves. Log-transformation compensated for the presence of heteroscedasticity in the data, but resulted in a bias on retransformation; therefore, TSS was corrected for each site using:

$$TSS = BCF \cdot aTurb^{b_1},$$

where the parameter b_1 is the slope of the linear regression on the log-transformed data; a is the retransformed intercept b_0 ($a = 10^{b_0}$), and the bias correction factors (*BCF*) for base-10 logarithmic transformed data is calculated as:

$$BCF = \frac{\sum_{i=1}^{n} 10^{e_i}}{n} .$$

Here, n is the number of samples; and e_i is the residual of each measurement in log units. The turbidity sensor and autosampler were deployed at a fixed location along the stream bank; and thus, represented a point measurement along the river cross section. Sediment concentration is known to vary along the cross-section and flow-integrated sampling techniques are often necessary (Edwards & Glysson, 1999). However, at our study sites, we believe the high stream-flow velocities that occurred during storm events contributed to well-mixed profiles and the dominance of fine-grained sediment in the suspended load, and as a result, did not warrant using a flow-integrated technique.

Discharge and rainfall data processing

We used data from all eight of the Mad River watershed rain gauges to compute an average basin rainfall using the Thiessen polygon weighted average method. Because some rain gauges were not consistently online over the monitoring period, the averaging was automated using all available online gauge data and a grid method as described by Han and Bray (2006). Average rainfall for the Shepard Brook and Mill Brook subwatersheds used a similar approach; however, given the size of these subwatersheds, the average rainfall essentially approximates the nearest gauge (i.e., Fayston Weather station) for Shepard Brook or nearest two rain gauges (i.e., the Sugarbush and Irasville Weather stations) for Mill Brook.

Stage-discharge relationships for Shepard Brook and Mill Brook were established by modeling the section of river used for monitoring in the HEC-RAS (version 4.1). The rating curves were calibrated using discharge measurements and the velocity area method. See Appendix B for details on the development of the stage-discharge relationships.

Counterpropagation Network

The counterpropagation network (CPN) is an ANN that combines elements of unsupervised and supervised learning. It is purely data-driven and self-adapts to learn nonlinear mappings between predictor inputs and a set of response vectors (desired output classes). Hecht-Nielsen (1988), designed the CPN by combining two ANN algorithms – a Kohonen self-organizing map (hidden layer) and a Grossberg oustar structure (output layer). The architecture most commonly utilizes a configuration referred to as forward-only, in which one is not interested in the inverse relation between input variables and target output classes. In this study, we used the forward mapping CPN configuration of Figure 5.2 to predict river discharge, which was subsequently used as input data to a second CPN for prediction of TSS.

The supervised learning approach of the forward-only CPN algorithm necessitates partitioning the data into training and testing data sets. The portion of data reserved for testing varies widely in ANN studies, but generally ranges from 20% to 50% of the data. We opted to split the data set chronologically using ~60% of the data for training and 40% for testing.

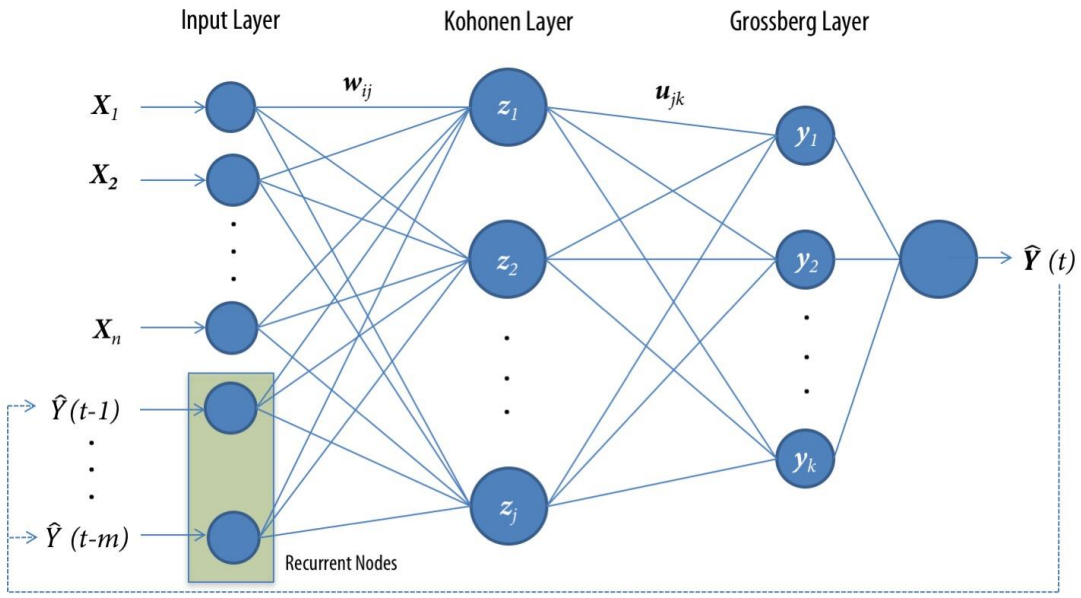


Figure 5.2. Recurrent counterpropagation network (rCPN) architecture with n input variables and m recurrent nodes

The network is fully connected meaning all nodes between the input, the Kohonen, and the Grossberg layers are connected by weight matrices w_{ij} and u_{jk}, respectively. The CPN input layer has *i=1:I* nodes that pass the input values to the hidden layer. The nodes *(j=1:J)* of the hidden layer (or Kohonen layer) cluster the input values based on a similarity metric. The output (i.e., Grossberg) layer has *k=1:K* nodes that help classify the output predicted at the Kohonen layer.

The number (*K*) of output nodes corresponds to the number of target output classes pre-determined by the user. To utilize the CPN for prediction of a continuous variable such as TSS or discharge, we discretized the output (target) variable into 100 classes, logarithmically spaced over the range of the variable. The target output data, either discharge or TSS, were preprocessed by rounding real-valued measurements into the nearest CPN class.

Prior to CPN training and training, all of the input data were normalized to values between 0 and 1 according to:

$$x_{norm_i} = \frac{(x_i - x_{min})}{(x_{max} - x_{min})} ,$$

Inputs were normalized separately over each variable type (e.g. precipitation, soil moisture, discharge).

The forward-only CPN utilizes a hybrid learning approach that features stages of unsupervised and supervised learning. Unsupervised learning is used during the first stage of network training (i.e., within the hidden layer) for the purpose of clustering the input data. An input vector ***x***, comprised of *I* variables, is passed to the hidden layer where a distance metric (in this case, Euclidean distance) is calculated between ***x*** and the weight vector, $\boldsymbol{w_j}$, associated with each Kohonen hidden node. The Kohonen node with the most similar weight vector (i.e., minimum Euclidean distance to the input vector, ***x***) is identified as the *winning* node; and the weight vector is updated according to:

$$\Delta w_{ij} = \begin{cases} \alpha(\boldsymbol{x} - \boldsymbol{w}_j), & \text{for } j = \text{winning node} \\ 0 & \text{otherwise} , \end{cases}$$

where α is a Kohonen learning rate subject to $\alpha > 0$. We used a constant learning rate ($\alpha = 0.7$) in this study. Next, a winner-take-all activation function ensures that the output value, z_j, associated with the winning Kohonen node are set to 1; all other nodes are set to 0. The Kohonen layer output may then be propagated to the Grossberg layer where the network output vector, $\hat{\boldsymbol{y}}$, is computed as:

$$\hat{\boldsymbol{y}} = \sum_{j=1}^{J} \boldsymbol{u}_j z_j \ .$$

The weights $\hat{\boldsymbol{u}}_{jk}$ are then updated during training as follows:

$$\Delta \boldsymbol{u}_j = \beta(\boldsymbol{y} - \hat{\boldsymbol{y}}) ,$$

where β is the Grossberg learning rate and $\boldsymbol{y}$ is the target output vector and $\hat{\boldsymbol{y}}$ is the network output. The Grossberg learning rate β was set to a constant value of 0.1 in this work. The training process is repeated for each of the paired training patterns in the training dataset until the CPN has learned the mapping between input variables and known target classes within some user-defined convergence criterion (in this work, a summed root-mean-square error value of $<10^{-5}$). After learning (i.e., training) is complete, the CPN hidden weights are fixed and the network may be used for prediction. The output vector of the Grossberg layer is processed using a winner takes all method such that the output vector component with the highest value is identified as the predicted class. In this way, the output layer approximates a Bayesian classifier.

One modification made to the original CPN algorithm for this research was the incorporation of a recurrent feedback loop to allow predictions from previous time-steps to be used as predictor (i.e., input) variables. The dotted lines in Figure 5.2 show lagged network estimates (in this case Discharge or TSS) added to the input vector at the current time step to help improve future predictions. We refer to this network as a recurrent counterpropagation network (rCPN). All data processing and coding of the rCPN was performed in MATLAB (v7.13)

Model Evaluation

The rCPN model performance was evaluated using several metrics that compare observed measurements to predicted outputs. One of the most common measures of ANN performance is the root mean square of errors (RMSE) calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i)^2}{N}},$$

where $\boldsymbol{y_i}$ is the observed target vector (for either discharge or TSS), $\widehat{\boldsymbol{y}}_{\boldsymbol{i}}$ is the predicted output vector after training, and N is the number of observations. RMSE ranges from 0, indicating a perfect fit, to $+\infty$, indicating no fit. RMSE is sensitive to extreme values, or the occasional large error.

Another assessment metric is the Nash-Sutcliff model efficiency (NSE) given by:

$$NSE = 1 - \frac{\sum_{i=1}^{N}\left(\hat{Y}_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2} ,$$

where Y_i and $\hat{Y}_i$ are the observed and predicted variables (discharge or TSS); $\bar{Y}_i$ is the mean of the observed variable; and N is the number of patterns. NSE ranges from 1 to $-\infty$ (no fit) and is a measure of the overall goodness of fit of the model. A value of 1 indicates a perfect fit, a value of 0 indicates poor fit and that the model prediction is no better than using the mean of the data, and a value <0 indicates the model performs worse than simply using the mean.

Model Application

TSS-turbidity rating curve development

Sampling suspended sediment during hydrological events provided paired TSS and turbidity sensor data over the period from 2013 – 2015, which enabled the development of TSS-turbidity rating curves. At the Mad River and tributary sites, TSS samples were obtained at turbidity values ranging from about 1 NTU to 1,600 NTU. Linear regression models fit to log-transformed TSS and turbidity values showed good fit at all sites with R^2-adj values ranging from 0.726 to 0.882 (Figure 5.3 and Table 5.2). Because the regression model slopes differed across the sites, we used individual models for each site. Turbidity and TSS relationships are known to be affected by the color and distribution of soil grain size; therefore, finding

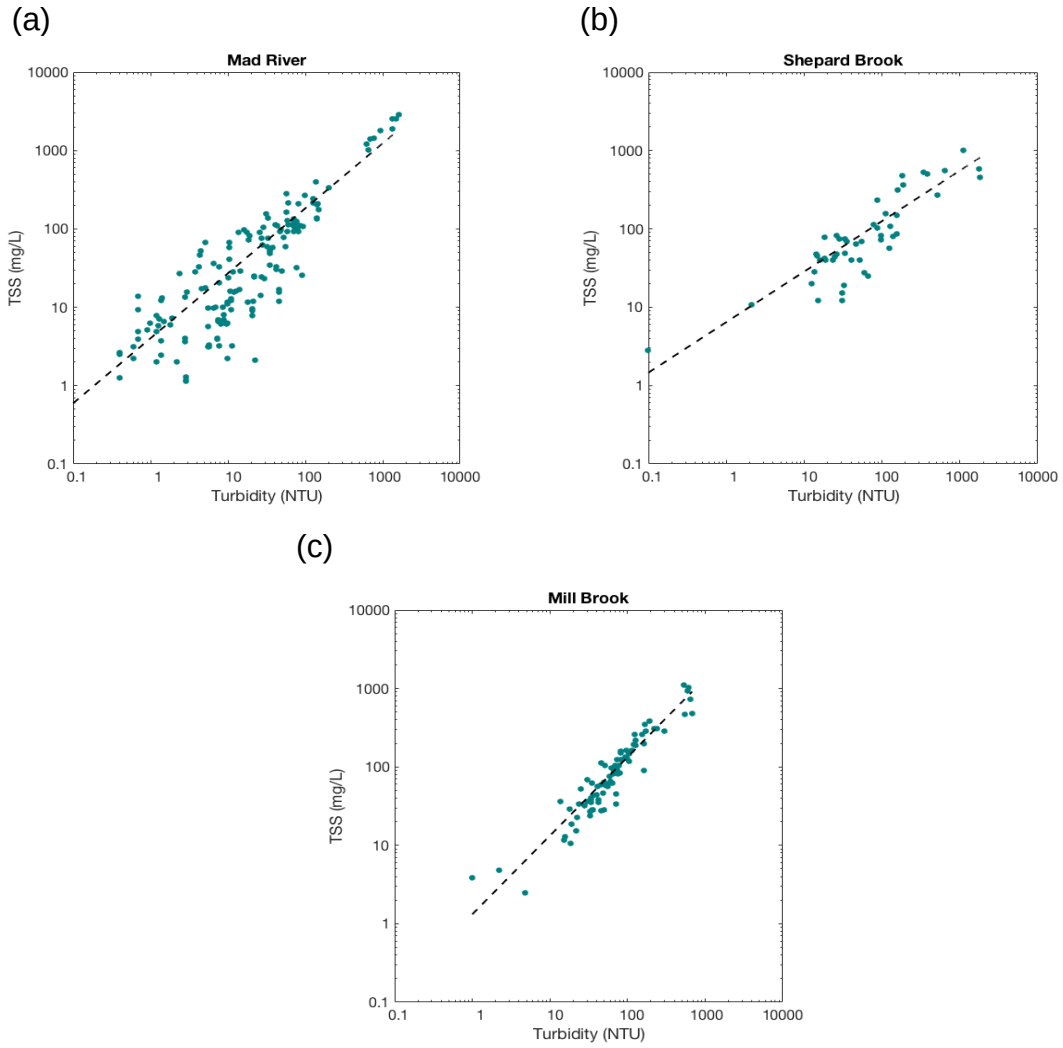


Figure 5.3. Relationship between TSS and turbidity for (a) Mad River, (b) Shepard Brook, and (c) Mill Brook. Dashed lines represent log-linear regression models.

Table 5.2. Parameters of fit log-linear rating curve of TSS and turbidity for monitoring sites.

Site	*a*	b_1	*BCF*	n	R^2 - adj
Mad River	2.8222	0.8292	1.4257	163	0.726
Shepard Brook	4.2248	0.7034	1.1688	49	0.752
Mill Brook	1.2179	1.0022	1.0838	83	0.882

differences between watersheds was not unexpected. However, because of the strong relationships between TSS and turbidity, using turbidity as a surrogate for suspended sediment was justified in this watershed; and we therefore calculate suspended sediment concentration from turbidity sensor readings to represent the actual suspended sediment concentration.

The TSS samples were also used to create SRCs at each site, again using log-linear regression models. The SRCs showed poorer fit compared to the turbidity-based TSS rating curves, especially at the Shepard Brook and Mill Brook sites (Figure 5.4 and Table 5.3). For the main stem site, the R^2-*adj* value was 0.594 indicating a reasonable fit. However, at the smaller, tributary scale, discharge is a poor predictor of TSS with R^2-*adj* values of only 0.287 and 0.131 for Shepard Brook and Mill Brook, respectively.

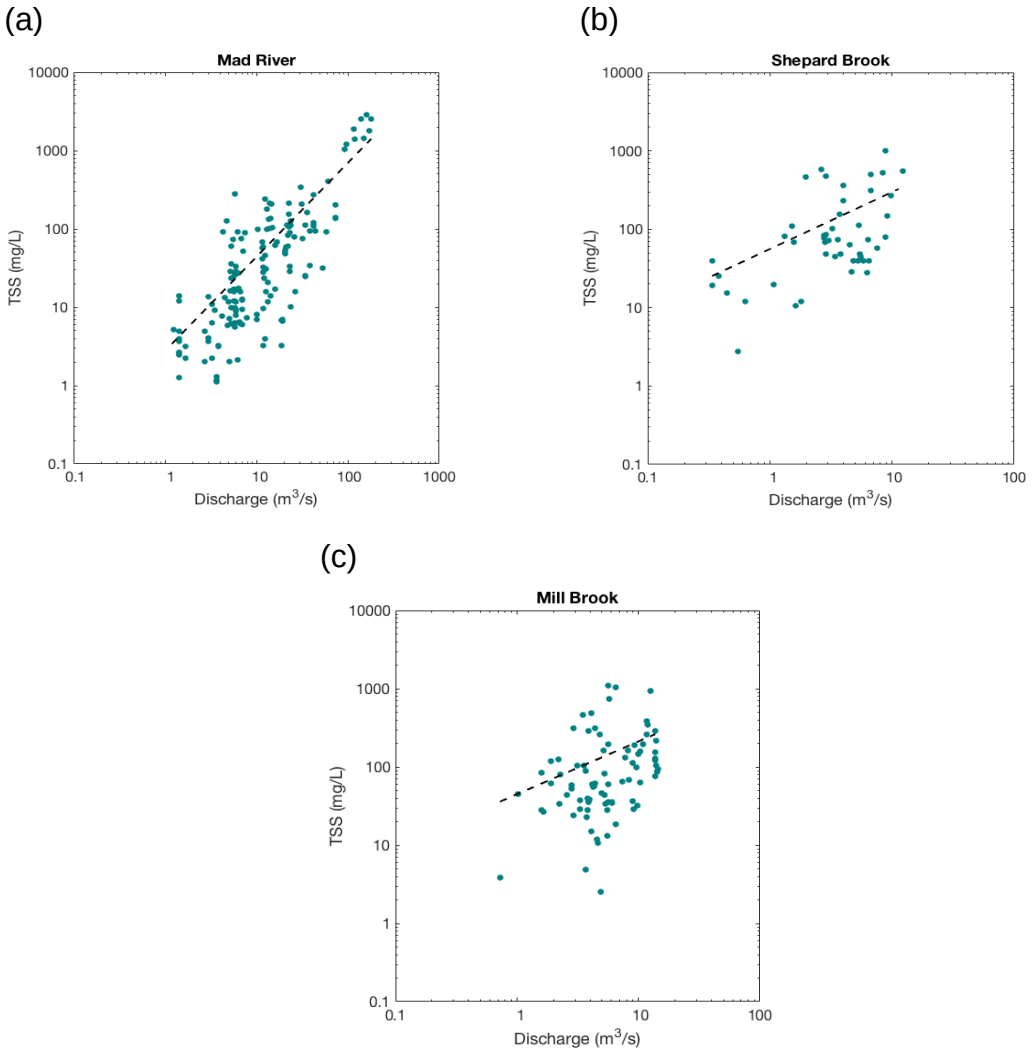


Figure 5.4. Relationship between TSS and discharge for (a) Mad River, (b) Shepard Brook, and (c) Mill Brook. Dashed lines represent log-linear regression models.

Table 5.3. Parameters of fit log-linear sediment rating curves for monitoring sites.

Site	*a*	b_1	*BCF*	n	R^2 - adj
Mad River	1.5545	1.2004	1.7767	163	0.594
Shepard Brook	32.7971	0.7263	1.7044	49	0.287
Mill Brook	24.2983	0.6746	1.8424	83	0.131

ANN data set preparation

The measured meteorological, discharge and sediment data were partitioned into a training and testing data for use with the rCPN model. Data from 2013 and 2014 comprised the training data set, and the data from 2015 was used for testing. Statistical analysis of the training and testing data (Table 5.4) showed higher mean, median, and maximum values for data (both discharge and TSS values) in the testing data set. While this is not ideal for training purposes, splitting the data chronologically is more reflective of real-world applications.

Table 5.4. Summary of statistical properties of training and testing data sets for each site

	Mad River		Shepard Brook		Mill Brook	
Statistic	*Train*	*Test*	*Train*	*Test*	*Train*	*Test*
No. Records	7071	5389	6924	5366	7198	5054
Discharge (m^3/s)						
Mean	4.00	6.65	0.56	0.53	0.94	1.39
Median	2.80	3.49	0.41	0.19	0.64	0.72
Min.	0.71	0.73	0.16	0.11	0.18	0.16
Max.	93.89	157.40	7.38	15.96	11.23	21.45
TSS (mg/L)						
Mean	11.0	19.8	6.5	7.3	5.9	7.8
Median	4.4	7.2	3.0	2.6	0.7	1.7
Min.	0.0	0.0	0.0	0.0	0.0	0.0
Max.	1035.3	1308.9	502.8	303.5	1034.4	1942.0

Temporal autocorrelation and cross-correlation analysis

To select the time period over which to lag the input variables, we performed a temporal autocorrelation for both discharge (Q) and TSS (Figures 5.5a and 5.6a) and cross-correlation analysis in both and TSS with all other variables (Figures 5.5 and 5.6, panels b-e). As expected with watersheds that vary in size, the temporal dependencies within and between variables differed among sites, especially between the Mad River and Mill Brook and Shepard Brook tributaries. The discharge (Q) correlogram (Figure 5.5a) illustrates strong autocorrelation at a time lag of 1 hour across all sites (***r*** of 0.971 to 0.986). The cross-

correlation of Q with the remaining hydrometeorological variables showed rainfall (P), soil moisture at shallow (10 cm depth - SM10) and deep (50 cm depth - SM50) to be positively correlated with Q; and water temperature (T) was negatively correlated. The time lags corresponding to the highest correlations are shaded; and the most notable differences between the Mad River watershed and its tributaries visible in both the (Q - P and Q - SM10 cross-correlograms of Figure 5.5b and 5.5c). This would be expected given differences in time of concentration of flow and varying spatial scales between the main stem and its tributaries.

We observed similar trends in the autocorrelation and cross-correlation of TSS data as displayed with discharge. TSS was positively correlated with other variables (P, SM10, SM50, Q); but in contrast to discharge, TSS was also positively correlated to water temperature (T), which may be indicative of a strong seasonal component. However, the correlation of both TSS and Q with T was weak across all sites (-.081 to 0.103). Analysis also showed soil moisture, at both depths, was less correlated with TSS than Q.

Compared to Q, TSS had higher maximum correlations with rainfall (max ***r*** of 0.459 – 0.569) at the tributary scale than was observed for Q (max ***r*** of 0.388 – 0.376) suggesting that sediment delivery is driven more by rainfall processes at the flashier, tributary scale compared to the main stem. Similar to Q, peak correlations between TSS and P occurred at shorter time lags in the tributaries compared to Mad River (Figure 5.6b), again indicating that rainfall is a key input for sediment prediction, especially in these smaller, flashier systems. While P is a better indicator of TSS at the tributary scale, the strong correlation of TSS with Q in the Mad River indicates that Q is the most important input variable for predicting TSS in higher-order systems. This implies that the selection of inputs will likely vary between watershed scales; and as a result, the rCPN maps different processes at the different watershed scales.

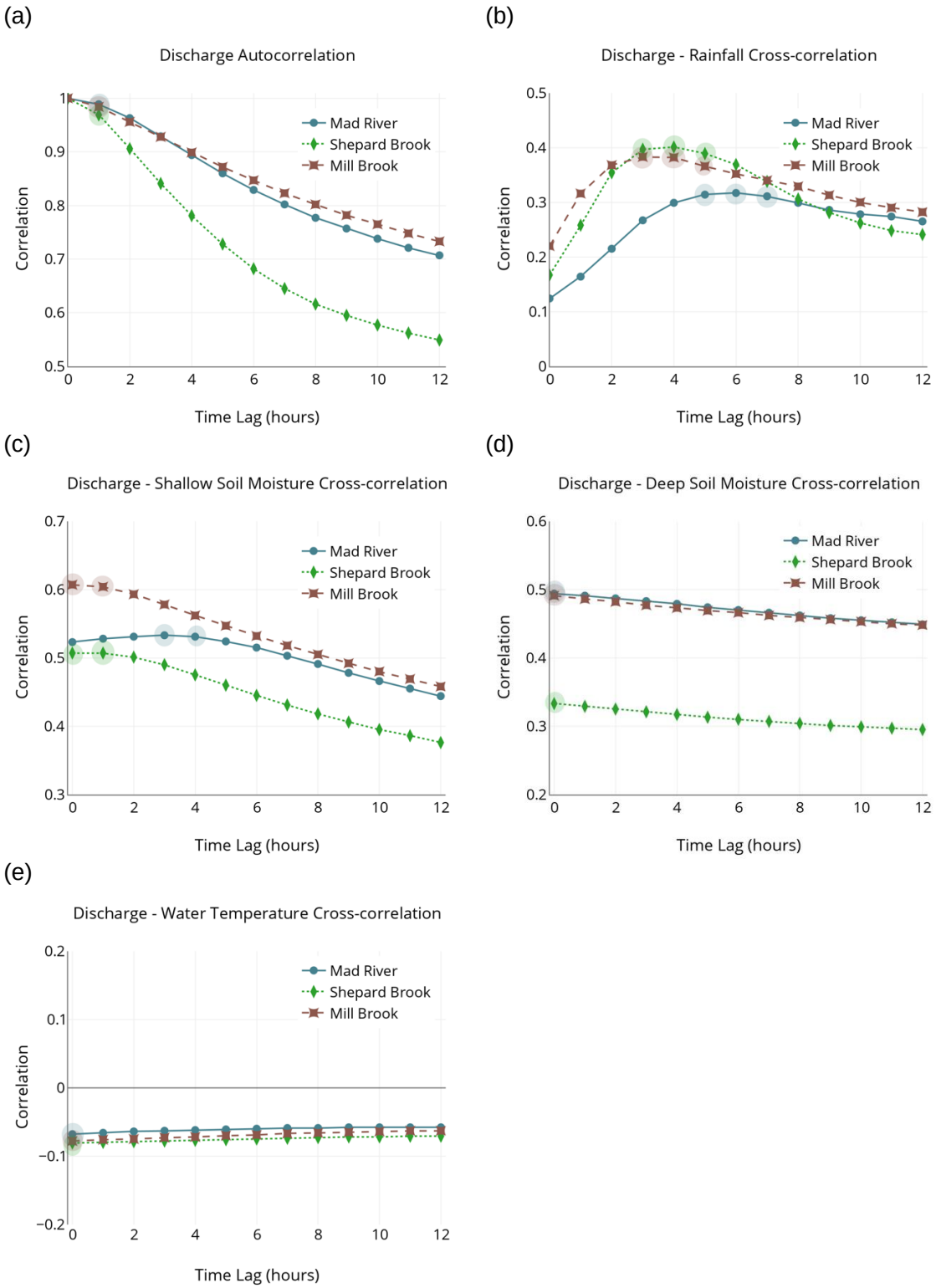


Figure 5.5. (a) Temporal auto-correlogram for discharge in the Mad River, Shepard Brook, and Mill Brook. Cross-correlograms of discharge with (b) rainfall, (c) shallow soil moisture (10cm depth), (d) deep soil moisture (50 cm depth), and (e) water temperature. Shaded points indicate the time lags used for the respective (rCPN) model input variables when predicting discharge.

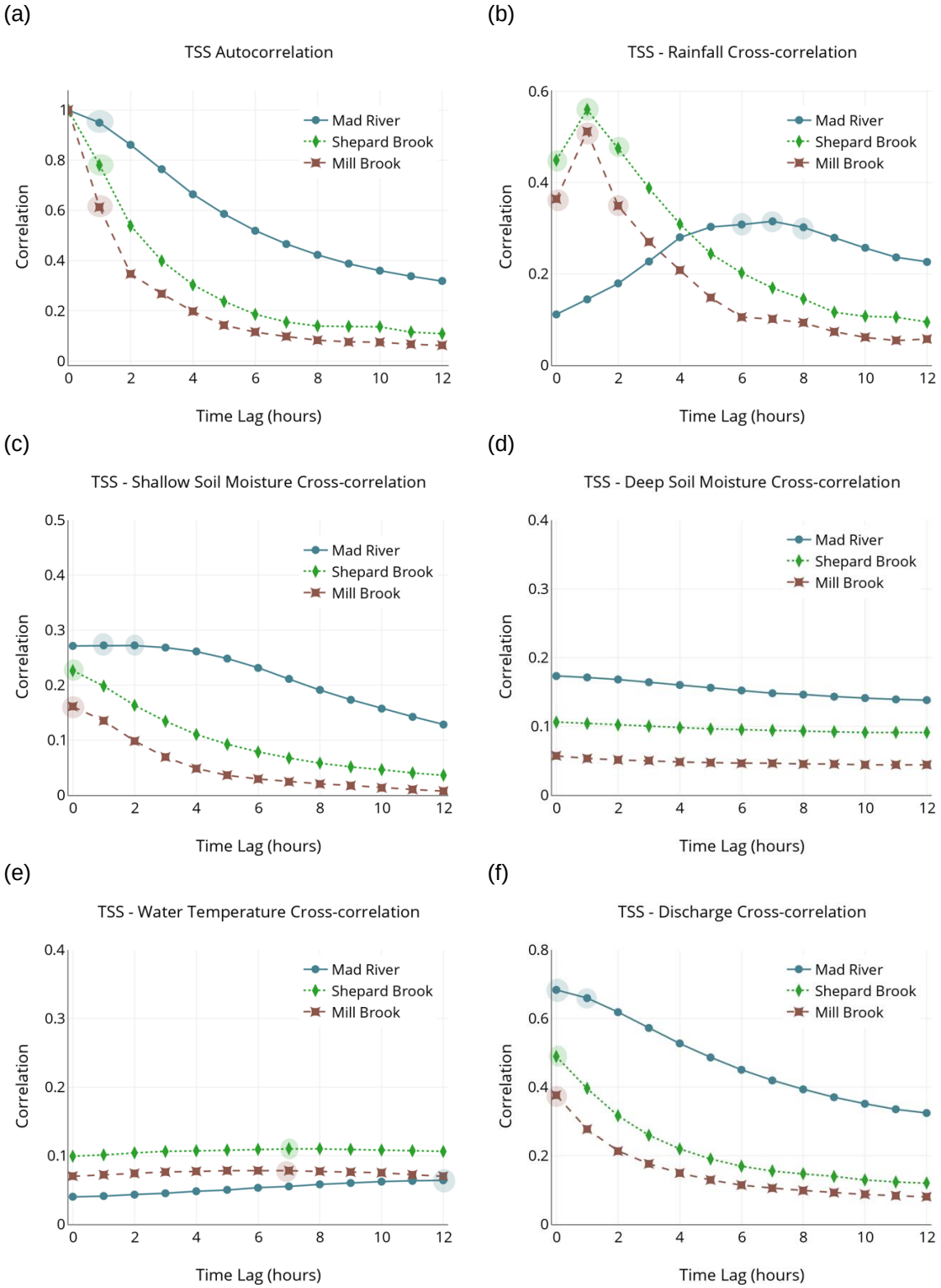


Figure 5.6. (a) Temporal auto-correlogram for TSS in the Mad River, Shepard Brook, and Mill Brook. Cross-correlograms of TSS with (b) rainfall, (c) shallow soil moisture (10cm depth), (d) deep soil moisture (50 cm), (e) water temperature, and (f) discharge. Shaded points indicate the time lags used for the respective (rCPN) model input variables to predict TSS.

rCPN model configuration

We used the correlograms and cross-correlograms for Q and TSS (Figures 5.5 and 5.6) to guide the selection (and temporal lag) of input variables, and then use two rCPN models in tandem to predict Q and TSS at both the Mad River and tributary scale. The model inputs are summarized in Table 5. Rather than using a single ANN to predict SSL, the tandem rCPN configuration (Figure 5.7) enables separate prediction of both Q and TSS without using *measured* antecedent Q or TSS as model inputs. SSL was then directly computed from the *predicted* Q and TSS time series.

The number of model input variables differed among the rCPN model tests. It is common to use multiple time lags of a single input variable, such as P, to a network to leverage the range of autocorrelation or cross-correlation. For the prediction of Q, we added three input nodes in order to lag P by three hours, two nodes for a two-hour lag for shallow soil moisture (SM10), and one node for deep soil moisture (SM50).

For predicting TSS (second rCPN in the tandem framework of Figure 5.7), we reduced the number of soil moisture (SM10) inputs to two time lags in the Mad River model and one (hr) lag when predicting at the tributaries. Three input nodes were needed to accommodate the lagged rainfall (P) at the Mad River site; and four nodes were used for P at the tributaries. In addition, the TSS models use additional nodes to accommodate the *estimated* Q from the prior Q prediction rCPN. We add two input nodes for *estimated* Q when predicting TSS at the Mad River site, and only one additional node for the tributaries.

Table 5.5. Summary of inputs to discharge and TSS prediction rCPN models

ANN Model	Inputs								
Mad River Discharge	P_{t-5}	P_{t-6}	P_{t-7}	$SM10_{t-3}$	$SM10_{t-4}$	$SM50_t$	T_t	$\boldsymbol{\hat{Q}_{t-1}}$	
Tributary Discharge	P_{t-3}	P_{t-4}	P_{t-5}	$SM10_t$	$SM10_{t-1}$	$SM50_t$	T_t	$\boldsymbol{\hat{Q}_{t-1}}$	
Mad River TSS	P_{t-6}	P_{t-7}	P_{t-8}	$SM10_{t-1}$	$SM10_{t-2}$	T_{t-12}	$\boldsymbol{\hat{Q}_t}$	$\boldsymbol{\hat{Q}_{t-1}}$	$\boldsymbol{\widehat{TSS}_{t-1}}$
Tributary TSS	P_t	P_{t-1}	P_{t-2}	P_{t-3}	$SM10_t$	T_{t-7}	$\boldsymbol{\hat{Q}_t}$	$\boldsymbol{\widehat{TSS}_{t-1}}$	

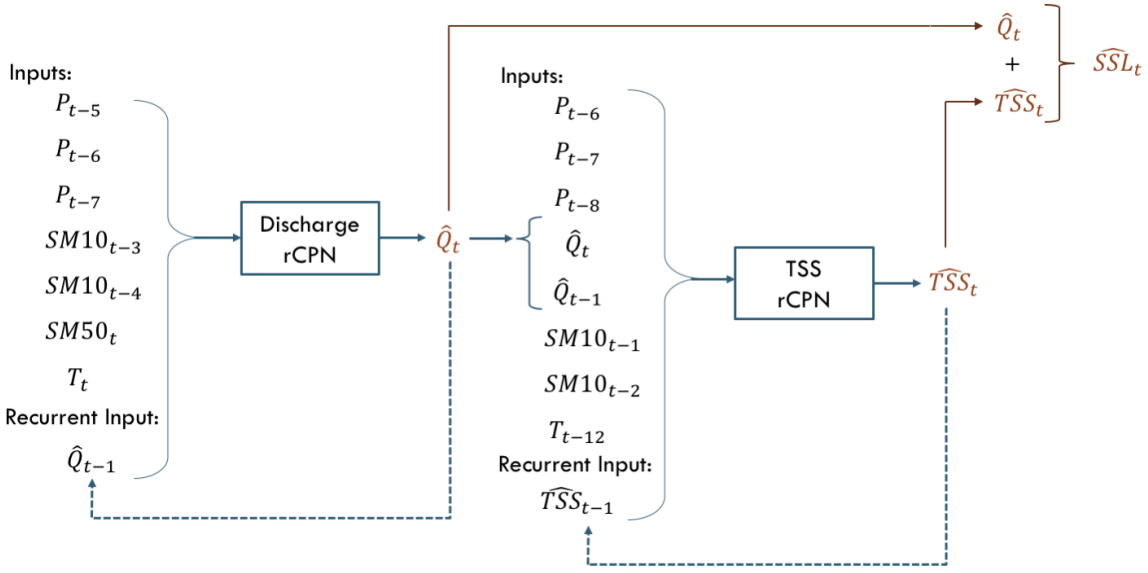


Figure 5.7. Configuration of suspended sediment load prediction model showing example inputs for the Mad River.

River discharge prediction

The first step toward predicting suspended sediment loads is to estimate river discharge (Q). Ideally, one would like to have measured discharge at the location where SSL needs to be predicted. Given that one is rarely that fortunate, we assessed the performance of the rCPN model for *predicting* Q at the Mad River and both tributaries, and used those predictions as input to a subsequent rCPN model for estimating SSL. Figure 5.8 presents a

subset (a 90-day period in the summer of 2015) of the testing data and model predictions for the Mad River and Mill Brook. Similar performance (NSE = 0.68) was observed on the testing data at both the main stem and the Mill Brook tributary. Model efficiency was lower at Shepard Brook with NSE = 0.43 for the testing data, and may be the result of the flashiness of the site in response to rainfall, which is also evident in Shepard Brook having the lowest autocorrelation in Q of the three sites (Figure 5a). However, model performance (Table 5.6) is at or above the performance metrics of existing studies on streamflow prediction at ungauged sites using hourly data (Besaw et al., 2010).

The rCPN predictions of Q showed conditional bias in the under-prediction of peak flows across all sites, (Figures 5.6b and 5.6d). This result is not necessarily surprising, as other studies have observed similar behavior (Besaw et al., 2010; Fi-John Chang & Chen, 2001; Firat, 2008), and can be partially attributed to the over-representation of low flows in the training data. The prediction of peak flow values can often be improved by training separate ANNs (Hu, Lam, & Ng, 2001; P. Singh & Deo, 2007) on different ranges of Q (e.g., train an ANN for high flow and another for flow). We used a single rCPN model and trained over the entire range of Q in this work because the model performance was acceptable and to simplify the model design for proof of concept.

Table 5.6. Summary statistics of the rCPN model performance for predicting discharge

	Training			**Testing**		
Model	**RMSE (m^3/s)**	**NSE**	**Correlation**	**RMSE (m^3/s)**	**NSE**	**Correlation**
Mad River	0.95	0.95	0.98	5.44	0.68	0.85
Shepard Brook	0.13	0.94	0.98	0.76	0.46	0.71
Mill Brook	0.20	0.95	0.98	1.05	0.68	0.83

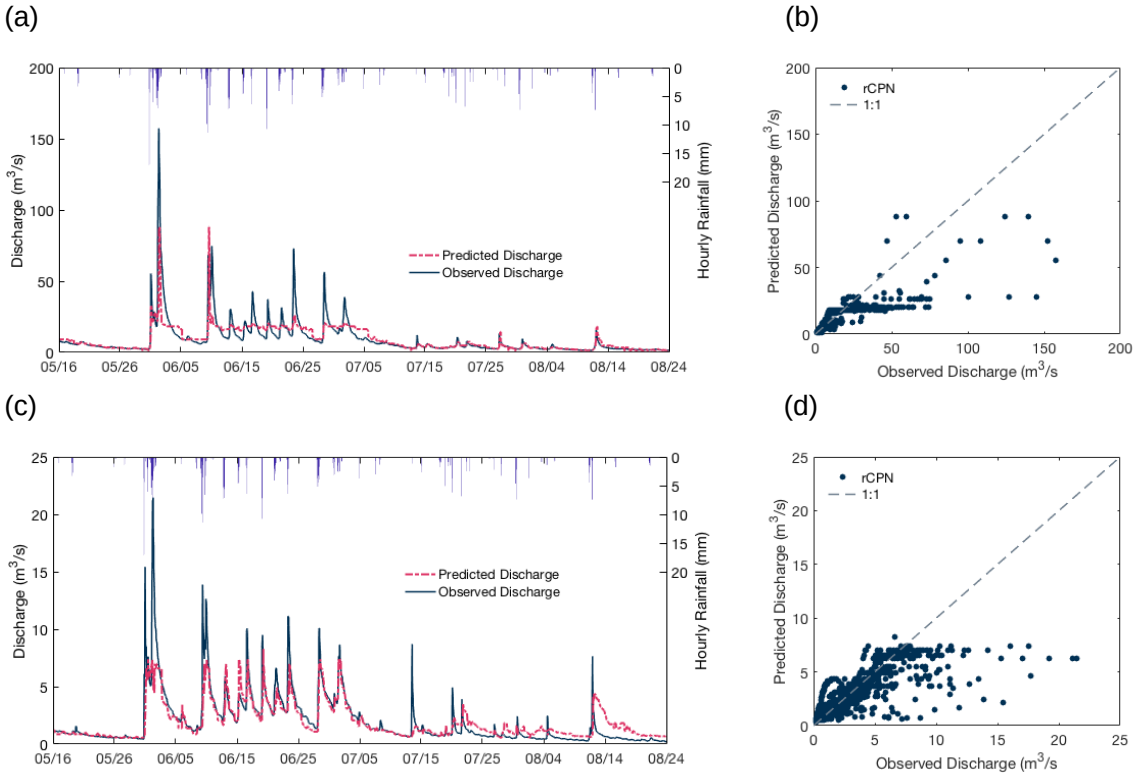


Figure 5.8. Hyetograph plotted against the predicted and observed hydrographs for a portion of the testing data set for (a) Mad River and (c) Mill Brook. Comparison of observed versus predicted discharge values for (b) Mad River and (d) Mill Brook.

Prediction of TSS and SSL

We used the predicted discharge from the Q rCPN along with other input data to first predict TSS using a second rCPN, and then combined the estimated TSS with the estimated Q, to provide predictions of SSL. We evaluated the TSS rCPN by training and testing on each of the sites separately, as this is more representative of a real-world application given limited historical observations. Based solely on traditional model performance metrics, the prediction of TSS using the rCPN is not very reliable (see summary statistics, Table 5.7). NSE ranged from 0 to 0.2 and correlation from 0.16 to 0.51 across the three sites.

Model predictions were notably biased low during the later summer portion of the testing data, compared to the early summer period (Figures 5.9a and 5.9c). This may indicate a seasonal trend in watershed TSS response that was not sufficiently represented in the training data set to enable the rCPN model to properly map it. The late summer storms are also, on average, smaller and shorter hydrological events suggesting the rCPN model may have a conditional bias to under-predict TSS during small events. Another explanation for the under prediction of certain storms result from rainfall records that do not always capture the actual rainfall. In the Mad River watershed, summer storm events are frequently isolated to smaller portions of the watershed; and as a result, the location of the rain gauges do not always reflect the actual rainfall. Dense rain gauge networks or augmentation of the rainfall record from radar data may improve the reliability of predictions.

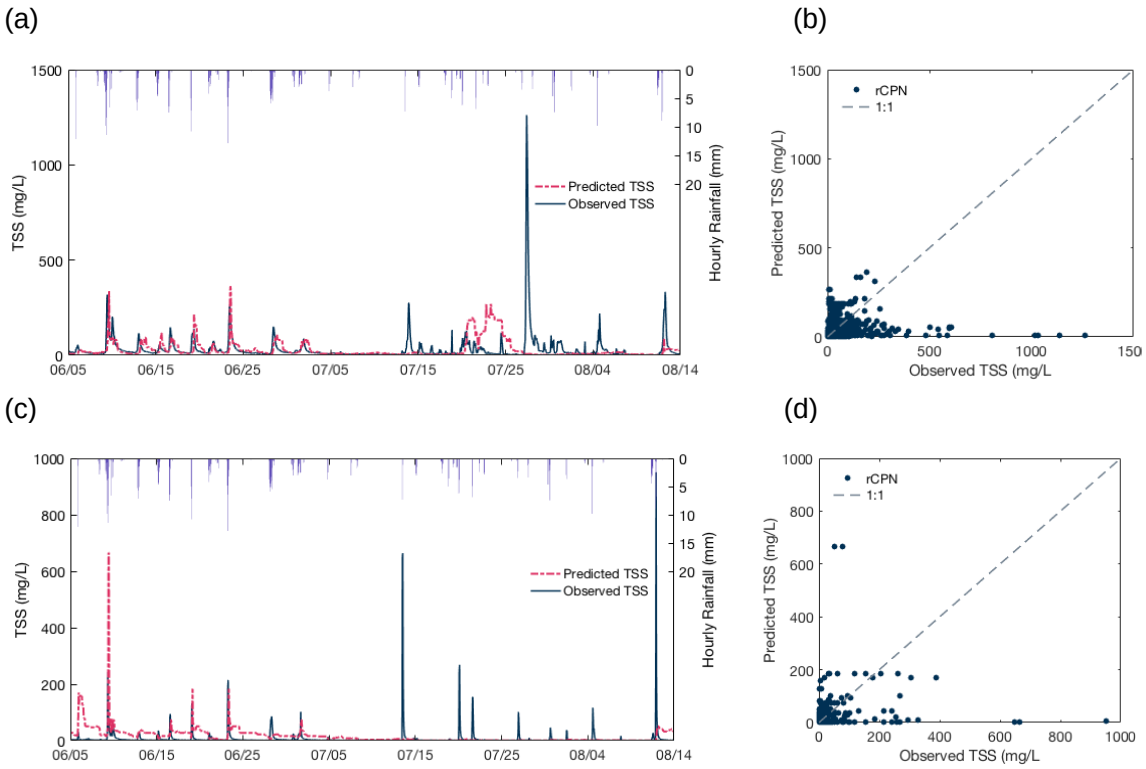


Figure 5.9. Hyetograph plotted against the predicted and observed sedigraphs for a portion of the testing data set for (a) Mad River and (c) Mill Brook. Comparison of observed versus predicted TSS values for (b) Mad River and (d) Mill Brook.

Existing studies on the prediction of TSS in ungauged locations are nearly non-existent; as such, there are few findings for comparison with our results. The few studies that have attempted prediction without measured discharge and antecedent sediment have been restricted to predicting SSL and at a daily time step (Alp & Cigizoglu, 2007; Raghuwanshi et al., 2006; Zhu et al., 2007). The larger prediction errors of TSS compared to Q may be due in part to the inherent complexity, hysteretic effects and weak correlations between Q and TSS, especially at the tributary scale (see the poor SRC model fit in Table 5.5 and Figures 5.4b and 5.4c). Training separate rCPNs (i.e., storm events and baseflow conditions separately) is one option that could be investigated to improve TSS model performance on an event basis.

Table 5.7. Summary statistics of model performance for TSS and SSL predictions

	Training			Testing		
Model	**RMSE (m^3/s)**	**NSE**	**Correlation**	**RMSE (mg/L)**	**NSE**	**Correlation**
TSS						
Mad River	11.2	0.88	0.94	60.7	0.0	0.20
Shepard Brook	12.5	0.56	0.78	16.79	0.23	0.51
Mill Brook	29.8	0.20	0.60	43.00	0.01	0.16
SSL						
Mad River	-	-	-	10.67	0.17	0.47
Shepard Brook	-	-	-	0.24	0.73	0.86
Mill Brook	-	-	-	1.76	0.08	0.36

One of the motivations for this research and a key desired outcome was to estimate sediment discharge from the river sub-basins. To assess the model performance in this manner, we combined the model estimates of Q and TSS to predict SSL, and then compared the predicted SSL to both observed SSL and estimates generated using the traditional SRC method. Figure 5.10, plots the rCPN predictions of SSL against observations over a portion

of the testing data set. Similar to the TSS predictions, the rCPN predictions are reasonable; however, portions of the testing data record (specifically the smaller later summer storms) have poor performance. Because discharge is an integral component of SSL, the majority of sediment load occurs during large hydrological flow events. Therefore, degraded model performance on smaller events may be acceptable for predicting long-term sediment discharge.

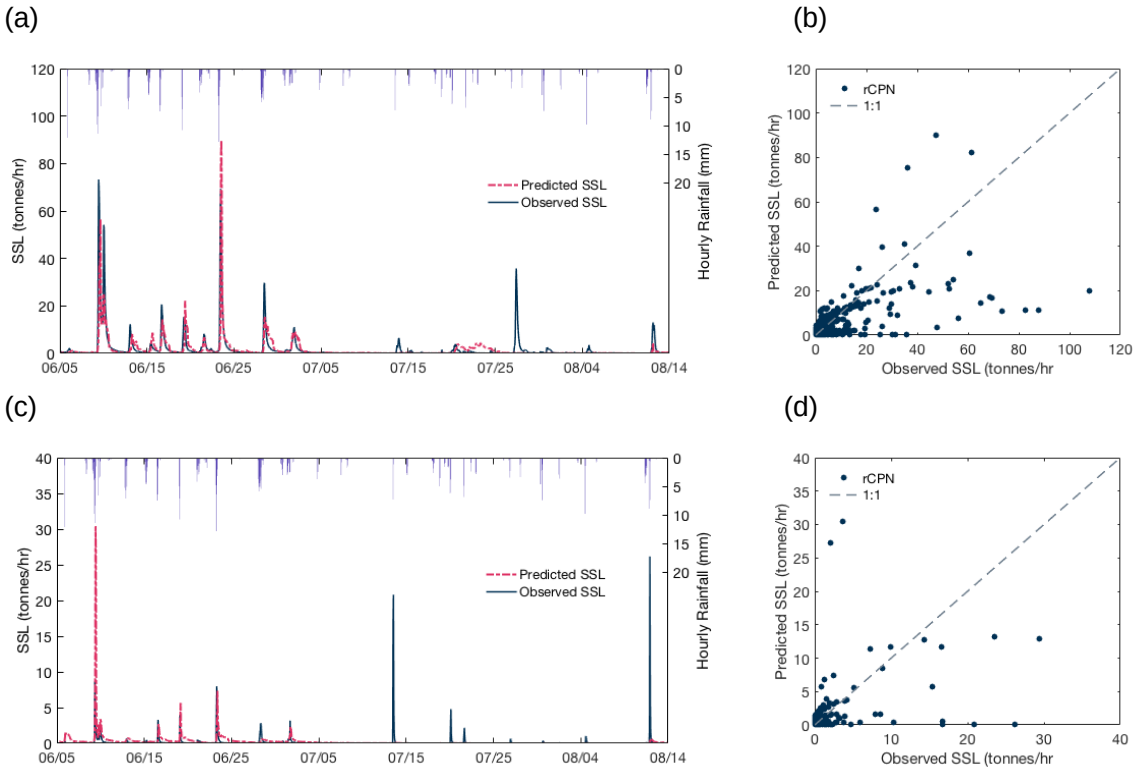


Figure 5.10. Hyetograph plotted against the predicted and observed sediment load for a portion of the testing data set for (a) Mad River and (c) Mill Brook. Comparison of observed versus predicted SSL values for (b) Mad River and (d) Mill Brook.

The performance metrics suggest better prediction of SSL compared to TSS (Table 5.7); however, there is a high degree of variability among sites. NSE ranges between 0.08 and 0.73; and correlation between 0.36 and 0.86 were observed for the rCPN-based predictions of SSL across all sites. Considering that the model uses only two years of measurement data for

training, the small size of the subwatersheds, and the time step is hourly, this performance is very favorable compared to limited published findings. Alp and Cigizoglu (2007) obtained a maximum value of NSE = 0.213 when modeling SSL at the daily time scale using only rainfall as inputs and on a much larger watershed compared to the Mad River. The rCPN predictions captured the general trend of SSL during hydrological events well. Figure 5.11 shows the rCPN and SRC predictions of SSL for Mill Brook over a two-week subset of the testing data and illustrates that the sediment pulse observed during events is more accurately modeled by the rCPN than using a SRC model.

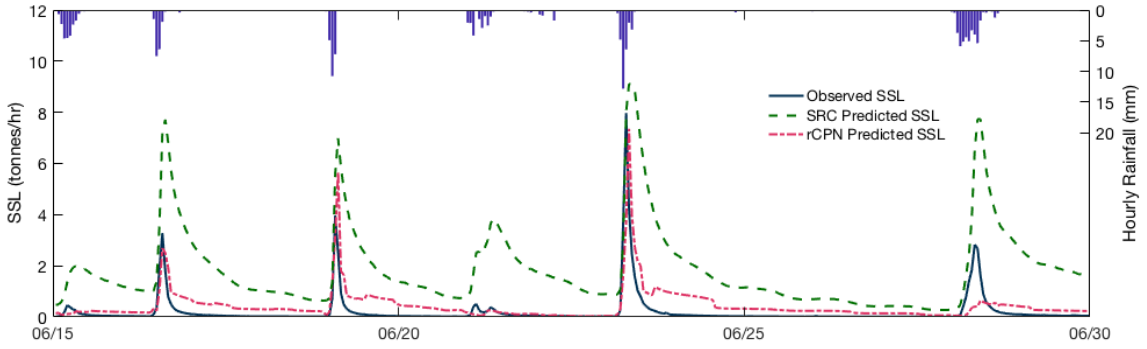


Figure 5.11. Comparison between observed SSL (solid line) and that predicted using a SRC model (green dashed line) and the rCPN model (red dashed line) over a two-week portion of testing data set for Mill Brook.

Table 5.8. Summary metrics of modeled SSL and SSY using SRC and the rCPN models compared to measured observations for the testing dataset (May 2015 to December 2015)

	Cumulative SSL (tonnes)			SSY (tonnes/km²)			Percent Error	
Site	**Observed**	**SRC**	**rCPN**	**Observed**	**SRC**	**rCPN**	**SRC**	**rCPN**
Mad River	8667	1561	3991	35.5	6.4	16.4	80%	-54%
Shepard Brook	284	986	259	6.4	22.1	5.8	247%	-9%
Mill Brook	730	2585	860	14.7	51.9	17.3	254%	18%

We compared the SSL estimates from both the rCPN and SRC models to observed SSL by calculating the cumulative SSL over the training data record (Table 5.8 and Figure 5.12). For all three sites, the cumulative SSL predicted by the rCPN model was closer to observed SSL. The percent error in total SSL load for the rCPN model ranged from -54% to 18%. For all sites, SRCs were found to greatly overestimate SSL with percent errors ranging from 80% to 254%, with both tributary sites having approximately 250% error. This is not unexpected, given the poor SRC model fit for Mill Brook and Shepard Brook. Other studies have found SRCs to be poor models for predicting suspended sediment data (Harrington & Harrington, 2013; Uhrich & Bragg, 2003; Warrick, Melack, & Goodridge, 2015). Of note, we only evaluated one type of SRC, the commonly applied log-linear regression model; however, other types (e.g., power-law or polynomial regression models) may provide some improvement in the SRC-based estimates.

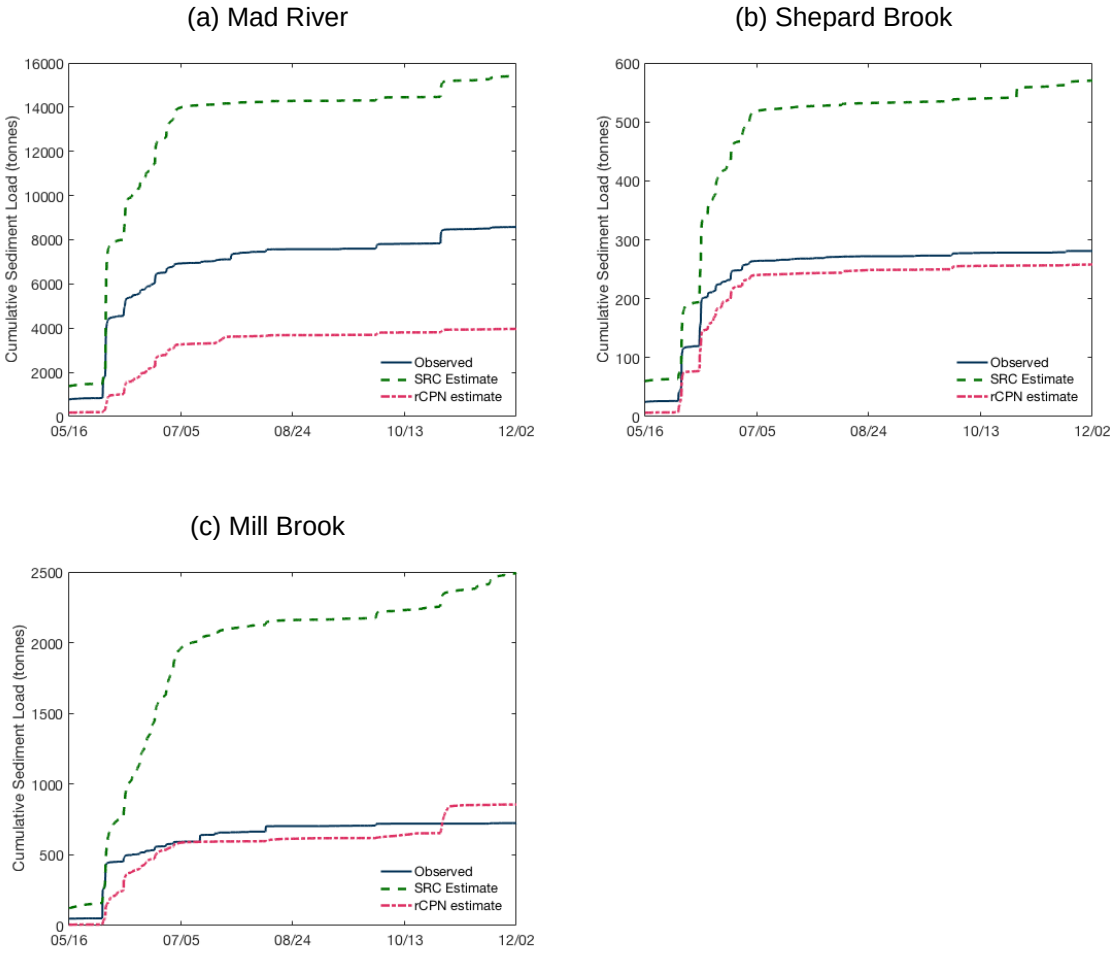


Figure 5.12. Cumulative SSL (observed vs predictions) over the entire testing data set for (a) Mad River, (b) Shepard Brook, and (c) Mill Brook.

Prediction in ungauged stream locations

In order to test the applicability of the rCPN model in ungauged stream locations, we trained the model on one of the tributaries, Shepard Brook, and then predicted SSL for the Mill Brook watershed. Figure 5.13 shows a comparison of the observed and predicted SSL for Mill Brook. While the Mill Brook and Shepard Brook watershed are both fourth-order streams, the magnitude of median and peak flows differ. To compensate for the difference, we utilized a simple scaling method that adjusts estimated discharge values based on the ratio of the drainage areas. No scaling factor was applied to estimates of TSS. When comparing the model performance at Mill Brook to the performance at Shepard Brook (i.e., rCPN trained on data from the same site), the predictions were biased lower and did not capture the magnitude of event peaks well.

The cumulative SSL estimated for Mill Brook using the rCPN model trained on data from Shepard Brook, shows predictions to be less accurate (magnitude of percent error increasing from 18% to -54%) than using training data from Mill Brook. Although this might be expected, the predictions still showed improvement compared to the SRC estimates (Figure 5.14). Given that larger peak TSS is observed in Mill Brook compared to Shepard Brook, it is not unexpected that training the model on Shepard Brook data and predicting in Mill Brook would result in lower biased estimates. To account for the difference in sediment yields between watersheds, a regionalization method would need to be employed without requiring in-stream observations. As such, the model as currently tested would be applicable for prediction in watersheds with similarly expected sediment yields. Transferability between locations, therefore, should be guided by watershed characteristics with existing studies on regionalization of hydrological models (Oudin, Andréassian, Perrin, Michel, & Le Moine, 2008; Razavi & Coulibaly, 2013) offering a potential guide to metrics such as drainage area,

basin relief, drainage density, mean annual rainfall, percent forest cover, mean soil depth and percent clay soils.

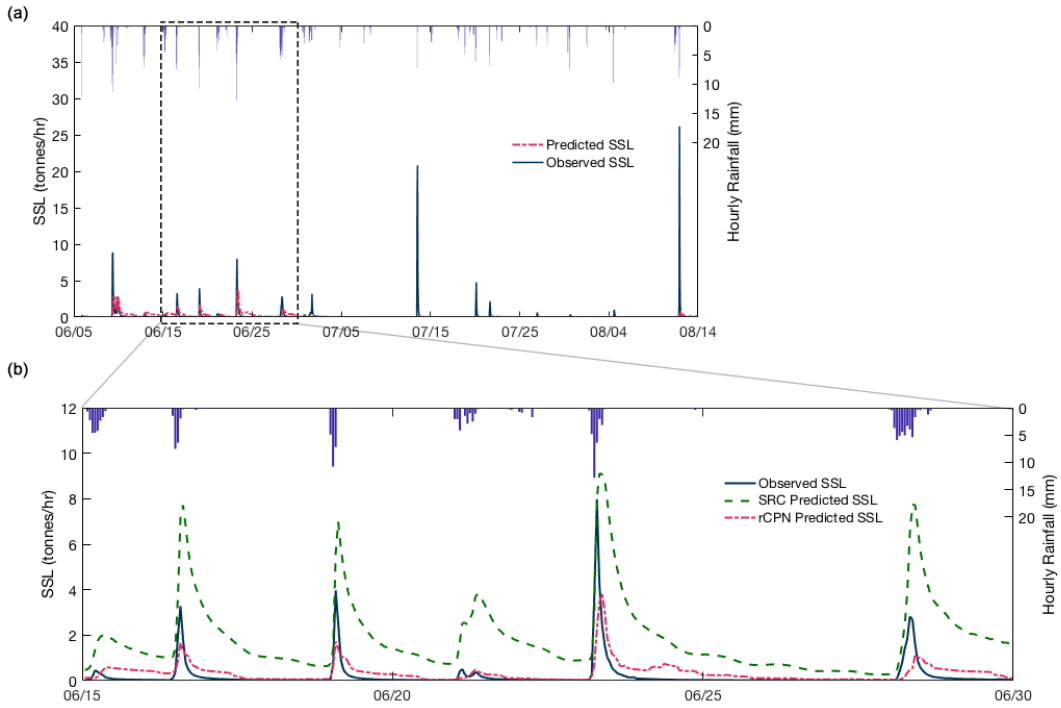


Figure 5.13. (a) Hyetograph plotted against the predicted and observed sediment load for a portion of the testing data set for Mill Brook using the rCPN model trained on Shepard Brook data. (b) Comparison between observed and predicted SSL for a two-week portion of testing data set for Mill Brook rCPN model trained on Shepard Brook data. SRC-based predictions of SSL shown for reference.

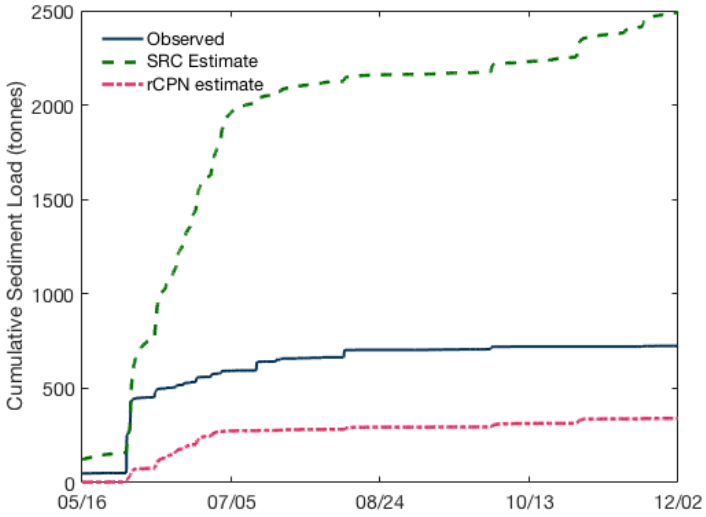


Figure 5.14. Cumulative SSL (observed vs predictions) over a two-week portion of the testing data set for Mill Brook using rCPN model trained on Shepard Brook site. SRC-based prediction of SSL is shown for reference

Conclusions

We present a versatile ANN model for predicting SSL in river systems using only local hydrometeorological inputs of rainfall, soil moisture, and temperature at the hourly time-scale. Two recurrent counterpropagation networks (rCPNs) were used in tandem to estimate river discharge and suspended sediment concentration in addition to SSL. Comparison to the traditional sediment rating curves showed the rCPN model was able to more accurately estimate SSL and more realistically model the timing and shape of the sedigraph during hydrological evens. The latter is of particular importance for capturing event-scale phenomena such as hysteresis in the SSC-Q relationship. Model performance in predicting TSS and SSL was better during large events, than the smaller, flashy, summer events indicating the model may be sensitive to the training data.

The rCPN model developed in this work represents an advancement for real-world applications over existing ANN suspended sediment prediction methods. To-date, SSC/SSL

prediction with ANNs has been restricted to using daily or longer time steps; we demonstrate the rCPN model's capability at finer resolution (hourly data). This is especially important for prediction in small river systems where suspended sediment dynamics predominantly occur at the sub-daily scale. Importantly, we also found that SSL could be modeled with reasonable accuracy, without using measured discharge and sediment data as input variables. This expands the utility of suspended sediment prediction made using ANNs to ungauged locations or those with limited monitoring records. The tandem rCPN framework accommodates either estimated or measured discharge for prediction of TSS and SSL depending on what is available at the target site.

The availability of high-frequency suspended sediment data was essential to the success of this study. In-stream turbidity sensors provided a cost-effective and reliable method of obtaining high-frequency sediment data; and we recommend their further adoption in streams without plankton or soluble colored organic compounds. The rCPN model presented here is primarily driven by rainfall data and therefore is a critical input. We recommend the deployment of a dense rain gauge network to account for both the spatial variability of rainfall and potential equipment malfunctions. The existing rain station network provided hourly measurements; however, the number in our study area was not sufficient to characterize the spatial variability of rainfall within the catchment. We also recommend the deployment of soil moisture sensors, given their relative low-cost and amount of information added to the model. If the latter is not possible, then substitution of soil moisture or baseflow indices may be appropriate. Finally, we recommend that variables and their respective time lags be selected on an individual watershed basis or within similarly grouped watersheds, given the differences we observed between the main stem and tributary watersheds.

We demonstrated reasonable predictions at watersheds lacking discharge or sediment data by using training data from a nearby watershed as a proof of concept. The nearby watershed featured similar topography and land use/cover and as such was expected to have similar sediment yields. Further studies on regionalization methods are required to extend this work to ungauged locations that are less similar to the training location. As presented, the data-driven method offers water resources managers a new tool for predicting suspended sediment discharge from river systems provided that local hydrometeorological data exist.

Acknowledgements

This research was supported by Vermont EPSCoR with funds from the National Science Foundation (NSF) Grant EPS-1101317 and EPS-1556770. Additional support provided by NSF Graduate Research Fellowship under Grant No. DGE-0925179NSF, The Vermont Water Resources and Lake Studies Center, and the Robert & Patricia Switzer Foundation. The authors acknowledge the additional contributions of Kristen Underwood, Beverley Wemple, Alex Morton, and Jordan Duffy and the support of landowners of monitoring sites.

References

Abrahart, R. J., Anctil, F., Coulibaly, P., Dawson, C. W., Mount, N. J., See, L. M., … Wilby, R. L. (2012). Two decades of anarchy? Emerging themes and outstanding challenges for neural network river forecasting. *Progress in Physical Geography*, *36*(4), 480–513. https://doi.org/10.1177/0309133312444943

Abrahart, R. J., See, L. M., Heppenstall, A. J., & White, S. M. (2008). Neural network estimation of suspended sediment: potential pitfalls and future directions. In *Practical Hydroinformatics* (pp. 139–161). Springer.

Afan, H. A., El-shafie, A., Mohtar, W. H. M. W., & Yaseen, Z. M. (2016). Past, present and prospect of an Artificial Intelligence (AI) based model for sediment transport prediction. *Journal of Hydrology*, *541*, *Part B*, 902–913. https://doi.org/10.1016/j.jhydrol.2016.07.048

Alp, M., & Cigizoglu, H. K. (2007). Suspended sediment load simulation by two artificial neural network methods using hydrometeorological data. *Environmental Modelling & Software*, *22*(1), 2–13. https://doi.org/10.1016/j.envsoft.2005.09.009

ASCE. (2000a). Artificial neural networks in hydrology. I: Preliminary concepts. *Journal of Hydrologic Engineering*, *5*(2), 115–123.

ASCE. (2000b). Artificial neural networks in hydrology. II: hydrologic applications. *Journal of Hydrologic Engineering*, *5*(2), 124–137.

Besaw, L. E., Rizzo, D. M., Bierman, P. R., & Hackett, W. R. (2010). Advances in ungauged streamflow prediction using artificial neural networks. *Journal of Hydrology*, *386*(1–4), 27–37. https://doi.org/10.1016/j.jhydrol.2010.02.037

Buyukyildiz, M., & Kumcu, S. Y. (2017). An Estimation of the Suspended Sediment Load Using Adaptive Network Based Fuzzy Inference System, Support Vector Machine and Artificial Neural Network Models. *Water Resources Management*, *31*(4), 1343–1359. https://doi.org/10.1007/s11269-017-1581-1

Chang, F.-J., Chang, L.-C., & Huang, H.-L. (2002). Real-time recurrent learning neural network for stream-flow forecasting. *Hydrological Processes*, *16*(13), 2577–2588. https://doi.org/10.1002/hyp.1015

Chang, F.-J., & Chen, Y.-C. (2001). A counterpropagation fuzzy-neural network modeling approach to real time streamflow prediction. *Journal of Hydrology*, *245*(1–4), 153–164. https://doi.org/10.1016/S0022-1694(01)00350-X

Cigizoglu, H. K., & Alp, M. (2006). Generalized regression neural network in modelling river sediment yield. *Advances in Engineering Software*, *37*(2), 63–68. https://doi.org/10.1016/j.advengsoft.2005.05.002

Cigizoglu, H. K., & Kisi, Ö. (2006). Methods to improve the neural network performance in suspended sediment estimation. *Journal of Hydrology*, *317*(3–4), 221–238. https://doi.org/10.1016/j.jhydrol.2005.05.019

Cobaner, M., Unal, B., & Kisi, O. (2009). Suspended sediment concentration estimation by an adaptive neuro-fuzzy and neural network approaches using hydro-meteorological data. *Journal of Hydrology*, *367*(1–2), 52–61. https://doi.org/10.1016/j.jhydrol.2008.12.024

Edwards, T. K., & Glysson, G. D. (1999). *Field methods for measurement of fluvial sediment: US Geological Survey Techniques of Water-Resources Investigations*. US Geological Survey.

Firat, M. (2008). Comparison of artificial intelligence techniques for river flow forecasting. *Hydrology and Earth System Sciences*, *12*(1), 123–139.

Gao, P. (2008). Understanding watershed suspended sediment transport. *Progress in Physical Geography*, *32*(3), 243–263. https://doi.org/10.1177/0309133308094849

Gettel, M., S. Gulliver, J., Kayhanian, M., DeGroot, G., Brand, J., Mohseni, O., & J. Erickson, A. (2011). Improving suspended sediment measurements by automatic samplers. *Journal of Environmental Monitoring*, *13*(10), 2703–2709. https://doi.org/10.1039/C1EM10258C

Han, D., & Bray, M. (2006). Automated Thiessen polygon generation. *Water Resources Research*, *42*(11), W11502. https://doi.org/10.1029/2005WR004365

Harrington, S. T., & Harrington, J. R. (2013). An assessment of the suspended sediment rating curve approach for load estimation on the Rivers Bandon and Owenabue, Ireland. *Geomorphology*, *185*, 27–38. https://doi.org/10.1016/j.geomorph.2012.12.002

Hecht-Nielsen, R. (1988). Applications of counterpropagation networks. *Neural Networks*, *1*(2), 131–139. https://doi.org/10.1016/0893-6080(88)90015-9

Heng, S., & Suetsugi, T. (2013). Using Artificial Neural Network to Estimate Sediment Load in Ungauged Catchments of the Tonle Sap River Basin, Cambodia. *Journal of Water Resource and Protection*, *05*(02), 111–123. https://doi.org/10.4236/jwarp.2013.52013

Horowitz, A. J. (2003). An evaluation of sediment rating curves for estimating suspended sediment concentrations for subsequent flux calculations. *Hydrological Processes*, *17*(17), 3387–3409. https://doi.org/10.1002/hyp.1299

Hu, T. S., Lam, K. C., & Ng, S. T. (2001). River flow time series prediction with a range-dependent neural network. *Hydrological Sciences Journal*, *46*(5), 729–745. https://doi.org/10.1080/02626660109492867

Joshi, R., Kumar, K., & Adhikari, V. P. S. (2016). Modelling suspended sediment concentration using artificial neural networks for Gangotri glacier. *Hydrological Processes*, *30*(9), 1354–1366. https://doi.org/10.1002/hyp.10723

Kakaei Lafdani, E., Moghaddam Nia, A., & Ahmadi, A. (2013). Daily suspended sediment load prediction using artificial neural networks and support vector machines. *Journal of Hydrology*, *478*, 50–62. https://doi.org/10.1016/j.jhydrol.2012.11.048

Kamel, K., Mahmoud, T., Bissonnais, Y., & Mahmoud, T. (2014). Assessment of the artificial neural networks to geomorphic modelling of sediment yield for ungauged catchments, Algeria. *Journal of Urban and Environmental Engineering*, *8*(2), 175–185. https://doi.org/10.4090/juee.2014.v8n2.175185

Kisi, O. (2005). Suspended sediment estimation using neuro-fuzzy and neural network approaches. *Hydrological Sciences Journal*, *50*(4). https://doi.org/10.1623/hysj.2005.50.4.683

Kisi, O., & Shiri, J. (2012). River suspended sediment estimation by climatic variables implication: Comparative study among soft computing techniques. *Computers & Geosciences*, *43*, 73–82. https://doi.org/10.1016/j.cageo.2012.02.007

Krueger, T., Quinton, J. N., Freer, J., Macleod, C. J. A., Bilotta, G. S., Brazier, R. E., … Haygarth, P. M. (2012). Comparing empirical models for sediment and phosphorus transfer from soils to water at field and catchment scale under data uncertainty. *European Journal of Soil Science*, *63*(2), 211–223. https://doi.org/10.1111/j.1365-2389.2011.01419.x

Kumar, D., Pandey, A., Sharma, N., & Flügel, W.-A. (2015). Modeling Suspended Sediment Using Artificial Neural Networks and TRMM-3B42 Version 7 Rainfall Dataset. *Journal of Hydrologic Engineering*, *20*(6), C4014007. https://doi.org/10.1061/(ASCE)HE.1943-5584.0001082

Kumar, D., Pandey, A., Sharma, N., & Flügel, W.-A. (2016). Daily suspended sediment simulation using machine learning approach. *CATENA*, *138*, 77–90. https://doi.org/10.1016/j.catena.2015.11.013

Liu, Q.-J., Shi, Z.-H., Fang, N.-F., Zhu, H.-D., & Ai, L. (2013). Modeling the daily suspended sediment concentration in a hyperconcentrated river on the Loess Plateau, China, using the Wavelet–ANN approach. *Geomorphology*, *186*, 181–190. https://doi.org/10.1016/j.geomorph.2013.01.012

Maier, H. R., Jain, A., Dandy, G. C., & Sudheer, K. P. (2010). Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environmental Modelling & Software*, *25*(8), 891–909.

Merritt, W. S., Letcher, R. A., & Jakeman, A. J. (2003). A review of erosion and sediment transport models. *Environmental Modelling & Software*, *18*(8–9), 761–799. https://doi.org/10.1016/S1364-8152(03)00078-1

Mount, N. J., & Abrahart, R. J. (2011). Load or concentration, logged or unlogged? Addressing ten years of uncertainty in neural network suspended sediment prediction. *Hydrological Processes*, *25*(20), 3144–3157. https://doi.org/10.1002/hyp.8033

Mukundan, R., Pradhanang, S. M., Schneiderman, E. M., Pierson, D. C., Anandhi, A., Zion, M. S., … Steenhuis, T. S. (2013). Suspended sediment source areas and future climate impact on soil erosion and sediment yield in a New York City water supply watershed, USA. *Geomorphology*, *183*(Supplement C), 110–119. https://doi.org/10.1016/j.geomorph.2012.06.021

Mustafa, M. R., Rezaur, R. B., Saiedi, S., & Isa, M. H. (2012). River Suspended Sediment Prediction Using Various Multilayer Perceptron Neural Network Training Algorithms—A Case Study in Malaysia. *Water Resources Management*, *26*(7), 1879–1897. https://doi.org/10.1007/s11269-012-9992-5

Nourani, V., & Kalantari, O. (2010). Integrated Artificial Neural Network for Spatiotemporal Modeling of Rainfall–Runoff–Sediment Processes. *Environmental Engineering Science*, *27*(5), 411–422. https://doi.org/10.1089/ees.2009.0353

Nourani, V., Kalantari, O., & Baghanam, A. H. (2012). Two semidistributed ANN-based models for estimation of suspended sediment load. *Journal of Hydrologic Engineering*, *17*(12), 1368–1380.

Nourani, V., Parhizkar, M., Vousoughi, F., & Amini, B. (2013). Capability of Artificial Neural Network for Detecting Hysteresis Phenomenon Involved in Hydrological Processes. *Journal of Hydrologic Engineering*, *19*(5), 896–906. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000870

Olson, S. A. (2014). Estimation of Flood Discharges at Selected Annual Exceedance Probabilities for Unregulated, Rural Streams in Vermont with a section on Vermont regional skew regression, by Veilleux, A.G. (Scientific Investigations Report No. 2014–5078). Reston, VA: US Geological Survey.

Olyaie, E., Banejad, H., Chau, K.-W., & Melesse, A. M. (2015). A comparison of various artificial intelligence approaches performance for estimating suspended sediment load of river systems: a case study in United States. *Environmental Monitoring and Assessment*, *187*(4). https://doi.org/10.1007/s10661-015-4381-1

Oudin, L., Andréassian, V., Perrin, C., Michel, C., & Le Moine, N. (2008). Spatial proximity, physical similarity, regression and ungaged catchments: A comparison of regionalization approaches based on 913 French catchments: A comparison of regionalization approaches on 913 catchments. *Water Resources Research*, *44*(3), n/a-n/a. https://doi.org/10.1029/2007WR006240

Perks, M. T., Owen, G. J., Benskin, C. M. H., Jonczyk, J., Deasy, C., Burke, S., … Haygarth, P. M. (2015). Dominant mechanisms for the delivery of fine sediment and phosphorus to fluvial networks draining grassland dominated headwater catchments. *Science of The Total Environment*, *523*, 178–190. https://doi.org/10.1016/j.scitotenv.2015.03.008

Pieri, L., Poggio, M., Vignudelli, M., & Bittelli, M. (2014). Evaluation of the WEPP model and digital elevation grid size, for simulation of streamflow and sediment yield in a heterogeneous catchment. *Earth Surface Processes and Landforms*, *39*(10), 1331–1344. https://doi.org/10.1002/esp.3527

Raghuwanshi, N., Singh, R., & Reddy, L. (2006). Runoff and Sediment Yield Modeling Using Artificial Neural Networks: Upper Siwane River, India. *Journal of Hydrologic Engineering*, *11*(1), 71–79. https://doi.org/10.1061/(ASCE)1084-0699(2006)11:1(71)

Rajaee, T., Mirbagheri, S. A., Zounemat-Kermani, M., & Nourani, V. (2009). Daily suspended sediment concentration simulation using ANN and neuro-fuzzy models. *Science of The Total Environment*, *407*(17), 4916–4927. https://doi.org/10.1016/j.scitotenv.2009.05.016

Rasmussen, P. P., Gray, J. R., Glysson, G. D., & Ziegler, A. C. (2011). Guidelines and procedures for computing time-series suspended-sediment concentrations and loads from in-stream turbidity-sensor and streamflow data. In *Applications of Hydraulics*. Reston, VA: US Geological Survey.

Razavi, T., & Coulibaly, P. (2013). Streamflow Prediction in Ungauged Basins: Review of Regionalization Methods. *Journal of Hydrologic Engineering*, *18*(8), 958–975. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000690

Shiri, J., & Kişi, Ö. (2012). Estimation of Daily Suspended Sediment Load by Using Wavelet Conjunction Models. *Journal of Hydrologic Engineering*, *17*(9), 986–1000. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000535

Singh, A., Imtiyaz, M., Isaac, R. K., & Denis, D. M. (2013). Comparison of Artificial Neural Network Models for Sediment Yield Prediction at Single Gauging Station of Watershed in Eastern India. *Journal of Hydrologic Engineering*, *18*(1), 115–120. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000601

Singh, P., & Deo, M. C. (2007). Suitability of different neural networks in daily flow forecasting. *Applied Soft Computing*, *7*(3), 968–978. https://doi.org/10.1016/j.asoc.2006.05.003

Stryker, J., Wemple, B., & Bomblies, A. (2017). Modeling sediment mobilization using a distributed hydrological model coupled with a bank stability model. *Water Resources Research*, *53*(3), 2051–2073. https://doi.org/10.1002/2016WR019143

Tananaev, N. I., & Debolskiy, M. V. (2014). Turbidity observations in sediment flux studies: Examples from Russian rivers in cold environments. *Geomorphology*, *218*, 63–71. https://doi.org/10.1016/j.geomorph.2013.09.031

Uhrich, M. A., & Bragg, H. M. (2003). *Monitoring Instream Turbidity to Estimate Continuous Suspended-Sediment Loads and Yields and Clay-Water Volumes in the Upper North Santiam River Basin, Oregon, 1998–2000* (Water-Resources Investigations Report No. 03-4098). Portland, Oregon: US Geological Survey.

U.S. Environmental Protection Agency. (1999). *Method 160.2: Total Suspended Solids (TSS) (Gravimetric, Dried at 103–105 ° C)*. Washington, D.C.

US EPA. (2013). National Summary of Impaired Waters and TMDL Information. Retrieved October 14, 2013, from http://iaspub.epa.gov/waters10/attains_nation_cy.control?p_report_type=T#causes_303d

Walling, D. E., Collins, A. L., & Stroud, R. W. (2008). Tracing suspended sediment and particulate phosphorus sources in catchments. *Journal of Hydrology*, *350*(3–4), 274–289. https://doi.org/10.1016/j.jhydrol.2007.10.047

Warrick, J. A., Melack, J. M., & Goodridge, B. M. (2015). Sediment yields from small, steep coastal watersheds of California. *Journal of Hydrology: Regional Studies*, *4*, 516–534. https://doi.org/10.1016/j.ejrh.2015.08.004

Zhu, Y.-M., Lu, X. X., & Zhou, Y. (2007). Suspended sediment flux modeling with artificial neural network: An example of the Longchuanjiang River in the Upper Yangtze Catchment, China. *Geomorphology*, *84*(1–2), 111–125. https://doi.org/10.1016/j.geomorph.2006.07.010

Zounemat-Kermani, M., Kişi, Ö., Adamowski, J., & Ramezani-Charmahineh, A. (2016). Evaluation of data driven models for river suspended sediment concentration modeling. *Journal of Hydrology*, *535*, 457–472. https://doi.org/10.1016/j.jhydrol.2016.02.012

CHAPTER 6. CONCLUSIONS AND FUTURE DIRECTIONS

Context and Conclusions

The body of work presented in this dissertation was motivated, in part, by witnessing first-hand the impact that Tropical Storm Irene had on our landscape and water resources in Vermont. At that time, the legacy effects caused by the storm's geomorphic changes were expected to result in years of increased erosion and loading of sediment and sediment-bound nutrients. In Vermont and much of the Northeast, comprehensive monitoring of geomorphic change and suspended sediment flux is limited in both spatial coverage and frequency. We have attempted to mitigate these limitations through the deployment of a network of in-stream turbidity sensors, river stage gauges, water quality samplers, and meteorological stations in the Mad River Valley between 2013 and 2016. During this same period, we also conducted extensive monitoring of geomorphic change along the Mad River and select other locations using terrestrial laser scanning (TLS) and unmanned aircraft system (UAS)-based photogrammetry.

The overarching objective of this work was to help improve the monitoring and understanding of watershed sediment dynamics in temperate regions with Vermont as a setting. This dissertation focuses on the development and evaluation of new measurement tools and data-driven methods to address three major gaps in our current state of knowledge; these include an ability to:

(1) more efficiently quantify and monitor streambank erosion across large areas,

(2) infer event sediment dynamics (i.e. sources and connectivity) within a watershed from sediment and discharge monitoring at the watershed and sub-watershed outlets, and

(3) predict suspended sediment loading from local hydrometeorological data in watersheds with limited measured sediment and discharge data.

In Chapter 2, we evaluated the use of UAS-based photogrammetry for capturing the streambank topography and for monitoring bank movement. Over 20 km of the Mad River, Shepard Brook, New Haven River, and Winooski River corridors were surveyed using the UAS to test the method for monitoring over large spatial areas. Data were collected during spring, summer, and autumn conditions between spring 2015 and spring 2016. In addition, we compared the UAS data to TLS and GPS surveys at seven sites encompassing a variety of streambank conditions. To assess the accuracy of UAS-derived topography for measuring horizontal bank retreat rates or changes in ground elevation along streambank surfaces, we developed a cross-sectional analysis that compared UAS data surveys to those obtained from TLS and GPS ground survey.

The cross-sectional analysis showed that the UAS-derived data reliably captured the bank surface and compared well to TLS and GPS data. However, the accuracy depended on the density of the vegetation along the bank. We observed the highest accuracies during "leaf-off" conditions in early spring. The UAS accuracy (median vertical error of 0.11 m during optimal conditions) also matched or improved on the data resolution currently available in airborne-lidar data sets, and with the benefit of greater flexibility in the timing and location of data collection. Therefore, we concluded that UAS-based photogrammetry shows great promise for capturing bank erosion and deposition in areas where banks are not completely obscured by vegetation. In addition, based on the efficiency of data collection and post-processing, UAS-based photogrammetry offers a new tool useful for long-term (annual or multi-year) monitoring of geomorphic change along river corridors. The method presented here should be broadly applicable to a wide variety of locations and terrain as long as areas are

not completely obscured by vegetation year-round as is found in some tropical and sub-tropical climates.

In Chapter 3, we furthered the evaluation of UAS-based photogrammetry for monitoring river channel movement by taking a complimentary approach the cross-section analysis presented in Chapter 2. In Chapter 3, we analyzed bank erosion by using the automatically generated digital elevation models (DEMs) from the UAS software to determine channel changes over time. We also used existing DEMs generated form earlier airborne lidar surveys to compare UAS surveys against early river surveys. We found that that river vegetation conditions had a large impact on the applicability of UAS-based photogrammetry to capture the topography along the river corridor. At the New Haven River site, reliable estimates of volumetric bank erosion was able to be calculated both between UAS surveys as well as UAS and lidar survey. Additionally, from our 2 year UAS survey effort we were able to make recommendations for future UAS surveying efforts related to river systems as well as summarize the effort involved to collect our survey data.

In Chapter 4, a novel approach for analyzing the suspended sediment concentration-discharge (SSC-Q) relationship is presented to characterize event sediment dynamics using the Mad River watershed as the study site. To do this, we (1) collected high-frequency suspended sediment data using in-stream turbidity sensors and identified detailed patterns of hysteretic behavior in the SSC-Q relationship for more than 600^{+} storm events and then (2) developed a machine-learning method capable of classifying the hysteretic patterns. We used the 2-D hysteresis images as inputs, and adapted an approach often used for handwritten character recognition, known as a restricted Boltzmann machine (RBM), to classify images of the SSC-Q plots. We were able to successfully train and test the RBM model using 600 observed storm events from the Mad River and its tributaries.

In analyzing the hysteresis patterns in the Mad River watershed, we identified an expanded set of classification patterns. When we investigated the average hydrological and meteorological characteristics of the events associated with each type of hysteresis we were able to attribute physical processes to specific patterns. By looking at the distribution of patterns from each monitoring site we were also able to infer connectivity and proximity of sediment sources within the watershed. We believe the hysteretic behavior in the SSC-Q relationship is a convenient mechanism for compressing the complexity of sediment dynamics in time and space onto a two-dimensional image and the data-driven method presented offers a novel tool to gain additional information from hysteresis analysis. The framework described is broadly applicable to study in other regions and for analyzing the relationship of discharge to variables beyond suspended sediment such other solutes (e.g. nitrate) or hydrological variables (e.g. soil moisture); the only limitation being the necessity for high temporal resolution data and data sets large enough for use with data-driven methods.

In Chapter 5, we used a recurrent counterpropagation neural network (rCPN) to predict suspended sediment loads (SSL) using only local hydrometeorological data from the Mad River and two of its tributaries, Shepard Brook and Mill Brook. The SSL model was tested and evaluated using past records and at an ungauged location. The vast majority of artificial neural network (ANN) models for predicting SSC and SSL use past *measurements* of discharge and suspended sediment to make predictions, limiting the real-world application to one time-step ahead forecasting. In this research, we predicted SSL using only local hydrometeorological data in a two-step model that allowed prediction of both discharge and suspended sediment at locations where measurements are not available.

We utilized two rCPN models in tandem to predict SSL by first predicting discharge, and then predicting suspended sediment using the estimated discharge. We found

performance of the model in predicting discharge met or exceeded previous findings. At all locations, we also found the performance of the rCPN model to be superior to the traditional sediment rating curve method in predicting SSL. The data-driven method offers water resources managers a flexible tool for predicting suspended sediment discharge from river systems providing local hydrometeorological data exist as well a limited record of Q and TSS data existing for model training from either the study location or from a similar watershed.

Collectively, this body of research provides new tools for catchment sediment studies and demonstrates application using a comprehensive data set collected in the Mad River watershed. The research focuses on the development of practical, data-driven approaches that can leverage the increasing amounts of data available from advances in water quality sensor technology. Because of the increasing volume of spatial and temporal data, there is tremendous interest in machine learning and deep learning research in a wide variety of applications. I undertook this research with a guiding principle that machine learning methods are most effective when grounded in a solid understanding of the research question and tailored to the task at hand. As such, the aim of this research was not to develop and utilize the most state of the art machine learning algorithm, but instead to combine robust data collection and processing with data-driven tools that trade modeling process-based understanding in favor of mapping the nonlinear dynamics embedded in the desired phenomena. These data-driven, statistical-based methods sacrifice process-based understanding in favor of ease of use and the ability to make predictions over broader regional scales. I view this research as a foundation for scaling up to incorporate more sensor network data from additional locations in order to answer additional questions related to watershed studies. Below, I briefly attempt to lay out a few future directions for further study.

Future Directions

Improving data processing workflow for geomorphic change detection using UAS and digital photogrammetry

The accurate georeferencing of multiple topographic surveys collected at multiple dates and sometimes with different types of equipment was an arduous task in the measurement of streambank topography. We focused our efforts on analysis at seven specific sites and could utilize ground control points (GCPs) for accurate georerencing. The use of GCPs become impractical however in scaling up data collection and processing to long river corridors. Advances in UAS GPS receiver and photogrammetry technology are expected to allow much more accurate direct georeferencing of topographic data. While this technology is still relatively new and has yet to meet the same performance standards of GCPs, especially in the vertical direction (Hugenholtz et al. 2016), it has the potential to transform the collection of topographic data at dramatically reduced costs (Carbonneau and Dietrich 2017). An automated workflow for georeferencing UAS-derived topography over entire rivers would then allow the direct comparison of multiple survey dates or to airborne lidar surveys by using a differencing of DEMs approach. The calculation of volume of erosion and deposition along the entire river corridor at a detailed level would provide a better understanding of the rate of bank erosion.

The automation of hydrological event separation and scaling up to larger data sets

Increasingly greater availability of high frequency suspended sediment and water quality data offer many exciting opportunities (Pellerin et al. 2016). Because many sediment and nutrient processes are driven by hydrological events, understanding the sediment-discharge behavior during individual storm events is often desired. Reliable automation of identification and separation of hydrological events in time series is of critical importance in such studies. In the hysteresis analysis presented in Chapter 3, we used a manual approach in

defining storm events in favor of achieving greatest control over the definition of events. A couple recent projects by Mei and Anognastou (2015) and Tang and Carey (2017) have resulted in new Matlab functions to automatically separate out hydrological events based on rainfall-runoff records and offer one potential solution to this challenge. However, this solution requires an iterative adjustment of parameters and threshold, and it is also inflexible in the ways events are defined. There appears to be unrealized opportunity to apply machine learning approaches to identifying hydrological events.

Understanding how sediment moves in time and space through watersheds

Recent research has suggested that management of hydrological connectivity of sediment sources may be one approach to sediment-bound nutrient management strategies (e.g. Sherriff et al. 2016). Reliable characterization of sediment connectivity demands an understanding of sediment movement spatially as well as temporally within watersheds. This research provides a foundation to undertake this type of study within the Mad River watershed. Our analysis of hysteresis patterns presented in Chapter 2 identified a shift in the distribution of patterns from upstream tributaries to downstream at the Mad River outlet. In addition, discrepancies in suspended sediment yield have been observed between the Mad River and its tributary watersheds (Hamshaw 2014). Expanding the analysis of hysteresis patterns to track shifts in the sediment response from individual storm events as they move from the tributaries to the watershed outlet is an approach worth considering. Additionally, other data sets such as the ongoing sediment tracer study and measurement of geomorphic change along the main stem channel could be integrated into a comprehensive analysis of sediment sources and movement over a multi-year period.

Implementation – data collection and real-world application

A question was posed in relation to this research at its conception stage: "what can ANNs and other approaches do to inform rapid analysis of watershed management interventions?" While this body of work did not directly address this question, it offers a guiding direction for future work. When data collection is based in the local communities, it offers an opportunity to further efforts of citizen science, which has shown to be of value for watershed management (Starkey et al. 2017). The data-driven approaches to suspended sediment monitoring and prediction in this work are based on data obtained from relatively low-cost and reliable sensor technology. As such, there is unrealized potential for incorporating new technologies in data collection and analysis in citizen and community-based monitoring programs. As part of watershed pollution reduction efforts, such as the Lake Champlain TMDL Implementation Plan, communities are being asked for greater involvement in restoration strategies. Implementation of monitoring system that uses low-cost and reliable sensors and is accessible to communities through data-driven analysis tools may be a way to support their efforts in watershed management.

COMPREHENSIVE BIBLIOGRAPHY

Abban, B., A. N. Thanos Papanicolaou, M. K. Cowles, C. G. Wilson, O. Abaci, K. Wacha, K. Schilling, and D. Schnoebelen. 2016. "An Enhanced Bayesian Fingerprinting Framework for Studying Sediment Source Dynamics in Intensively Managed Landscapes." *Water Resources Research* 52 (6):4646–73. https://doi.org/10.1002/2015WR018030.

Abrahart, Robert J., François Anctil, Paulin Coulibaly, Christian W. Dawson, Nick J. Mount, Linda M. See, Asaad Y. Shamseldin, Dimitri P. Solomatine, Elena Toth, and Robert L. Wilby. 2012. "Two Decades of Anarchy? Emerging Themes and Outstanding Challenges for Neural Network River Forecasting." *Progress in Physical Geography* 36 (4):480–513. https://doi.org/10.1177/0309133312444943.

Abrahart, Robert J., L. M. See, A. J. Heppenstall, and S. M. White. 2008. "Neural Network Estimation of Suspended Sediment: Potential Pitfalls and Future Directions." In *Practical Hydroinformatics*, 139–161. Springer.

Abrahart, Robert J., and S.M. White. 2001. "Modelling Sediment Transfer in Malawi: Comparing Backpropagation Neural Network Solutions against a Multiple Linear Regression Benchmark Using Small Data Sets." *Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere* 26 (1):19–24. https://doi.org/10.1016/S1464-1909(01)85008-5.

Adeyanju, I. A., O. O. Awodoye, and E. O. Omidiora. 2016. "Performance Evaluation of an Improved Self-Organizing Feature Map and Modified Counter Propagation Network in Face Recognition." *British Journal of Mathematics & Computer Science* 14 (3):1.

Afan, Haitham Abdulmohsin, Ahmed El-shafie, Wan Hanna Melini Wan Mohtar, and Zaher Mundher Yaseen. 2016. "Past, Present and Prospect of an Artificial Intelligence (AI) Based Model for Sediment Transport Prediction." *Journal of Hydrology* 541, Part B (October):902–13. https://doi.org/10.1016/j.jhydrol.2016.07.048.

Aich, Valentin, Alexander Zimmermann, and Helmut Elsenbeer. 2014. "Quantification and Interpretation of Suspended-Sediment Discharge Hysteresis Patterns: How Much Data Do We Need?" *CATENA* 122 (November):120–29. https://doi.org/10.1016/j.catena.2014.06.020.

Aksoy, Hafzullah, and M. Levent Kavvas. 2005. "A Review of Hillslope and Watershed Scale Erosion and Sediment Transport Models." *CATENA*, 25 Years of Assessment of Erosion, 64 (2–3):247–71. https://doi.org/10.1016/j.catena.2005.08.008.

Alp, Murat, and Hikmet Kerem Cigizoglu. 2007. "Suspended Sediment Load Simulation by Two Artificial Neural Network Methods Using Hydrometeorological Data." *Environmental Modelling & Software* 22 (1):2–13. https://doi.org/10.1016/j.envsoft.2005.09.009.

ASCE. 2000a. "Artificial Neural Networks in Hydrology. I: Preliminary Concepts." *Journal of Hydrologic Engineering* 5 (2):115–23.

———. 2000b. "Artificial Neural Networks in Hydrology. II: Hydrologic Applications." *Journal of Hydrologic Engineering* 5 (2):124–37.

Asselman, Nathalie E. M. 1999. "Suspended Sediment Dynamics in a Large Drainage Basin: The River Rhine." *Hydrological Processes* 13 (10):1437–50. https://doi.org/10.1002/(SICI)1099-1085(199907)13:10<1437::AID-HYP821>3.0.CO;2-J.

Atieh, M., S. L. Mehltretter, B. Gharabaghi, and R. Rudra. 2015. "Integrative Neural Networks Model for Prediction of Sediment Rating Curve Parameters for Ungauged Basins." *Journal of Hydrology* 531 (December):1095–1107. https://doi.org/10.1016/j.jhydrol.2015.11.008.

Bakker, Maarten, and Stuart N. Lane. 2017. "Archival Photogrammetric Analysis of River–floodplain Systems Using Structure from Motion (SfM) Methods." *Earth Surface Processes and Landforms* 42 (8):1274–86. https://doi.org/10.1002/esp.4085.

Bangen, Sara G., Joseph M. Wheaton, Nicolaas Bouwes, Boyd Bouwes, and Chris Jordan. 2014. "A Methodological Intercomparison of Topographic Survey Techniques for Characterizing Wadeable Streams and Rivers." *Geomorphology* 206 (February):343–61. https://doi.org/10.1016/j.geomorph.2013.10.010.

Barg, Lori, and Mike Blazewicz. 2003. "Assessment of Fluvial Geomorphology in Relation to Erosion and Landslides in the Mad River Watershed in Central Vermont." Waterbury, VT: Vermont Geological Society.

Bauer, D.W., D.J. Mulla, and A.C. Sekely. 2002. "Streambank Slumping and Its Contribution to the Phosphorus and Suspended Sediment Loads of the Blue Earth River, Minnesota." *Journal of Soil and Water Conservation* 57:243. https://doi.org/Article.

Bayram, Adem, Murat Kankal, and Hizir Önsoy. 2011. "Estimation of Suspended Sediment Concentration from Turbidity Measurements Using Artificial Neural Networks." *Environmental Monitoring and Assessment* 184 (7):4355–65. https://doi.org/10.1007/s10661-011-2269-2.

Besaw, Lance E., Donna M. Rizzo, Paul R. Bierman, and William R. Hackett. 2010. "Advances in Ungauged Streamflow Prediction Using Artificial Neural Networks." *Journal of Hydrology* 386 (1–4):27–37. https://doi.org/10.1016/j.jhydrol.2010.02.037.

Besaw, Lance E., Donna M. Rizzo, Michael Kline, Kristen L. Underwood, Jeffrey J. Doris, Leslie A. Morrissey, and Keith Pelletier. 2009. "Stream Classification Using Hierarchical Artificial Neural Networks: A Fluvial Hazard Management Tool." *Journal of Hydrology* 373 (1–2):34–43. https://doi.org/10.1016/j.jhydrol.2009.04.007.

Betts, Alan K. 2012. "Historic Trends and Future Climatic Projections for Vermont." presented at the Vulnerability Assessment Workshop, Montpelier, VT, July 9.

Borah, Deva K., and Maitreyee Bera. 2004. "Watershed-Scale Hydrologic and Nonpoint-Source Pollution Models: Review of Applications." *Transactions of the ASAE* 47 (3):789–803.

Borg, Jaron, Mandar M. Dewoolkar, and Paul Bierman. 2014. "Assessment of Streambank Stability-A Case Study." In *Geo-Congress 2014 Technical Papers*, 1007–16. Atlanta, Georgia: American Society of Civil Engineers. https://doi.org/10.1061/9780784413272.098.

Brasington, J., D. Vericat, and I. Rychkov. 2012. "Modeling River Bed Morphology, Roughness, and Surface Sedimentology Using High Resolution Terrestrial Laser Scanning." *Water Resources Research* 48 (11):W11519. https://doi.org/10.1029/2012WR012223.

Bremer, Magnus, and Oliver Sass. 2012. "Combining Airborne and Terrestrial Laser Scanning for Quantifying Erosion and Deposition by a Debris Flow Event." *Geomorphology* 138 (1):49–60. https://doi.org/10.1016/j.geomorph.2011.08.024.

Brodu, N., and D. Lague. 2012. "3D Terrestrial Lidar Data Classification of Complex Natural Scenes Using a Multi-Scale Dimensionality Criterion: Applications in Geomorphology." *ISPRS Journal of Photogrammetry and Remote Sensing* 68 (March):121–34. https://doi.org/10.1016/j.isprsjprs.2012.01.006.

Buchanan, B. P., M. T. Walter, G. N. Nagle, and R. L. Schneider. 2012. "Monitoring and Assessment of a River Restoration Project in Central New York." *River Research and Applications* 28 (2):216–33. https://doi.org/10.1002/rra.1453.

Buendia, Cristina, Damià Vericat, Ramon J. Batalla, and Chris N. Gibbins. 2015. "Temporal Dynamics of Sediment Transport and Transient in-Channel Storage in a Highly Erodible Catchment." *Land Degradation & Development*, March, n/a-n/a. https://doi.org/10.1002/ldr.2348.

Bussi, Gianbattista, Félix Francés, Enguerrand Horel, José Andrés López-Tarazón, and Ramón J. Batalla. 2014. "Modelling the Impact of Climate Change on Sediment Yield in a Highly Erodible Mediterranean Catchment." *Journal of Soils and Sediments* 14 (12):1921–37. https://doi.org/10.1007/s11368-014-0956-7.

Buyukyildiz, Meral, and Serife Yurdagul Kumcu. 2017. "An Estimation of the Suspended Sediment Load Using Adaptive Network Based Fuzzy Inference System, Support Vector Machine and Artificial Neural Network Models." *Water Resources Management* 31 (4):1343–59. https://doi.org/10.1007/s11269-017-1581-1.

Carbonneau, Patrice E., and James T. Dietrich. 2017. "Cost-Effective Non-Metric Photogrammetry from Consumer-Grade sUAS: Implications for Direct Georeferencing of Structure from Motion Photogrammetry." *Earth Surface Processes and Landforms* 42 (3):473–86. https://doi.org/10.1002/esp.4012.

Cavalli, Marco, Beatrice Goldin, Francesco Comiti, Francesco Brardinoni, and Lorenzo Marchi. 2017. "Assessment of Erosion and Deposition in Steep Mountain Basins by Differencing Sequential Digital Terrain Models." *Geomorphology*, SEDIMENT DYNAMICS IN ALPINE BASINS, 291 (August):4–16. https://doi.org/10.1016/j.geomorph.2016.04.009.

Chang, Fi-John, and Yen-Chang Chen. 2001. "A Counterpropagation Fuzzy-Neural Network Modeling Approach to Real Time Streamflow Prediction." *Journal of Hydrology* 245 (1–4):153–64. https://doi.org/10.1016/S0022-1694(01)00350-X.

Chang, F.-John, Li-Chiu Chang, and Hau-Lung Huang. 2002. "Real-Time Recurrent Learning Neural Network for Stream-Flow Forecasting." *Hydrological Processes* 16 (13):2577–2588. https://doi.org/10.1002/hyp.1015.

Chea, Ratha, Gaël Grenouillet, and Sovan Lek. 2016. "Evidence of Water Quality Degradation in Lower Mekong Basin Revealed by Self-Organizing Map." *PLOS ONE* 11 (1):e0145527. https://doi.org/10.1371/journal.pone.0145527.

Chen, Dong, and Jennifer G. Duan. 2006. "Modeling Channel Morphologic Change in the West Jordan River, Utah." In *World Environmental and Water Resource Congress 2006*, 1–10. Omaha, Nebraska: American Society of Civil Engineers. https://doi.org/10.1061/40856(200)63.

Cheraghi, Mohsen, Seifeddine Jomaa, Graham C. Sander, and D. A. Barry. 2016. "Hysteretic Sediment Fluxes in Rainfall-Driven Soil Erosion: Particle Size Effects." *Water Resources Research* 52 (11):8613–29. https://doi.org/10.1002/2016WR019314.

Cigizoglu, Hikmet Kerem, and Murat Alp. 2006. "Generalized Regression Neural Network in Modelling River Sediment Yield." *Advances in Engineering Software* 37 (2):63–68. https://doi.org/10.1016/j.advengsoft.2005.05.002.

Cigizoglu, Hikmet Kerem, and Özgür Kisi. 2006. "Methods to Improve the Neural Network Performance in Suspended Sediment Estimation." *Journal of Hydrology* 317 (3–4):221–38. https://doi.org/10.1016/j.jhydrol.2005.05.019.

Cobaner, M., B. Unal, and O. Kisi. 2009. "Suspended Sediment Concentration Estimation by an Adaptive Neuro-Fuzzy and Neural Network Approaches Using Hydro-Meteorological Data." *Journal of Hydrology* 367 (1–2):52–61. https://doi.org/10.1016/j.jhydrol.2008.12.024.

Collins, A. L., and D. E. Walling. 2004. "Documenting Catchment Suspended Sediment Sources: Problems, Approaches and Prospects." *Progress in Physical Geography* 28 (2):159–96. https://doi.org/10.1191/0309133304pp409ra.

Colomina, I., and P. Molina. 2014. "Unmanned Aerial Systems for Photogrammetry and Remote Sensing: A Review." *ISPRS Journal of Photogrammetry and Remote Sensing* 92 (June):79–97. https://doi.org/10.1016/j.isprsjprs.2014.02.013.

Cook, Aaron, and Venkatesh Merwade. 2009. "Effect of Topographic Data, Geometric Configuration and Modeling Approach on Flood Inundation Mapping." *Journal of Hydrology* 377 (1–2):131–42. https://doi.org/10.1016/j.jhydrol.2009.08.015.

Cook, Kristen L. 2017. "An Evaluation of the Effectiveness of Low-Cost UAVs and Structure from Motion for Geomorphic Change Detection." *Geomorphology* 278 (February):195–208. https://doi.org/10.1016/j.geomorph.2016.11.009.

Coulibaly, P., and N. D. Evora. 2007. "Comparison of Neural Network Methods for Infilling Missing Daily Weather Records." *Journal of Hydrology* 341 (1):27–41. https://doi.org/10.1016/j.jhydrol.2007.04.020.

Daly, Erin R., Ronald B. Miller, and Garey A. Fox. 2015. "Modeling Streambank Erosion and Failure along Protected and Unprotected Composite Streambanks." *Advances in Water Resources* 81 (July):114–27. https://doi.org/10.1016/j.advwatres.2015.01.004.

Dapporto, Stefano, Massimo Rinaldi, Nicola Casagli, and Pietro Vannocci. 2003. "Mechanisms of Riverbank Failure along the Arno River, Central Italy." *Earth Surface Processes and Landforms* 28 (12):1303–23. https://doi.org/10.1002/esp.550.

Darby, Stephen E., and Colin R. Thorne. 1996. "Development and Testing of Riverbank-Stability Analysis." *Journal of Hydraulic Engineering* 122 (8):443–54. https://doi.org/10.1061/(ASCE)0733-9429(1996)122:8(443).

Davis, C., and J. Fox. 2009. "Sediment Fingerprinting: Review of the Method and Future Improvements for Allocating Nonpoint Source Pollution." *Journal of Environmental Engineering* 135 (7):490–504. https://doi.org/10.1061/(ASCE)0733-9372(2009)135:7(490).

De Castro, Leandro N. 2006. *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*. Chapman & Hall/CRC Computer and Information Science Series. Boca Raton: Chapman & Hall/CRC.

De Rose, Ronald C., and Les R. Basher. 2011. "Measurement of River Bank and Cliff Erosion from Sequential LIDAR and Historical Aerial Photography." *Geomorphology* 126 (1–2):132–47. https://doi.org/10.1016/j.geomorph.2010.10.037.

Dharia, Abhijit, and Hojjat Adeli. 2003. "Neural Network Model for Rapid Forecasting of Freeway Link Travel Time." *Engineering Applications of Artificial Intelligence* 16 (7):607–13. https://doi.org/10.1016/j.engappai.2003.09.011.

Dietrich, James T. 2017. "Bathymetric Structure-from-Motion: Extracting Shallow Stream Bathymetry from Multi-View Stereo Photogrammetry." *Earth Surface Processes and Landforms* 42 (2):355–64. https://doi.org/10.1002/esp.4060.

Drgan, Viktor, Špela Župerl, Marjan Vračko, Claudia Ileana Cappelli, and Marjana Novič. 2017. "CPANNatNIC Software for Counter-Propagation Neural Network to Assist in Read-Across." *Journal of Cheminformatics* 9 (May):30. https://doi.org/10.1186/s13321-017-0218-y.

Duvert, Clément, Nicolas Gratiot, Olivier Evrard, Oldrich Navratil, Julien Némery, Christian Prat, and Michel Esteves. 2010. "Drivers of Erosion and Suspended Sediment Transport in Three Headwater Catchments of the Mexican Central Highlands." *Geomorphology* 123 (3–4):243–56. https://doi.org/10.1016/j.geomorph.2010.07.016.

Eaton, Brett C. 2006. "Bank Stability Analysis for Regime Models of Vegetated Gravel Bed Rivers." *Earth Surface Processes and Landforms* 31 (11):1438–44. https://doi.org/10.1002/esp.1364.

Eder, A., P. Strauss, T. Krueger, and J.N. Quinton. 2010. "Comparative Calculation of Suspended Sediment Loads with Respect to Hysteresis Effects (in the Petzenkirchen Catchment, Austria)." *Journal of Hydrology* 389 (1–2):168–76. https://doi.org/10.1016/j.jhydrol.2010.05.043.

Edwards, T. K., and G. D. Glysson. 1999. *Field Methods for Measurement of Fluvial Sediment: US Geological Survey Techniques of Water-Resources Investigations*. US Geological Survey.

Eltner, Anette, Andreas Kaiser, Antonio Abellan, and Marcus Schindewolf. 2017. "Time Lapse Structure from Motion Photogrammetry for Continuous Geomorphic Monitoring." *Earth Surface Processes and Landforms*. https://doi.org/10.1002/esp.4178.

Evans, Christopher, and Trevor D. Davies. 1998. "Causes of Concentration/Discharge Hysteresis and Its Potential as a Tool for Analysis of Episode Hydrochemistry." *Water Resources Research* 34 (1):129–37. https://doi.org/10.1029/97WR01881.

Evans, D. J., C. E. Gibson, and R. S. Rossell. 2006. "Sediment Loads and Sources in Heavily Modified Irish Catchments: A Move towards Informed Management Strategies." *Geomorphology*, Sediment and Geochemical BudgetsPapers in honour of Professor Olav Slaymaker, 79 (1–2):93–113. https://doi.org/10.1016/j.geomorph.2005.09.018.

Firat, M. 2008. "Comparison of Artificial Intelligence Techniques for River Flow Forecasting." *Hydrology and Earth System Sciences* 12 (1):123–139.

Fischer, Asja, and Christian Igel. 2014. "Training Restricted Boltzmann Machines: An Introduction." *Pattern Recognition* 47 (1):25–39. https://doi.org/10.1016/j.patcog.2013.05.025.

Fitzgerald, Evan P., and Lisa C. Godfrey. 2008. "Upper Mad River Corridor Plan." Waitsfield, VT: Friends of the Mad River.

Flener, Claude, Matti Vaaja, Anttoni Jaakkola, Anssi Krooks, Harri Kaartinen, Antero Kukko, Elina Kasvi, Hannu Hyyppä, Juha Hyyppä, and Petteri Alho. 2013. "Seamless Mapping of River Channels at High Resolution Using Mobile LiDAR and UAV-Photography." *Remote Sensing* 5 (12):6382–6407. https://doi.org/10.3390/rs5126382.

Foucher, Anthony, Sébastien Salvador-Blanes, Rosalie Vandromme, Olivier Cerdan, and Marc Desmet. 2017. "Quantification of Bank Erosion in a Drained Agricultural Lowland Catchment." *Hydrological Processes* 31 (6):1424–37. https://doi.org/10.1002/hyp.11117.

Fox, Garey A., Glenn V. Wilson, Andrew Simon, Eddy J. Langendoen, Onur Akay, and John W. Fuchs. 2007. "Measuring Streambank Erosion due to Ground Water Seepage: Correlation to Bank Pore Water Pressure, Precipitation and Stream Stage." *Earth Surface Processes and Landforms* 32 (10):1558–73. https://doi.org/10.1002/esp.1490.

Fryirs, Kirstie. 2013. "(Dis)Connectivity in Catchment Sediment Cascades: A Fresh Look at the Sediment Delivery Problem." *Earth Surface Processes and Landforms* 38 (1):30–46. https://doi.org/10.1002/esp.3242.

Gao, Peng. 2008. "Understanding Watershed Suspended Sediment Transport." *Progress in Physical Geography* 32 (3):243–63. https://doi.org/10.1177/0309133308094849.

Gao, Peng, and Maria Josefson. 2012. "Event-Based Suspended Sediment Dynamics in a Central New York Watershed." *Geomorphology* 139–140 (February):425–37. https://doi.org/10.1016/j.geomorph.2011.11.007.

Garvey, Kerrie M. 2012. "Quantifying Erosion and Deposition Due to Stream Planform Change Using High Spatial Resolution Digital Orthophotography and Lidar Data." M.S. Thesis, Burlington, VT: University of Vermont.

Gellis, A. C. 2013. "Factors Influencing Storm-Generated Suspended-Sediment Concentrations and Loads in Four Basins of Contrasting Land Use, Humid-Tropical Puerto Rico." *CATENA* 104 (May):39–57. https://doi.org/10.1016/j.catena.2012.10.018.

Gellis, Allen C., and Rajith Mukundan. 2013. "Watershed Sediment Source Identification: Tools, Approaches, and Case Studies." *Journal of Soils and Sediments* 13 (10):1655–57. https://doi.org/10.1007/s11368-013-0778-z.

George, Lance, and Jann Yankausas. 2011. "Preserving Affordable Manufactured Home Communities in Rural America: A Case Study." Housing Assistance Council.

Gettel, Melissa, John S. Gulliver, Masoud Kayhanian, Gregory DeGroot, Joshua Brand, Omid Mohseni, and Andrew J. Erickson. 2011. "Improving Suspended Sediment Measurements by Automatic Samplers." *Journal of Environmental Monitoring* 13 (10):2703–9. https://doi.org/10.1039/C1EM10258C.

Gomez, B., Y. Cui, A.J. Kettner, D.H. Peacock, and J.P.M. Syvitski. 2009. "Simulating Changes to the Sediment Transport Regime of the Waipaoa River, New Zealand, Driven by Climate Change in the Twenty-First Century." *Global and Planetary Change* 67 (3–4):153–66. https://doi.org/10.1016/j.gloplacha.2009.02.002.

Goode, Jaime R., Charles H. Luce, and John M. Buffington. 2012. "Enhanced Sediment Delivery in a Changing Climate in Semi-Arid Mountain Basins: Implications for Water Resource Management and Aquatic Habitat in the Northern Rocky Mountains." *Geomorphology* 139–140 (February):1–15. https://doi.org/10.1016/j.geomorph.2011.06.021.

Govindaraju, Rao S., and A. Ramachandra Rao. 2000. *Artificial Neural Networks in Hydrology*. Water Science and Technology Library, v. 36. Dordrecht Boston, MA: Kluwer Academic Publishers.

Gray, John R., and Jeffrey W. Gartner. 2009. "Technological Advances in Suspended-Sediment Surrogate Monitoring." *Water Resources Research* 45 (4):W00D29. https://doi.org/10.1029/2008WR007063.

Grove, James R., Jacky Croke, and Christopher Thompson. 2013. "Quantifying Different Riverbank Erosion Processes during an Extreme Flood Event." *Earth Surface Processes and Landforms* 38 (12):1393–1406. https://doi.org/10.1002/esp.3386.

Hamshaw, Scott D. 2014. "Suspended Sediment Prediction Using Artificial Neural Networks and Local Hydrometeorological Data." M.S. Thesis, Burlington, VT: University of Vermont.

Hamshaw, Scott D., Thomas Bryce, Donna M. Rizzo, Jarlath O'Neil-Dunne, Jeff Frolik, and Mandar M. Dewoolkar. 2017. "Quantifying Streambank Movement and Topography Using Unmanned Aircraft System Photogrammetry with Comparison to Terrestrial Laser Scanning." *River Research and Applications* 33 (8):1354–67. https://doi.org/10.1002/rra.3183.

Han, D., and M. Bray. 2006. "Automated Thiessen Polygon Generation." *Water Resources Research* 42 (11):W11502. https://doi.org/10.1029/2005WR004365.

Harrington, Seán T., and Joseph R. Harrington. 2013. "An Assessment of the Suspended Sediment Rating Curve Approach for Load Estimation on the Rivers Bandon and Owenabue, Ireland." *Geomorphology* 185 (March):27–38. https://doi.org/10.1016/j.geomorph.2012.12.002.

Hecht-Nielsen, Robert. 1988. "Applications of Counterpropagation Networks." *Neural Networks* 1 (2):131–39. https://doi.org/10.1016/0893-6080(88)90015-9.

Heng, Sokchhay, and Tadashi Suetsugi. 2013. "Using Artificial Neural Network to Estimate Sediment Load in Ungauged Catchments of the Tonle Sap River Basin, Cambodia." *Journal of Water Resource and Protection* 5 (2):111–23. https://doi.org/10.4236/jwarp.2013.52013.

Heritage, George L., and David J. Milan. 2009. "Terrestrial Laser Scanning of Grain Roughness in a Gravel-Bed River." *Geomorphology* 113 (1–2):4–11. https://doi.org/10.1016/j.geomorph.2009.03.021.

Hinton, Geoffrey E. 2012. "A Practical Guide to Training Restricted Boltzmann Machines." In *Neural Networks: Tricks of the Trade*, edited by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, 599–619. Lecture Notes in Computer Science 7700. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-35289-8_32.

Hinton, Geoffrey E., L. Deng, D. Yu, G. E. Dahl, A. r Mohamed, N. Jaitly, A. Senior, et al. 2012. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups." *IEEE Signal Processing Magazine* 29 (6):82–97. https://doi.org/10.1109/MSP.2012.2205597.

Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. 2006. "A Fast Learning Algorithm for Deep Belief Nets." *Neural Computation* 18 (7):1527–1554.

Hinton, Geoffrey E., and R. R. Salakhutdinov. 2006. "Reducing the Dimensionality of Data with Neural Networks." *Science* 313 (5786):504–7. https://doi.org/10.1126/science.1127647.

Hodge, Rebecca, James Brasington, and Keith Richards. 2009. "Analysing Laser-Scanned Digital Terrain Models of Gravel Bed Surfaces: Linking Morphology to Sediment Transport Processes and Hydraulics." *Sedimentology* 56 (7):2024–43. https://doi.org/10.1111/j.1365-3091.2009.01068.x.

Hohenthal, J., P. Alho, J. Hyyppa, and H. Hyyppa. 2011. "Laser Scanning Applications in Fluvial Studies." *Progress in Physical Geography* 35 (6):782–809. https://doi.org/10.1177/0309133311414605.

Horowitz, Arthur J. 2003. "An Evaluation of Sediment Rating Curves for Estimating Suspended Sediment Concentrations for Subsequent Flux Calculations." *Hydrological Processes* 17 (17):3387–3409. https://doi.org/10.1002/hyp.1299.

Hu, T. S., K. C. Lam, and S. T. Ng. 2001. "River Flow Time Series Prediction with a Range-Dependent Neural Network." *Hydrological Sciences Journal* 46 (5):729–45. https://doi.org/10.1080/02626660109492867.

Hugenholtz, Chris, Owen Brown, Jordan Walker, Thomas Barchyn, Paul Nesbit, Maja Kucharczyk, and Steve Myshak. 2016. "Spatial Accuracy of UAV-Derived Orthoimagery and Topography: Comparing Photogrammetric Models Processed with Direct Geo-Referencing and Ground Control Points." *GEOMATICA* 70 (1):21–30. https://doi.org/10.5623/cig2016-102.

Hugenholtz, Chris H., Ken Whitehead, Owen W. Brown, Thomas E. Barchyn, Brian J. Moorman, Adam LeClair, Kevin Riddell, and Tayler Hamilton. 2013. "Geomorphological Mapping with a Small Unmanned Aircraft System (sUAS): Feature Detection and Accuracy Assessment of a Photogrammetrically-Derived Digital Terrain Model." *Geomorphology* 194 (July):16–24. https://doi.org/10.1016/j.geomorph.2013.03.023.

Hughes, Michael L., Patricia F. McDowell, and W. Andrew Marcus. 2006. "Accuracy Assessment of Georectified Aerial Photographs: Implications for Measuring Lateral Channel Movement in a GIS." *Geomorphology* 74 (1–4):1–16. https://doi.org/10.1016/j.geomorph.2005.07.001.

Jaboyedoff, Michel, Denis Demers, Jacques Locat, Ariane Locat, Pascal Locat, Thierry Oppikofer, Denis Robitaille, and Dominique Turmel. 2009. "Use of Terrestrial Laser Scanning for the Characterization of Retrogressive Landslides in Sensitive Clay and Rotational Landslides in River Banks." *Canadian Geotechnical Journal* 46 (12):1379–90. https://doi.org/10.1139/T09-073.

Jain, Sharad Kumar. 2001. "Development of Integrated Sediment Rating Curves Using ANNs." *Journal of Hydraulic Engineering* 127 (1):30–37.

James, M. R., and S. Robson. 2012. "Straightforward Reconstruction of 3D Surfaces and Topography with a Camera: Accuracy and Geoscience Application." *Journal of Geophysical Research: Earth Surface* 117 (F3):F03017. https://doi.org/10.1029/2011JF002289.

James, Mike R., Stuart Robson, and Mark W. Smith. 2017. "3-D Uncertainty-Based Topographic Change Detection with Structure-from-Motion Photogrammetry: Precision Maps for Ground Control and Directly Georeferenced Surveys." *Earth Surface Processes and Landforms*, January. https://doi.org/10.1002/esp.4125.

Javernick, L., J. Brasington, and B. Caruso. 2014. "Modeling the Topography of Shallow Braided Rivers Using Structure-from-Motion Photogrammetry." *Geomorphology* 213 (May):166–82. https://doi.org/10.1016/j.geomorph.2014.01.006.

Joshi, Rajesh, Kireet Kumar, and Vijay Pal Singh Adhikari. 2016. "Modelling Suspended Sediment Concentration Using Artificial Neural Networks for Gangotri Glacier." *Hydrological Processes* 30 (9):1354–66. https://doi.org/10.1002/hyp.10723.

Kakaei Lafdani, E., A. Moghaddam Nia, and A. Ahmadi. 2013. "Daily Suspended Sediment Load Prediction Using Artificial Neural Networks and Support Vector Machines." *Journal of Hydrology* 478 (January):50–62. https://doi.org/10.1016/j.jhydrol.2012.11.048.

Kamel, Khanchoul, Tourki Mahmoud, Yves Bissonnais, and Tourki Mahmoud. 2014. "Assessment of the Artificial Neural Networks to Geomorphic Modelling of Sediment Yield for Ungauged Catchments, Algeria." *Journal of Urban and Environmental Engineering* 8 (2):175–85. https://doi.org/10.4090/juee.2014.v8n2.175185.

Kisi, Ozgur. 2005. "Suspended Sediment Estimation Using Neuro-Fuzzy and Neural Network Approaches." *Hydrological Sciences Journal* 50 (4). https://doi.org/10.1623/hysj.2005.50.4.683.

Kisi, Ozgur, and Jalal Shiri. 2012. "River Suspended Sediment Estimation by Climatic Variables Implication: Comparative Study among Soft Computing Techniques." *Computers & Geosciences* 43 (June):73–82. https://doi.org/10.1016/j.cageo.2012.02.007.

Kline, Michael, and Barry Cahoon. 2010. "Protecting River Corridors in Vermont1." *JAWRA Journal of the American Water Resources Association* 46 (2):227–236. https://doi.org/10.1111/j.1752-1688.2010.00417.x.

Krueger, T., J. N. Quinton, J. Freer, C. J. A. Macleod, G. S. Bilotta, R. E. Brazier, J. M. B. Hawkins, and P. M. Haygarth. 2012. "Comparing Empirical Models for Sediment and Phosphorus Transfer from Soils to Water at Field and Catchment Scale under Data Uncertainty." *European Journal of Soil Science* 63 (2):211–23. https://doi.org/10.1111/j.1365-2389.2011.01419.x.

Kumar, Bimlesh. 2012. "Neural Network Prediction of Bed Material Load Transport." *Hydrological Sciences Journal* 57 (5):956–66. https://doi.org/10.1080/02626667.2012.687108.

Kumar, Dheeraj, Ashish Pandey, Nayan Sharma, and Wolfgang-Albert Flügel. 2015. "Modeling Suspended Sediment Using Artificial Neural Networks and TRMM-3B42 Version 7 Rainfall Dataset." *Journal of Hydrologic Engineering* 20 (6):C4014007. https://doi.org/10.1061/(ASCE)HE.1943-5584.0001082.

———. 2016. "Daily Suspended Sediment Simulation Using Machine Learning Approach." *CATENA* 138 (March):77–90. https://doi.org/10.1016/j.catena.2015.11.013.

Lague, Dimitri, Nicolas Brodu, and Jérôme Leroux. 2013. "Accurate 3D Comparison of Complex Topography with Terrestrial Laser Scanner: Application to the Rangitikei Canyon (N-Z)." *ISPRS*

Journal of Photogrammetry and Remote Sensing 82 (August):10–26. https://doi.org/10.1016/j.isprsjprs.2013.04.009.

Lai, Yong G., Robert E. Thomas, Yavuz Ozeren, Andrew Simon, Blair P. Greimann, and Kuowei Wu. 2015. "Modeling of Multilayer Cohesive Bank Erosion with a Coupled Bank Stability and Mobile-Bed Model." *Geomorphology* 243 (August):116–29. https://doi.org/10.1016/j.geomorph.2014.07.017.

Landers, Mark N., and Terry Sturm. 2013. "Hysteresis in Suspended Sediment to Turbidity Relations due to Changing Particle Size Distributions." *Water Resources Research*, July, n/a-n/a. https://doi.org/10.1002/wrcr.20394.

Langendoen, Eddy J., Andrew Simon, Lauren Klimetz, Bankhead Natasha, and Michael E. Ursic. 2012. "Quantifying Sediment Loadings from Streambank Erosion in Selected Agricultural Watersheds Draining to Lake Champlain." Technical Report 72. Grand Isle, VT: Lake Champlain Basin Program.

Langlois, Jacques L., Dale W. Johnson, and Guy R. Mehuys. 2005. "Suspended Sediment Dynamics Associated with Snowmelt Runoff in a Small Mountain Stream of Lake Tahoe (Nevada)." *Hydrological Processes* 19 (18):3569–80. https://doi.org/10.1002/hyp.5844.

Larochelle, Hugo, Michael Mandel, Razvan Pascanu, and Yoshua Bengio. 2012. "Learning Algorithms for the Classification Restricted Boltzmann Machine." *Journal of Machine Learning Research* 13 (Mar):643–669.

Lawler, D. M. 1993. "The Measurement of River Bank Erosion and Lateral Channel Change: A Review." *Earth Surface Processes and Landforms* 18 (9):777–821. https://doi.org/10.1002/esp.3290180905.

Lawler, D. M., J. R. Grove, J. S. Couperthwaite, and G. J. L. Leeks. 1999. "Downstream Change in River Bank Erosion Rates in the Swale-Ouse System, Northern England." *Hydrological Processes* 13 (7):977–92. https://doi.org/10.1002/(SICI)1099-1085(199905)13:7<977::AID-HYP785>3.0.CO;2-5.

Lawler, D. M., G. E. Petts, I. D. L. Foster, and S. Harper. 2006. "Turbidity Dynamics during Spring Storm Events in an Urban Headwater River System: The Upper Tame, West Midlands, UK." *Science of The Total Environment*, Urban Environmental Research in the UK: The Urban Regeneration and the Environment (NERC URGENT) Programme and associated studies., 360 (1–3):109–26. https://doi.org/10.1016/j.scitotenv.2005.08.032.

Lefrançois, J., C. Grimaldi, C. Gascuel-Odoux, and N. Gilliet. 2007. "Suspended Sediment and Discharge Relationships to Identify Bank Degradation as a Main Sediment Source on Small Agricultural Catchments." *Hydrological Processes* 21 (21):2923–33. https://doi.org/10.1002/hyp.6509.

Lejot, J., C. Delacourt, H. Piégay, T. Fournier, M-L. Trémélo, and P. Allemand. 2007. "Very High Spatial Resolution Imagery for Channel Bathymetry and Topography from an Unmanned Mapping Controlled Platform." *Earth Surface Processes and Landforms* 32 (11):1705–25. https://doi.org/10.1002/esp.1595.

Liu, Qian-Jin, Zhi-Hua Shi, Nu-Fang Fang, Hua-De Zhu, and Lei Ai. 2013. "Modeling the Daily Suspended Sediment Concentration in a Hyperconcentrated River on the Loess Plateau, China, Using the Wavelet–ANN Approach." *Geomorphology* 186 (March):181–90. https://doi.org/10.1016/j.geomorph.2013.01.012.

Lloyd, C. E. M., J. E. Freer, P. J. Johnes, and A. L. Collins. 2016a. "Using Hysteresis Analysis of High-Resolution Water Quality Monitoring Data, Including Uncertainty, to Infer Controls on Nutrient and Sediment Transfer in Catchments." *Science of The Total Environment* 543, Part A (February):388–404. https://doi.org/10.1016/j.scitotenv.2015.11.028.

———. 2016b. "Technical Note: Testing an Improved Index for Analysing Storm Discharge-Concentration Hysteresis." *Hydrology and Earth System Sciences* 20 (2):625–32. https://doi.org/10.5194/hess-20-625-2016.

Maier, Holger R., Ashu Jain, Graeme C. Dandy, and K. PKPS Sudheer. 2010. "Methods Used for the Development of Neural Networks for the Prediction of Water Resource Variables in River Systems: Current Status and Future Directions." *Environmental Modelling & Software* 25 (8):891–909.

Martin, Sarah, Martha Conklin, and Roger Bales. 2014. "Seasonal Accumulation and Depletion of Local Sediment Stores of Four Headwater Catchments." *Water* 6 (7):2144–63. https://doi.org/10.3390/w6072144.

Megnounif, Abdesselam, Abdelali Terfous, and Sylvain Ouillon. 2013. "A Graphical Method to Study Suspended Sediment Dynamics during Flood Events in the Wadi Sebdou, NW Algeria (1973–2004)." *Journal of Hydrology* 497 (August):24–36. https://doi.org/10.1016/j.jhydrol.2013.05.029.

Mei, Yiwen, and Emmanouil N. Anagnostou. 2015. "A Hydrograph Separation Method Based on Information from Rainfall and Runoff Records." *Journal of Hydrology* 523 (April):636–49. https://doi.org/10.1016/j.jhydrol.2015.01.083.

Merritt, W.S., R.A. Letcher, and A.J. Jakeman. 2003. "A Review of Erosion and Sediment Transport Models." *Environmental Modelling & Software* 18 (8–9):761–99. https://doi.org/10.1016/S1364-8152(03)00078-1.

Micheletti, Natan, Jim H. Chandler, and Stuart N. Lane. 2015. "Investigating the Geomorphological Potential of Freely Available and Accessible Structure-from-Motion Photogrammetry Using a Smartphone." *Earth Surface Processes and Landforms* 40 (4):473–86. https://doi.org/10.1002/esp.3648.

Milan, David J., George L. Heritage, and David Hetherington. 2007. "Application of a 3D Laser Scanner in the Assessment of Erosion and Deposition Volumes and Channel Change in a Proglacial River." *Earth Surface Processes and Landforms* 32 (11):1657–1674. https://doi.org/10.1002/esp.1592.

Milan, D.j., G.l. Heritage, A.r.g. Large, and N.s. Entwistle. 2010. "Mapping Hydraulic Biotopes Using Terrestrial Laser Scan Data of Water Surface Properties." *Earth Surface Processes and Landforms* 35 (8):918–931. https://doi.org/10.1002/esp.1948.

Miřijovský, Jakub, and Jakub Langhammer. 2015. "Multitemporal Monitoring of the Morphodynamics of a Mid-Mountain Stream Using UAS Photogrammetry." *Remote Sensing* 7 (7):8586–8609. https://doi.org/10.3390/rs70708586.

Miřijovský, Jakub, Monika Šulc Michalková, Otmar Petyniak, Zdeněk Máčka, and Milan Trizna. 2015. "Spatiotemporal Evolution of a Unique Preserved Meandering System in Central Europe — The Morava River near Litovel." *CATENA* 127 (April):300–311. https://doi.org/10.1016/j.catena.2014.12.006.

Molina, José-Luis, Pablo Rodríguez-Gonzálvez, Mª Carmen Molina, Diego González-Aguilera, and Fernando Espejo. 2014. "Geomatic Methods at the Service of Water Resources Modelling." *Journal of Hydrology* 509 (February):150–62. https://doi.org/10.1016/j.jhydrol.2013.11.034.

Mount, Nick J., and Robert J. Abrahart. 2011. "Load or Concentration, Logged or Unlogged? Addressing Ten Years of Uncertainty in Neural Network Suspended Sediment Prediction." *Hydrological Processes* 25 (20):3144–3157. https://doi.org/10.1002/hyp.8033.

Mount, Nick J., and Tim Stott. 2008. "A Discrete Bayesian Network to Investigate Suspended Sediment Concentrations in an Alpine Proglacial Zone." *Hydrological Processes* 22 (18):3772–84. https://doi.org/10.1002/hyp.6981.

Mukundan, Rajith, Soni M. Pradhanang, Elliot M. Schneiderman, Donald C. Pierson, Aavudai Anandhi, Mark S. Zion, Adão H. Matonse, David G. Lounsbury, and Tammo S. Steenhuis. 2013. "Suspended Sediment Source Areas and Future Climate Impact on Soil Erosion and Sediment Yield in a New York City Water Supply Watershed, USA." *Geomorphology*, Studying water-erosion processes with geoinformatics, 183 (Supplement C):110–19. https://doi.org/10.1016/j.geomorph.2012.06.021.

Mustafa, M. R., R. B. Rezaur, S. Saiedi, and M. H. Isa. 2012. "River Suspended Sediment Prediction Using Various Multilayer Perceptron Neural Network Training Algorithms—A Case Study in Malaysia." *Water Resources Management* 26 (7):1879–97. https://doi.org/10.1007/s11269-012-9992-5.

Nagy, H. M., K. Watanabe, and M. Hirano. 2002. "Prediction of Sediment Load Concentration in Rivers Using Artificial Neural Network Model." *Journal of Hydraulic Engineering* 128:588.

Nathan, R. J., and T. A. McMahon. 1990. "Evaluation of Automated Techniques for Base Flow and Recession Analyses." *Water Resources Research* 26 (7):1465–73. https://doi.org/10.1029/WR026i007p01465.

Neugirg, F., M. Stark, A. Kaiser, M. Vlacilova, M. Della Seta, F. Vergari, J. Schmidt, M. Becht, and F. Haas. 2016. "Erosion Processes in Calanchi in the Upper Orcia Valley, Southern Tuscany, Italy Based on Multitemporal High-Resolution Terrestrial LiDAR and UAV Surveys." *Geomorphology* 269 (September):8–22. https://doi.org/10.1016/j.geomorph.2016.06.027.

Nourani, Vahid, and Omid Kalantari. 2010. "Integrated Artificial Neural Network for Spatiotemporal Modeling of Rainfall–Runoff–Sediment Processes." *Environmental Engineering Science* 27 (5):411–22. https://doi.org/10.1089/ees.2009.0353.

Nourani, Vahid, Omid Kalantari, and Aida Hosseini Baghanam. 2012. "Two Semidistributed ANN-Based Models for Estimation of Suspended Sediment Load." *Journal of Hydrologic Engineering* 17 (12):1368–1380.

Nourani, Vahid, M. Parhizkar, F. Vousoughi, and B. Amini. 2013. "Capability of Artificial Neural Network for Detecting Hysteresis Phenomenon Involved in Hydrological Processes." *Journal of Hydrologic Engineering* 19 (5):896–906. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000870.

O'Connor, Peter, Daniel Neil, Shih-Chii Liu, Tobi Delbruck, and Michael Pfeiffer. 2013. "Real-Time Classification and Sensor Fusion with a Spiking Deep Belief Network." *Frontiers in Neuroscience* 7 (October). https://doi.org/10.3389/fnins.2013.00178.

Olson, Scott A. 2014. "Estimation of Flood Discharges at Selected Annual Exceedance Probabilities for Unregulated, Rural Streams in Vermont with a Section on Vermont Regional Skew Regression, by Veilleux, A.G." 2014–5078. Scientific Investigations Report. Reston, VA: US Geological Survey.

Olyaie, Ehsan, Hossein Banejad, Kwok-Wing Chau, and Assefa M. Melesse. 2015. "A Comparison of Various Artificial Intelligence Approaches Performance for Estimating Suspended Sediment Load of River Systems: A Case Study in United States." *Environmental Monitoring and Assessment* 187 (4). https://doi.org/10.1007/s10661-015-4381-1.

Onderka, Milan, Andreas Krein, Sebastian Wrede, Núria Martínez-Carreras, and Lucien Hoffmann. 2012. "Dynamics of Storm-Driven Suspended Sediments in a Headwater Catchment Described by Multivariable Modeling." *Journal of Soils and Sediments* 12 (4):620–35. https://doi.org/10.1007/s11368-012-0480-6.

O'Neal, Michael A., and James E. Pizzuto. 2011. "The Rates and Spatial Patterns of Annual Riverbank Erosion Revealed through Terrestrial Laser-Scanner Surveys of the South River, Virginia." *Earth Surface Processes and Landforms* 36 (5):695–701. https://doi.org/10.1002/esp.2098.

Osman, Akode M., and Colin R. Thorne. 1988. "Riverbank Stability Analysis. I: Theory." *Journal of Hydraulic Engineering* 114 (2):134–50. https://doi.org/10.1061/(ASCE)0733-9429(1988)114:2(134).

Oudin, Ludovic, Vazken Andréassian, Charles Perrin, Claude Michel, and Nicolas Le Moine. 2008. "Spatial Proximity, Physical Similarity, Regression and Ungaged Catchments: A Comparison of Regionalization Approaches Based on 913 French Catchments: A Comparison of Regionalization Approaches on 913 Catchments." *Water Resources Research* 44 (3):n/a-n/a. https://doi.org/10.1029/2007WR006240.

Ouédraogo, Mohamar Moussa, Aurore Degré, Charles Debouche, and Jonathan Lisein. 2014. "The Evaluation of Unmanned Aerial System-Based Photogrammetry and Terrestrial Laser Scanning to Generate DEMs of Agricultural Watersheds." *Geomorphology* 214 (June):339–55. https://doi.org/10.1016/j.geomorph.2014.02.016.

Pearce, Andrea R., Donna M. Rizzo, Mary C. Watzin, and Gregory K. Druschel. 2013. "Unraveling Associations between Cyanobacteria Blooms and In-Lake Environmental Conditions in Missisquoi Bay, Lake Champlain, USA, Using a Modified Self-Organizing Map." *Environmental Science & Technology* 47 (24):14267–74. https://doi.org/10.1021/es403490g.

Pellerin, Brian A., Beth A. Stauffer, Dwane A. Young, Daniel J. Sullivan, Suzanne B. Bricker, Mark R. Walbridge, Gerard A. Clyde, and Denice M. Shaw. 2016. "Emerging Tools for Continuous Nutrient Monitoring Networks: Sensors Advancing Science and Water Resources Protection." *JAWRA Journal of the American Water Resources Association* 52 (4):993–1008. https://doi.org/10.1111/1752-1688.12386.

Perks, M. T., G. J. Owen, C. McW. H. Benskin, J. Jonczyk, C. Deasy, S. Burke, S. M. Reaney, and P. M. Haygarth. 2015. "Dominant Mechanisms for the Delivery of Fine Sediment and Phosphorus to Fluvial Networks Draining Grassland Dominated Headwater Catchments." *Science of The Total Environment* 523 (August):178–90. https://doi.org/10.1016/j.scitotenv.2015.03.008.

Perroy, Ryan L., Bodo Bookhagen, Gregory P. Asner, and Oliver A. Chadwick. 2010. "Comparison of Gully Erosion Estimates Using Airborne and Ground-Based LiDAR on Santa Cruz Island, California." *Geomorphology* 118 (3–4):288–300. https://doi.org/10.1016/j.geomorph.2010.01.009.

Piégay, H., S. E. Darby, E. Mosselman, and N. Surian. 2005. "A Review of Techniques Available for Delimiting the Erodible River Corridor: A Sustainable Approach to Managing Bank Erosion." *River Research and Applications* 21 (7):773–89. https://doi.org/10.1002/rra.881.

Pieri, Linda, Matteo Poggio, Marco Vignudelli, and Marco Bittelli. 2014. "Evaluation of the WEPP Model and Digital Elevation Grid Size, for Simulation of Streamflow and Sediment Yield in a Heterogeneous Catchment." *Earth Surface Processes and Landforms* 39 (10):1331–44. https://doi.org/10.1002/esp.3527.

Pietroń, Jan, Jerker Jarsjö, Anna O. Romanchenko, and Sergey R. Chalov. 2015. "Model Analyses of the Contribution of in-Channel Processes to Sediment Concentration Hysteresis Loops." *Journal of Hydrology* 527 (August):576–89. https://doi.org/10.1016/j.jhydrol.2015.05.009.

Pirasteh, Saied, and Jonathan Li. 2016. "Landslides Investigations from Geoinformatics Perspective: Quality, Challenges, and Recommendations." *Geomatics, Natural Hazards and Risk*, October, 1–18. https://doi.org/10.1080/19475705.2016.1238850.

PRISM Climate Group. 2015. "30-Yr Normal Preipitation: Annual, Period: 1981-2010." Oregon State University. http://prism.oregonstate.edu.

Raghuwanshi, N., R. Singh, and L. Reddy. 2006. "Runoff and Sediment Yield Modeling Using Artificial Neural Networks: Upper Siwane River, India." *Journal of Hydrologic Engineering* 11 (1):71–79. https://doi.org/10.1061/(ASCE)1084-0699(2006)11:1(71).

Rajaee, Taher, Seyed Ahmad Mirbagheri, Mohammad Zounemat-Kermani, and Vahid Nourani. 2009. "Daily Suspended Sediment Concentration Simulation Using ANN and Neuro-Fuzzy Models." *Science of The Total Environment* 407 (17):4916–27. https://doi.org/10.1016/j.scitotenv.2009.05.016.

Ramos, Tiago B., Maria C. Gonçalves, Maria A. Branco, David Brito, Sara Rodrigues, José-Miguel Sánchez-Pérez, Sabine Sauvage, et al. 2015. "Sediment and Nutrient Dynamics during Storm Events in the Enxoé Temporary River, Southern Portugal." *CATENA* 127 (April):177–90. https://doi.org/10.1016/j.catena.2015.01.001.

Rasmussen, Patrick P., John R. Gray, G. Douglas Glysson, and Andrew C. Ziegler. 2011. "Guidelines and Procedures for Computing Time-Series Suspended-Sediment Concentrations and Loads from in-Stream Turbidity-Sensor and Streamflow Data." In *Applications of Hydraulics*. Reston, VA: US Geological Survey.

Razavi, Tara, and Paulin Coulibaly. 2013. "Streamflow Prediction in Ungauged Basins: Review of Regionalization Methods." *Journal of Hydrologic Engineering* 18 (8):958–75. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000690.

Reid, Leslie M., and Thomas Dunne. 1996. *Rapid Evaluation of Sediment Budgets.* Catena Verlag.

Reinfelds, Ivars. 1997. "Reconstruction of Changes in Bankfull Width: A Comparison of Surveyed Cross-Sections and Aerial Photography." *Applied Geography* 17 (3):203–13. https://doi.org/10.1016/S0143-6228(97)00005-2.

Resop, J. P., and W. C. Hession. 2010. "Terrestrial Laser Scanning for Monitoring Streambank Retreat: Comparison with Traditional Surveying Techniques." *Journal of Hydraulic Engineering* 136:794–98. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000233.

Rhoades, Erica L., Michael A. O'Neal, and James E. Pizzuto. 2009. "Quantifying Bank Erosion on the South River from 1937 to 2005, and Its Importance in Assessing Hg Contamination." *Applied Geography* 29 (1):125–34. https://doi.org/10.1016/j.apgeog.2008.08.005.

Rinaldi, Massimo, and Stephen E. Darby. 2007. "9 Modelling River-Bank-Erosion Processes and Mass Failure Mechanisms: Progress towards Fully Coupled Simulations." In *Developments in Earth Surface Processes*, edited by Helmut Habersack, Hervé Piégay, and Massimo Rinaldi, 11:213–39. Gravel-Bed Rivers VI: From Process Understanding to River Restoration. Amsterdam, Netherlands: Elsevier. http://www.sciencedirect.com/science/article/pii/S0928202507111263.

Rizzo, Donna M., and David E. Dougherty. 1994. "Characterization of Aquifer Properties Using Artificial Neural Networks: Neural Kriging." *Water Resources Research* 30 (2):483–497. https://doi.org/10.1029/93WR02477.

Roze, A., J. C. Zufferey, A. Beyeler, and A. McClellan. 2014. "eBee RTK Accuracy Assessment." *White Paper Sense Fly*. https://www.sensefly.com/fileadmin/user_upload/sensefly/documents/eBee-RTK-Accuracy-Assessment.pdf.

Rumelhart, D. E., J. L. McClelland, and G. E. Hinton. 1986. "Parallel Distributed Processing: Explorations in the Microstructure of Cognition."

Sarikaya, Ruhi, Geoffrey E. Hinton, and Anoop Deoras. 2014. "Application of Deep Belief Networks for Natural Language Understanding." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22 (4):778–84. https://doi.org/10.1109/TASLP.2014.2303296.

Savitzky, Abraham, and Marcel JE Golay. 1964. "Smoothing and Differentiation of Data by Simplified Least Squares Procedures." *Analytical Chemistry* 36 (8):1627–1639.

Schmelter, M.L., and D. K. Stevens. 2013. "Traditional and Bayesian Statistical Models in Fluvial Sediment Transport." *Journal of Hydraulic Engineering* 139 (3):336–40. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000672.

Seeger, M, M.-P Errea, S Beguería, J Arnáez, C Martí, and J.M García-Ruiz. 2004. "Catchment Soil Moisture and Rainfall Characteristics as Determinant Factors for Discharge/Suspended Sediment Hysteretic Loops in a Small Headwater Catchment in the Spanish Pyrenees." *Journal of Hydrology* 288 (3–4):299–311. https://doi.org/10.1016/j.jhydrol.2003.10.012.

Shahbazi, Mozhdeh, Gunho Sohn, Jérôme Théau, and Patrick Menard. 2015. "Development and Evaluation of a UAV-Photogrammetry System for Precise 3D Environmental Modeling." *Sensors* 15 (11):27493–524. https://doi.org/10.3390/s151127493.

Sherriff, Sophie C., John S. Rowan, Owen Fenton, Philip Jordan, Alice R. Melland, Per-Erik Mellander, and Daire Ó hUallacháin. 2016. "Storm Event Suspended Sediment-Discharge Hysteresis and Controls in Agricultural Watersheds: Implications for Watershed Scale Sediment Management." *Environmental Science & Technology* 50 (4):1769–78. https://doi.org/10.1021/acs.est.5b04573.

Shiri, Jalal, and Özgur Kişi. 2012. "Estimation of Daily Suspended Sediment Load by Using Wavelet Conjunction Models." *Journal of Hydrologic Engineering* 17 (9):986–1000. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000535.

Simon, A., A. Curini, S. E. Darby, and E. J. Langendoen. 2000. "Bank and near-Bank Processes in an Incised Channel." *Geomorphology* 35 (3):193–217. https://doi.org/10.1016/S0169-555X(00)00036-2.

Simon, Andrew, and Massimo Rinaldi. 2006. "Disturbance, Stream Incision, and Channel Evolution: The Roles of Excess Transport Capacity and Boundary Materials in Controlling Channel Response." *Geomorphology*, 37th Binghamton Geomorphology SymposiumThe Human Role in Changing Fluvial Systems, 79 (3–4):361–83. https://doi.org/10.1016/j.geomorph.2006.06.037.

Singh, Ajai, Mohd Imtiyaz, R. K. Isaac, and D. M. Denis. 2013. "Comparison of Artificial Neural Network Models for Sediment Yield Prediction at Single Gauging Station of Watershed in Eastern India." *Journal of Hydrologic Engineering* 18 (1):115–20. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000601.

Singh, Pankaj, and M.C. Deo. 2007. "Suitability of Different Neural Networks in Daily Flow Forecasting." *Applied Soft Computing* 7 (3):968–78. https://doi.org/10.1016/j.asoc.2006.05.003.

Smith, Hugh G., and Deirdre Dragovich. 2009. "Interpreting Sediment Delivery Processes Using Suspended Sediment-Discharge Hysteresis Patterns from Nested Upland Catchments, South-Eastern Australia." *Hydrological Processes* 23 (17):2415–26. https://doi.org/10.1002/hyp.7357.

Smith, Mark William, and Damià Vericat. 2015. "From Experimental Plots to Experimental Landscapes: Topography, Erosion and Deposition in Sub-Humid Badlands from Structure-from-Motion Photogrammetry." *Earth Surface Processes and Landforms* 40 (12):1656–71. https://doi.org/10.1002/esp.3747.

Smolensky, Paul. 1986. "Information Processing in Dynamical Systems: Foundations of Harmony Theory." Boulder, CO: University of Colorado at Boulder Department of Computer Science. http://www.dtic.mil/docs/citations/ADA620727.

Soler, Montserrat, Jérôme Latron, and Francesc Gallart. 2008. "Relationships between Suspended Sediment Concentrations and Discharge in Two Small Research Basins in a Mountainous Mediterranean Area (Vallcebre, Eastern Pyrenees)." *Geomorphology* 98 (1–2):143–52. https://doi.org/10.1016/j.geomorph.2007.02.032.

Solomatine, Dimitri P., and Avi Ostfeld. 2008. "Data-Driven Modelling: Some Past Experiences and New Approaches." *Journal of Hydroinformatics* 10 (1):3. https://doi.org/10.2166/hydro.2008.015.

Starkey, Eleanor, Geoff Parkin, Stephen Birkinshaw, Andy Large, Paul Quinn, and Ceri Gibson. 2017. "Demonstrating the Value of Community-Based ('citizen Science') Observations for Catchment Modelling and Characterisation." *Journal of Hydrology* 548 (May):801–17. https://doi.org/10.1016/j.jhydrol.2017.03.019.

Stryker, J., B. Wemple, and A. Bomblies. 2017. "Modeling Sediment Mobilization Using a Distributed Hydrological Model Coupled with a Bank Stability Model." *Water Resources Research* 53 (3):2051–73. https://doi.org/10.1002/2016WR019143.

Tamminga, A., C. Hugenholtz, B. Eaton, and M. Lapointe. 2015. "Hyperspatial Remote Sensing of Channel Reach Morphology and Hydraulic Fish Habitat Using an Unmanned Aerial Vehicle (UAV): A First Assessment in the Context of River Research and Management." *River Research and Applications* 31 (3):379–91. https://doi.org/10.1002/rra.2743.

Tamminga, Aaron D., Brett C. Eaton, and Chris H. Hugenholtz. 2015. "UAS-Based Remote Sensing of Fluvial Change Following an Extreme Flood Event." *Earth Surface Processes and Landforms* 40 (11):1464–76. https://doi.org/10.1002/esp.3728.

Tananaev, N.I., and M.V. Debolskiy. 2014. "Turbidity Observations in Sediment Flux Studies: Examples from Russian Rivers in Cold Environments." *Geomorphology* 218 (August):63–71. https://doi.org/10.1016/j.geomorph.2013.09.031.

Tang, Weigang, and Sean K. Carey. 2017. "HydRun: A MATLAB Toolbox for Rainfall–runoff Analysis." *Hydrological Processes*. https://doi.org/10.1002/hyp.11185.

Testolin, Alberto, Ivilin Stoianov, Michele De Filippo De Grazia, and Marco Zorzi. 2013. "Deep Unsupervised Learning on a Desktop PC: A Primer for Cognitive Scientists." *Frontiers in Psychology* 4. https://doi.org/10.3389/fpsyg.2013.00251.

Thakur, Praveen K., Chalantika Laha, and S. P. Aggarwal. 2012. "River Bank Erosion Hazard Study of River Ganga, Upstream of Farakka Barrage Using Remote Sensing and GIS." *Natural Hazards* 61 (3):967–87. https://doi.org/10.1007/s11069-011-9944-z.

Thoma, David P., Satish C. Gupta, Marvin E. Bauer, and C.E. Kirchoff. 2005. "Airborne Laser Scanning for Riverbank Erosion Assessment." *Remote Sensing of Environment* 95 (4):493–501. https://doi.org/10.1016/j.rse.2005.01.012.

Tieleman, Tijmen. 2008. "Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient." In *Proceedings of the 25th International Conference on Machine Learning*, 1064–1071. ACM. http://dl.acm.org/citation.cfm?id=1390290.

Tseng, Chih-Ming, Ching-Weei Lin, Colin P. Stark, Jin-Kin Liu, Li-Yuan Fei, and Yu-Chung Hsieh. 2013. "Application of a Multi-Temporal, LiDAR-Derived, Digital Terrain Model in a Landslide-Volume Estimation: Multi-Temporal LiDAR DTM in Landslide Volume Estimation." *Earth Surface Processes and Landforms*, July, 1587–1601. https://doi.org/10.1002/esp.3454.

Uhrich, Mark A., and Heather M. Bragg. 2003. "Monitoring Instream Turbidity to Estimate Continuous Suspended-Sediment Loads and Yields and Clay-Water Volumes in the Upper North Santiam River Basin, Oregon, 1998–2000." Water-Resources Investigations Report 03–4098. Portland, Oregon: US Geological Survey.

Underwood, Kristen L. 2004. "Phase 2 Stream Geomorphic Assessment New Haven River Watershed Addison County, Vermont." Middlebury, Vermont: Addison County Regional Planning Commission.

U.S. Environmental Protection Agency. 1999. *Method 160.2: Total Suspended Solids (TSS) (Gravimetric, Dried at 103–105 ° C)*. Washington, D.C.

US EPA. 2013. "National Summary of Impaired Waters and TMDL Information." Watershed Assessment, Tracking & Environmental Results. 2013. http://iaspub.epa.gov/waters10/attains_nation_cy.control?p_report_type=T#causes_303d.

Vallet, J, F Panissod, Christoph Strecha, and Matthieu Tracol. 2012. "Photogrammetric Performance of an Ultra Light Weight Swinglet UAV." *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXVIII-1/C22:253–58.

Vaughan, M. C. H., W. B. Bowden, J. B. Shanley, A. Vermilyea, R. Sleeper, A. J. Gold, S. M. Pradhanang, et al. 2017. "High-Frequency Dissolved Organic Carbon and Nitrate Measurements Reveal Differences in Storm Hysteresis and Loading in Relation to Land Cover and Seasonality: High-Resolution Doc and Nitrate Dynamics." *Water Resources Research*, July. https://doi.org/10.1002/2017WR020491.

Walling, D. E., and A. L. Collins. 2008. "The Catchment Sediment Budget as a Management Tool." *Environmental Science & Policy* 11 (2):136–43. https://doi.org/10.1016/j.envsci.2007.10.004.

Walling, D. E., A. L. Collins, and R. W. Stroud. 2008. "Tracing Suspended Sediment and Particulate Phosphorus Sources in Catchments." *Journal of Hydrology*, Characterization and apportionment of nutrient and sediment sources in catchments, 350 (3–4):274–89. https://doi.org/10.1016/j.jhydrol.2007.10.047.

Warrick, Jonathan A., John M. Melack, and Blair M. Goodridge. 2015. "Sediment Yields from Small, Steep Coastal Watersheds of California." *Journal of Hydrology: Regional Studies* 4 (September):516–34. https://doi.org/10.1016/j.ejrh.2015.08.004.

Wass, P. D., S. D. Marks, J. W. Finch, G. J. L. Leeks, and J. K. Ingram. 1997. "Monitoring and Preliminary Interpretation of in-River Turbidity and Remote Sensed Imagery for Suspended Sediment Transport Studies in the Humber Catchment." *Science of The Total Environment*, U.K. Fluxes to the North Sea, Land Ocean Interaction Study (LOIS) Rivers Basins Research, the First Two Years, 194–195 (February):263–83. https://doi.org/10.1016/S0048-9697(96)05370-3.

Watts, Adam C., Vincent G. Ambrosia, and Everett A. Hinkley. 2012. "Unmanned Aircraft Systems in Remote Sensing and Scientific Research: Classification and Considerations of Use." *Remote Sensing* 4 (6):1671–92. https://doi.org/10.3390/rs4061671.

Weber, M. D., and G. B. Pasternack. 2017. "Valley-Scale Morphology Drives Differences in Fluvial Sediment Budgets and Incision Rates during Contrasting Flow Regimes." *Geomorphology* 288 (July):39–51. https://doi.org/10.1016/j.geomorph.2017.03.018.

Wemple, Beverley C., Gordon E. Clark, Donald S. Ross, and Donna M. Rizzo. 2017. "Identifying the Spatial Pattern and Importance of Hydro-Geomorphic Drainage Impairments on Unpaved Roads in the Northeastern USA." *Earth Surface Processes and Landforms*, January, n/a-n/a. https://doi.org/10.1002/esp.4113.

Westoby, M. J., J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds. 2012. "'Structure-from-Motion' Photogrammetry: A Low-Cost, Effective Tool for Geoscience Applications." *Geomorphology* 179 (December):300–314. https://doi.org/10.1016/j.geomorph.2012.08.021.

Wheaton, Joseph M., James Brasington, Stephen E. Darby, and David A. Sear. 2010. "Accounting for Uncertainty in DEMs from Repeat Topographic Surveys: Improved Sediment Budgets." *Earth Surface Processes and Landforms* 35 (2):136–56. https://doi.org/10.1002/esp.1886.

Whitehead, Ken, and Chris H. Hugenholtz. 2014. "Remote Sensing of the Environment with Small Unmanned Aircraft Systems (UASs), Part 1: A Review of Progress and Challenges." *Journal of Unmanned Vehicle Systems* 2 (3):69–85. https://doi.org/10.1139/juvs-2014-0006.

Whitehead, Ken, Chris H. Hugenholtz, Stephen Myshak, Owen Brown, Adam LeClair, Aaron Tamminga, Thomas E. Barchyn, Brian Moorman, and Brett Eaton. 2014. "Remote Sensing of the Environment

with Small Unmanned Aircraft Systems (UASs), Part 2: Scientific and Commercial Applications." *Journal of Unmanned Vehicle Systems* 2 (3):86–102. https://doi.org/10.1139/juvs-2014-0007.

Williams, Garnett P. 1989. "Sediment Concentration versus Water Discharge during Single Hydrologic Events in Rivers." *Journal of Hydrology* 111 (1):89–106.

Williams, Richard, James Brasington, Damia Vericat, Murray Hicks, Fred Labrosse, and Mark Neal. 2011. "Chapter Twenty - Monitoring Braided River Change Using Terrestrial Laser Scanning and Optical Bathymetric Mapping." In *Geomorphological Mapping Methods and Applications*, Volume 15:507–32. Oxford, UK: Elsevier. http://www.sciencedirect.com/science/article/pii/B9780444534460000203.

Woodget, A. S., P. E. Carbonneau, F. Visser, and I. P. Maddock. 2015. "Quantifying Submerged Fluvial Topography Using Hyperspatial Resolution UAS Imagery and Structure from Motion Photogrammetry." *Earth Surface Processes and Landforms* 40 (1):47–64. https://doi.org/10.1002/esp.3613.

Yang, Chih Ted, Reza Marsooli, and Mohammad Taghi Aalami. 2009. "Evaluation of Total Load Sediment Transport Formulas Using ANN." *International Journal of Sediment Research* 24 (3):274–86. https://doi.org/10.1016/S1001-6279(10)60003-0.

Yang, J., Y. Liu, E. Xing, and A. Hauptmann. 2007. "Harmonium Models for Semantic Video Representation and Classification." In *Proceedings of the 2007 SIAM International Conference on Data Mining*, 378–89. Proceedings. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611972771.34.

Yeshaneh, Eleni, Alexander Eder, and Günter Blöschl. 2014. "Temporal Variation of Suspended Sediment Transport in the Koga Catchment, North Western Ethiopia and Environmental Implications." *Hydrological Processes* 28 (24):5972–84. https://doi.org/10.1002/hyp.10090.

Zhu, Yun-Mei, X. X. Lu, and Yue Zhou. 2007. "Suspended Sediment Flux Modeling with Artificial Neural Network: An Example of the Longchuanjiang River in the Upper Yangtze Catchment, China." *Geomorphology* 84 (1–2):111–25. https://doi.org/10.1016/j.geomorph.2006.07.010.

Zounemat-Kermani, Mohammad, Özgür Kişi, Jan Adamowski, and Abdollah Ramezani-Charmahineh. 2016. "Evaluation of Data Driven Models for River Suspended Sediment Concentration Modeling." *Journal of Hydrology* 535 (April):457–72. https://doi.org/10.1016/j.jhydrol.2016.02.012.

Zuecco, G., D. Penna, M. Borga, and H. J. van Meerveld. 2016. "A Versatile Index to Characterize Hysteresis between Hydrological Variables at the Runoff Event Timescale." *Hydrological Processes* 30 (9):1449–66. https://doi.org/10.1002/hyp.10681.

APPENDIX A: FIELD MONITORING SITES AND DATA SUMMARY

Photos of Turbidity Monitoring Stations

Mad River

Mad River turbidity sensor installation looking downstream. USGS gauge on opposite bank

Mad River turbidity sensor installation during storm events looking upstream

Mad River

Mad River turbidity sensor installation looking downstream. USGS gauge on opposite bank

Mad River turbidity sensor installation during storm event looking upstream

Shepard Brook

Shepard Brook turbidity sensor and stage sensor installation

Shepard Brook turbidity sensor installation during storm event looking upstream

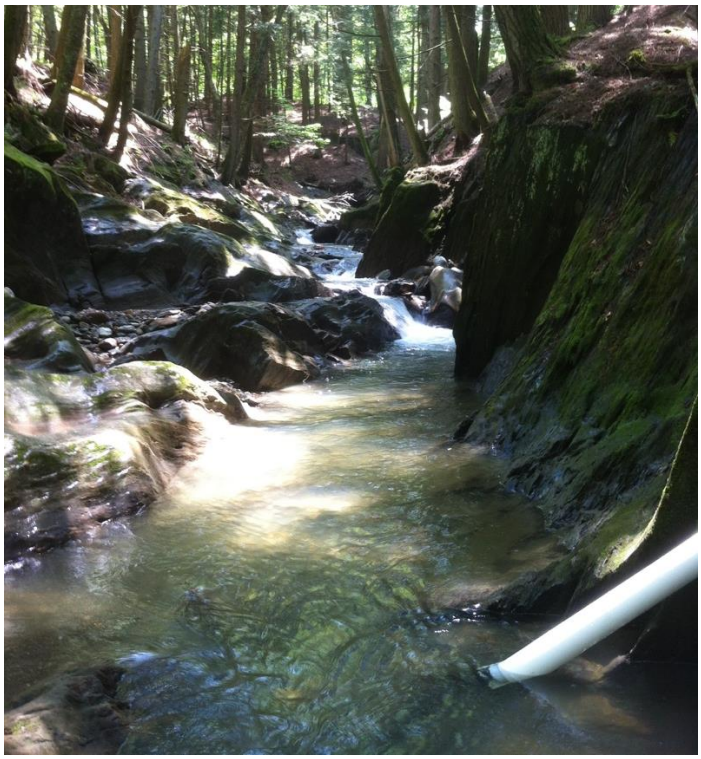
Mill Brook

Mill Brook turbidity sensor and stage sensor installation looking upstream

Mill Brook monitoring site location during storm event looking upstream

High Bridge Brook

High Bridge Brook turbidity sensor and stage sensor installation looking upstream



Detail of High Bridge Brook sensor installation in channel, looking upstream along channel

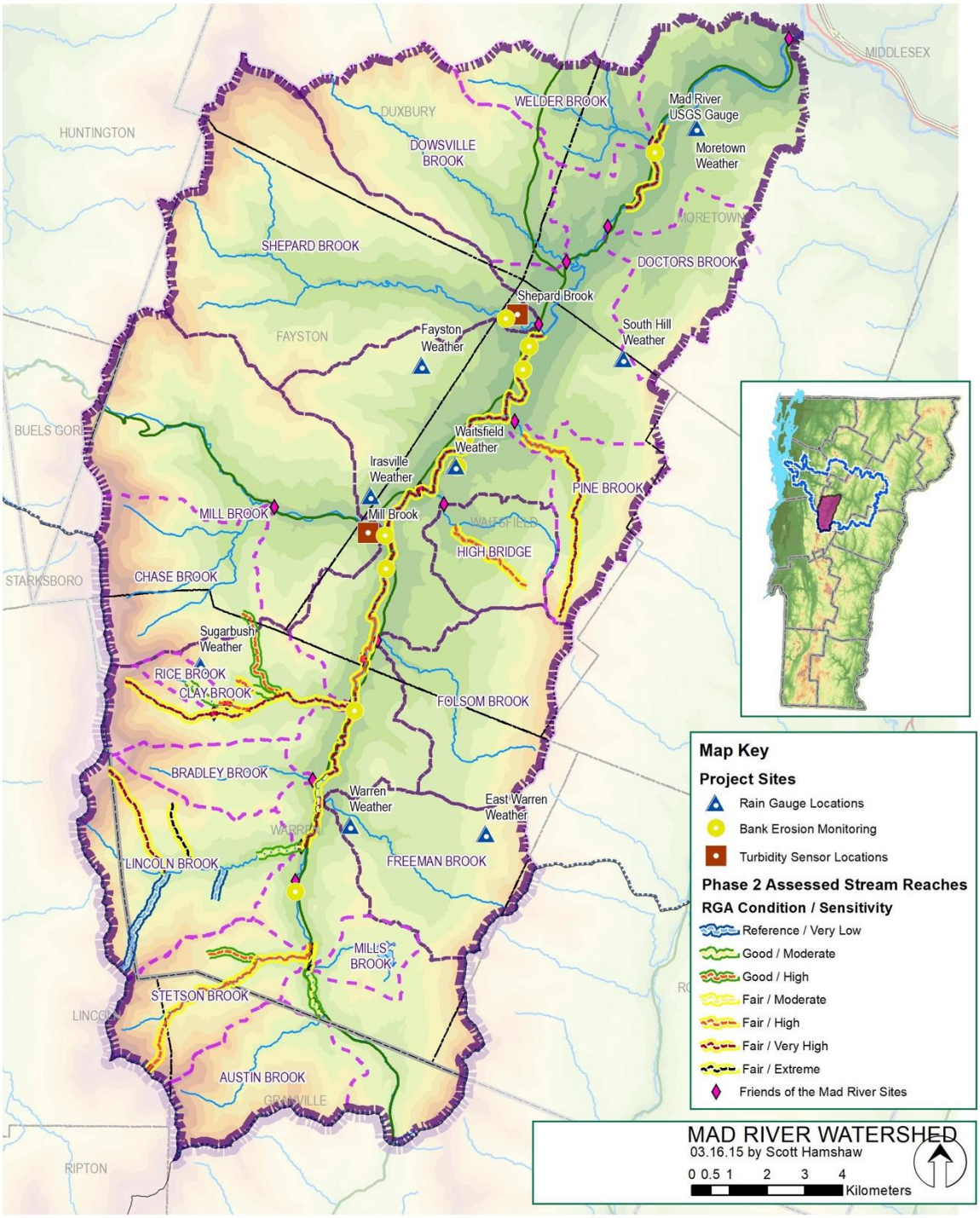
Folsom Brook

Folsom Brook turbidity sensor and stage sensor installation looking upstream during storm event

Freeman Brook

Freeman Brook turbidity sensor and stage sensor installation looking downstream

Study Area Map

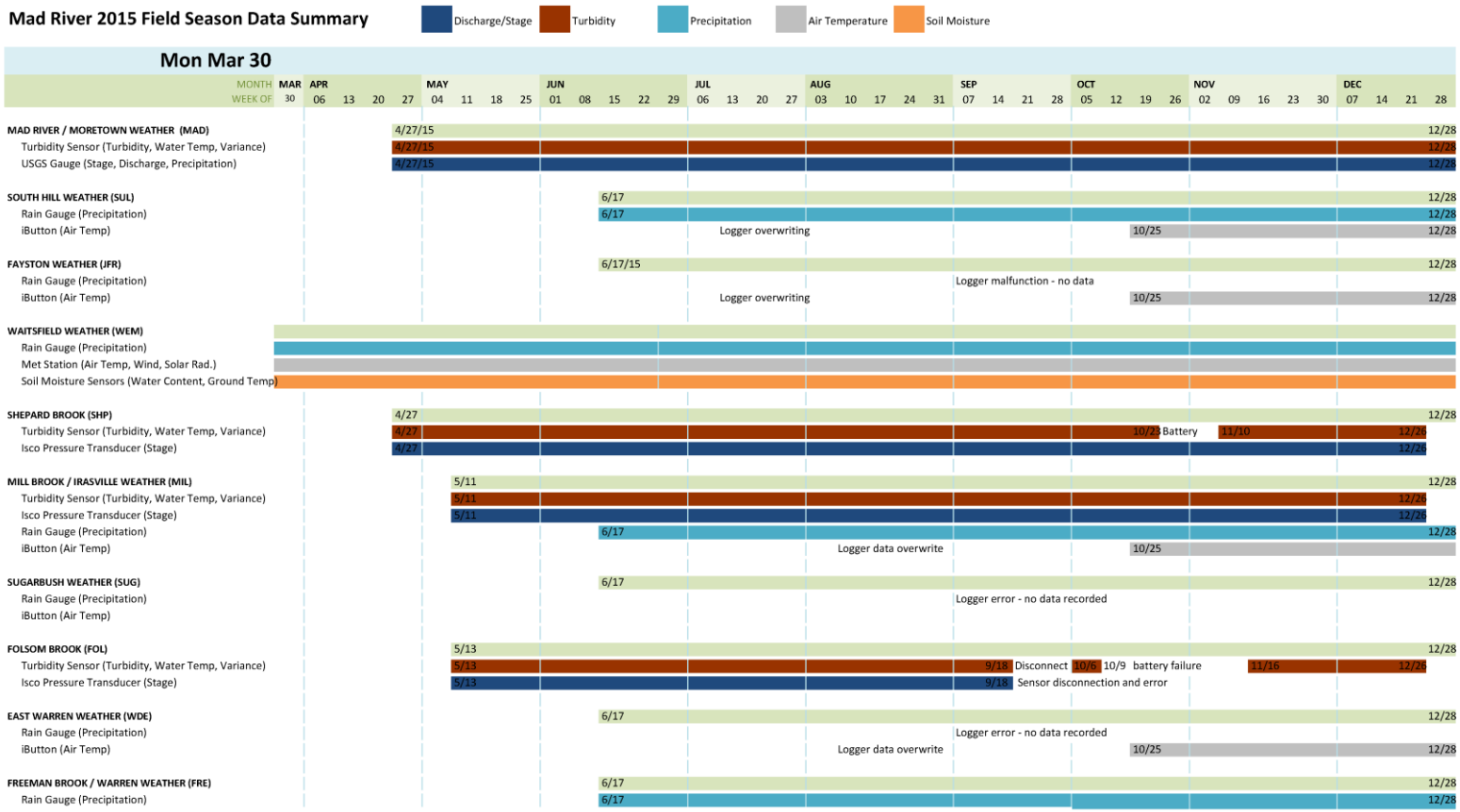


Summary of Monitoring Data

Mad River Watershed Data Summary — **2015 Field Season**

Last Updated: 3/2/16

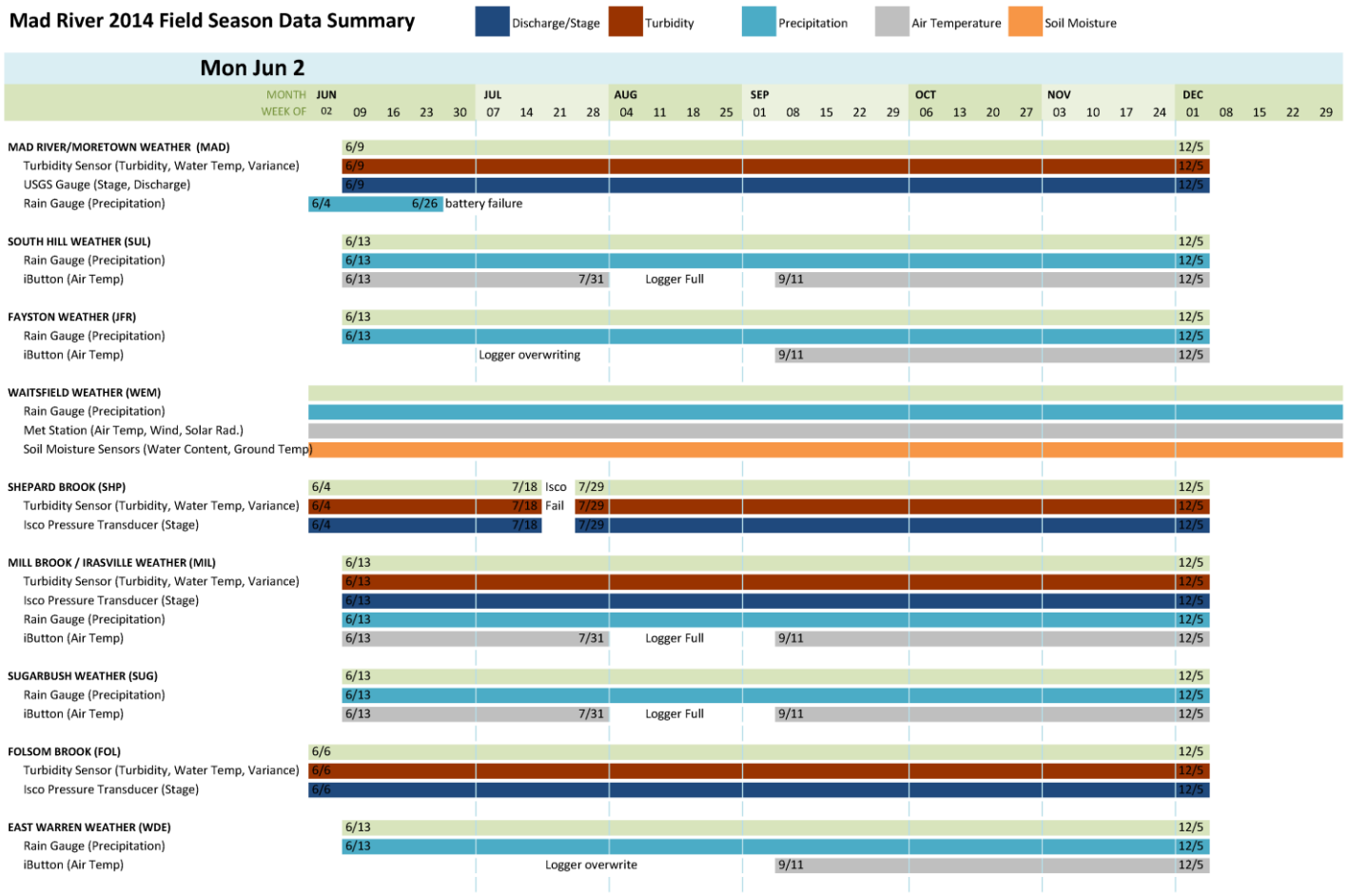
Site Name	Site Location	Site Code				Short Code
Mad River / Moretown Weather	**Sainsbury Property**	**WR_MadRvr_0428800 / WR_MoretownMet_544**				**MAD**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	4/27/15	12/28/15	
Water Temperature	15-min	°C	DTS-12	4/27/15	12/28/15	
Stage	15-min	ft	USGS Gauge	N/A	N/A	
Discharge	15-min	cfs	USGS Gauge	N/A	N/A	
Precipitation	0.01 in	in	USGS Gauge	N/A		
Fayston Weather	**Jefferys Property**	**WR_FaystonMet_1470**				**JFR**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Precipitation	0.01 in	in	Onset Rain Gauge	6/17/15	12/28/15	Bad logger, no data recorded
Air Temperature	30-min	°C	iButton thermochron	6/17/15	12/28/15	No data until 10/25/15
South Hill Weather	**Sullivan Property**	**WR_SHillMet_1410**				**SUL**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Precipitation	0.01 in	in	Onset Rain Gauge	6/17/15	12/28/15	
Air Temperature	30-min	°C	iButton thermochron	6/17/15	12/28/15	No data until 10/25/15
Shepard Brook	**Reynolds Property**	**WR_ShepBk_640**				**SHP**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	4/27/15		Battery fail from 10/23-11/10
Water Temperature	15-min	°C	DTS-12	4/27/15		Battery fail from 10/23-11/11
Stage	15-min	m	Isco Module	4/27/15		
Mill Brook / Irasville Weather	**Mad River Massage**	**WR_MillBk_715 / WR_IrasvilleMet_715**				**MIL**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	5/11/15	12/28/15	
Water Temperature	15-min	°C	DTS-12	5/11/15	12/28/15	
Stage	15-min	m	Isco Module	5/11/15	12/28/15	
Air Temperature	30-min	°C	iButton thermochron	6/17/15	12/28/15	No data until 10/25/15
Precipitation	0.01 in	in	Onset Rain Gauge	6/17/15	12/28/15	
Sugarbush Weather	**Sugarbush**	**WR_SgrbshMet_2170**				**SUG**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Air Temperature	30-min	°C	iButton thermochron	N/A	N/A	
Precipitation	0.01 in	in	Onset Rain Gauge	6/17/15	12/28/15	Bad logger, no data recorded
Folsom Brook	**River House Condos**	**WR_FolsomBk_750**				**FOL**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	5/13/15	12/26/15	Gap from 9/18-10/6 and 10/9-11/16
Water Temperature	15-min	°C	DTS-12	5/13/15	12/26/15	Gap from 9/18-10/6 and 10/9-11/17
Stage	15-min	m	Isco Module	5/13/15	12/26/15	No stage after 9/18/15 due to sensor error
Freeman Brook / Warren Weather	**Perot Property**	**WR_FreeBk_895 / WR_WarrenMet_895**				**FRE**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Air Temperature	30-min	°C	iButton thermochron	6/17/15	12/28/15	No data until 10/25/15
Precipitation	0.01 in	in	Onset Rain Gauge	6/17/15	12/28/15	
East Warren Weather	**Wade Property**	**WR_EWarrenMet_1880**				**WDE**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Air Temperature	30-min	°C	iButton thermochron	6/17/15	12/28/15	No data until 10/25/15
Precipitation	0.01 in	in	Onset Rain Gauge	6/17/15	12/28/15	Bad logger, no data recorded


Mad River 2015 Field Season Data Summary
Discharge/Stage
Turbidity
Precipitation
Air Temperature
Soil Moisture
Mon Mar 30
MONTH
WEEK OF
MAR 30
APR 06 13 20 27
MAY 04 11 18 25
JUN 01 08 15 22 29
JUL 06 13 20 27
AUG 03 10 17 24 31
SEP 07 14 21 28
OCT 05 12 19 26
NOV 02 09 16 23 30
DEC 07 14 21 28
MAD RIVER / MORETOWN WEATHER (MAD)
Turbidity Sensor (Turbidity, Water Temp, Variance)
USGS Gauge (Stage, Discharge, Precipitation)
4/27/15
12/28
SOUTH HILL WEATHER (SUL)
Rain Gauge (Precipitation)
iButton (Air Temp)
6/17
Logger overwriting
10/25
FAYSTON WEATHER (JFR)
Rain Gauge (Precipitation)
iButton (Air Temp)
6/17/15
Logger malfunction - no data
WAITSFIELD WEATHER (WEM)
Rain Gauge (Precipitation)
Met Station (Air Temp, Wind, Solar Rad.)
Soil Moisture Sensors (Water Content, Ground Temp)
SHEPARD BROOK (SHP)
Turbidity Sensor (Turbidity, Water Temp, Variance)
Isco Pressure Transducer (Stage)
4/27
10/23 Battery
11/10
12/26
MILL BROOK / IRASVILLE WEATHER (MIL)
Turbidity Sensor (Turbidity, Water Temp, Variance)
Isco Pressure Transducer (Stage)
Rain Gauge (Precipitation)
iButton (Air Temp)
5/11
6/17
Logger data overwrite
10/25
SUGARBUSH WEATHER (SUG)
Rain Gauge (Precipitation)
iButton (Air Temp)
6/17
Logger error - no data recorded
FOLSOM BROOK (FOL)
Turbidity Sensor (Turbidity, Water Temp, Variance)
Isco Pressure Transducer (Stage)
5/13
9/18 Disconnect
10/6 10/9 battery failure
11/16
12/26
9/18 Sensor disconnection and error
EAST WARREN WEATHER (WDE)
Rain Gauge (Precipitation)
iButton (Air Temp)
6/17
Logger error - no data recorded
Logger data overwrite
10/25
FREEMAN BROOK / WARREN WEATHER (FRE)
Rain Gauge (Precipitation)
6/17
12/28

Mad River Watershed Data Summary **2014 Field Season**

Last Updated: 3/2/16

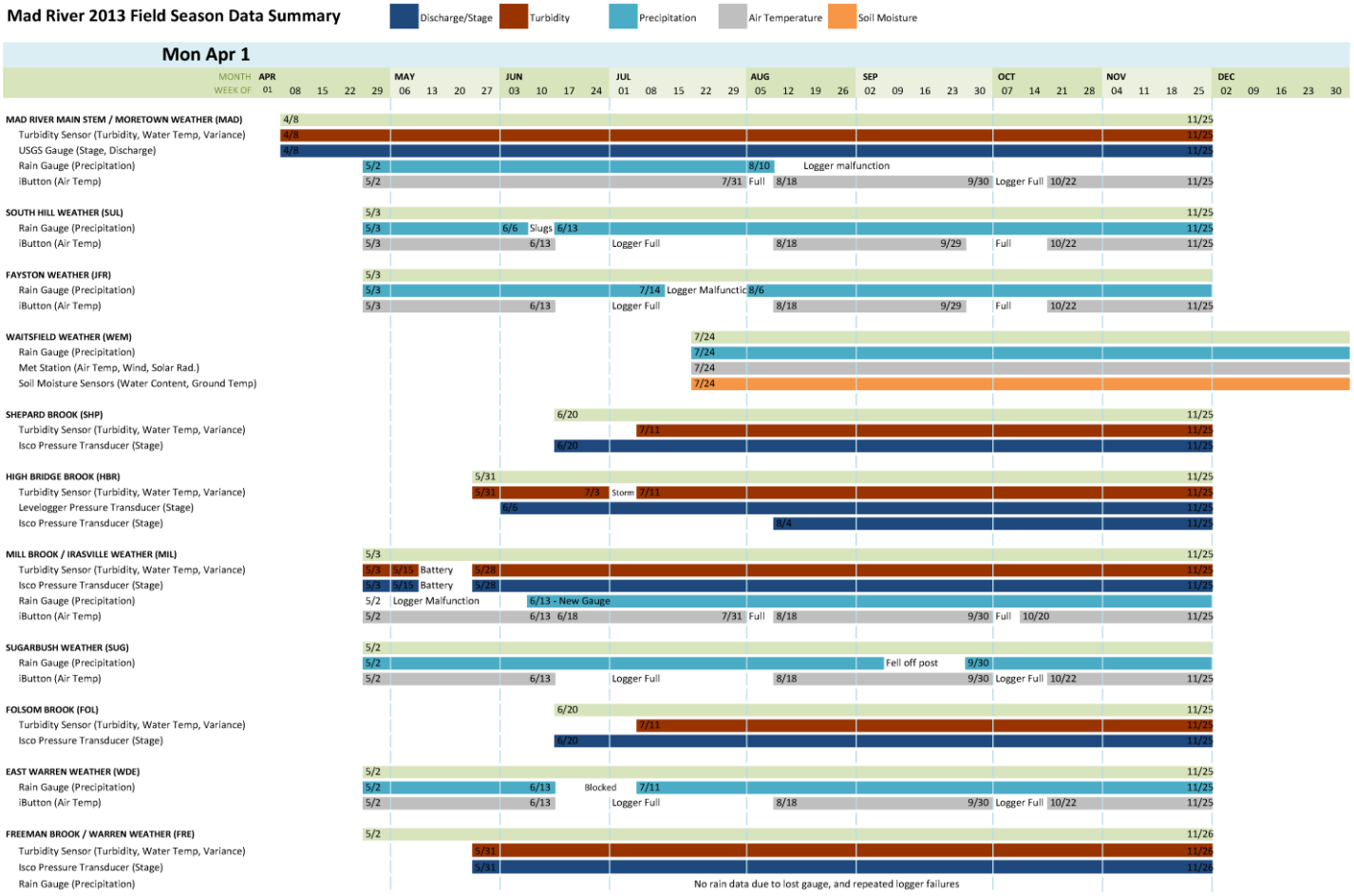
Site Name	Site Location	Site Code				Short Code
Mad River / Moretown Weather	**Sainsbury Property**	**WR_MadRvr_0428800 / WR_MoretownMet_544**				**MAD**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	6/9/14	12/5/14	
Water Temperature	15-min	°C	DTS-12	6/9/14	12/5/14	
Stage	15-min	ft	USGS Gauge	N/A	N/A	
Discharge	15-min	cfs	USGS Gauge	N/A	N/A	
Precipitation	0.01 in	in	Onset Rain Gauge	6/4/14	12/5/14	Battery failure after 6/26/14
Fayston Weather	**Jefferys Property**	**WR_FaystonMet_1470**				**JFR**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Precipitation	0.01 in	in	Onset Rain Gauge	6/13/14	12/5/14	
Air Temperature	30-min	°C	iButton thermochron	6/13/14	12/5/14	No data until 9/11/14
South Hill Weather	**Sullivan Property**	**WR_SHillMet_1410**				**SUL**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Precipitation	0.01 in	in	Onset Rain Gauge	6/13/14	12/5/14	
Air Temperature	30-min	°C	iButton thermochron	6/13/14	12/5/14	Logger full from 7/31 - 9/11
Shepard Brook	**Reynolds Property**	**WR_ShepBk_640**				**SHP**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	6/4/14	125/14	Isco failure 7/18 - 7/29
Water Temperature	15-min	°C	DTS-12	6/4/14	12/5/14	Isco failure 7/18 - 7/29
Stage	15-min	m	Isco Module	6/14/14	12/5/14	Isco failure 7/18 - 7/29
Mill Brook / Irasville Weather	**Mad River Massage**	**WR_MillBk_715 / WR_IrasvilleMet_715**				**MIL**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	6/13/14	12/5/14	
Water Temperature	15-min	°C	DTS-12	6/13/14	12/5/14	
Stage	15-min	m	Isco Module	6/13/14	12/5/14	
Air Temperature	30-min	°C	iButton thermochron	6/13/14	12/5/14	Logger full from 7/31 - 9/11
Precipitation	0.01 in	in	Onset Rain Gauge	6/13/14	12/5/14	
Sugarbush Weather	**Sugarbush**	**WR_SgrbshMet_2170**				**SUG**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Air Temperature	30-min	°C	iButton thermochron	6/13/14	12/5/14	
Precipitation	0.01 in	in	Onset Rain Gauge	6/13/14	12/5/14	
Folsom Brook	**River House Condos**	**WR_FolsomBk_750**				**FOL**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	6/6/14		
Water Temperature	15-min	°C	DTS-12	6/6/14		
Stage	15-min	m	Isco Module	6/6/14		
East Warren Weather	**Wade Property**	**WR_EWarrenMet_1880**				**WDE**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Air Temperature	30-min	°C	iButton thermochron	6/13/14	12/5/14	No data until 9/11/14
Precipitation	0.01 in	in	Onset Rain Gauge	6/13/14	12/5/14	


Mad River 2014 Field Season Data Summary
Discharge/Stage
Turbidity
Precipitation
Air Temperature
Soil Moisture
Mon Jun 2
MONTH
WEEK OF
JUN 02 09 16 23 30
JUL 07 14 21 28
AUG 04 11 18 25
SEP 01 08 15 22 29
OCT 06 13 20 27
NOV 03 10 17 24
DEC 01 08 15 22 29
MAD RIVER/MORETOWN WEATHER (MAD)
6/9
12/5
Turbidity Sensor (Turbidity, Water Temp, Variance)
6/9
12/5
USGS Gauge (Stage, Discharge)
6/9
12/5
Rain Gauge (Precipitation)
6/4
6/26 battery failure
SOUTH HILL WEATHER (SUL)
6/13
12/5
Rain Gauge (Precipitation)
6/13
12/5
iButton (Air Temp)
6/13
7/31
Logger Full
9/11
12/5
FAYSTON WEATHER (JFR)
6/13
12/5
Rain Gauge (Precipitation)
6/13
12/5
iButton (Air Temp)
Logger overwriting
9/11
12/5
WAITSFIELD WEATHER (WEM)
Rain Gauge (Precipitation)
Met Station (Air Temp, Wind, Solar Rad.)
Soil Moisture Sensors (Water Content, Ground Temp)
SHEPARD BROOK (SHP)
6/4
7/18 Isco
7/29
12/5
Turbidity Sensor (Turbidity, Water Temp, Variance)
6/4
7/18 Fail
7/29
12/5
Isco Pressure Transducer (Stage)
6/4
7/18
7/29
12/5
MILL BROOK / IRASVILLE WEATHER (MIL)
6/13
12/5
Turbidity Sensor (Turbidity, Water Temp, Variance)
6/13
12/5
Isco Pressure Transducer (Stage)
6/13
12/5
Rain Gauge (Precipitation)
6/13
12/5
iButton (Air Temp)
6/13
7/31
Logger Full
9/11
12/5
SUGARBUSH WEATHER (SUG)
6/13
12/5
Rain Gauge (Precipitation)
6/13
12/5
iButton (Air Temp)
6/13
7/31
Logger Full
9/11
12/5
FOLSOM BROOK (FOL)
6/6
12/5
Turbidity Sensor (Turbidity, Water Temp, Variance)
6/6
12/5
Isco Pressure Transducer (Stage)
6/6
12/5
EAST WARREN WEATHER (WDE)
6/13
12/5
Rain Gauge (Precipitation)
6/13
12/5
iButton (Air Temp)
Logger overwrite
9/11
12/5

Mad River Watershed Data Summary — 2013 Field Season

Last Updated: 3/2/16

Site Name	Site Location	Site Code				Short Code
Mad River / Moretown Weather	Sainsbury Property	WR_MadRvr_0428800 / WR_MoretownMet_544				MAD
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	4/8/13	11/25/13	
Water Temperature	15-min	°C	DTS-12	4/8/13	11/25/13	
Stage	15-min	ft	USGS Gauge	N/A	N/A	
Discharge	15-min	cfs	USGS Gauge	N/A	N/A	
Precipitation	0.01 in	in	Onset Rain Gauge	5/2/13	11/25/13	Logger malfunction after 8/10/13
Air Temperature	30-min	°C	iButton thermochron	5/2/13	11/25/13	
Fayston Weather	Jefferys Property	WR_FaystonMet_1470				JFR
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Precipitation	0.01 in	in	Onset Rain Gauge	5/3/13	11/25/13	Logger malfunction from 7/14-8/6
Air Temperature	30-min	°C	iButton thermochron	5/3/13	11/25/13	Logger full from 6/13-8/18, 9/29-10/22
South Hill Weather	Sullivan Property	WR_SHillMet_1410				SUL
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Precipitation	0.01 in	in	Onset Rain Gauge	5/3/13	11/25/13	Slugs clogged gauge from 6/6 - 6/13
Air Temperature	30-min	°C	iButton thermochron	5/3/13	11/25/13	Logger full from 6/13-8/18, 9/29-10/22
Shepard Brook	Reynolds Property	WR_ShepBk_640				SHP
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	7/11/13	11/25/13	
Water Temperature	15-min	°C	DTS-12	7/11/13	11/25/13	
Stage	15-min	m	Isco Module	6/20/13	11/25/13	
High Bridge Brook	Joslin Property	WR_HighBrdBk_730				HBR
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	5/31/13	11/25/13	Offline from 7/4 - 7/11
Water Temperature	15-min	°C	DTS-12	5/31/13	11/25/13	Offline from 7/4 - 7/11
Stage	15-min	ft	Levelogger	6/6/13	11/25/13	
Barometric Pressure	15-min	ft	Barologger	6/6/13	11/25/13	
Stage	15-min	m	Isco Module	8/4/13	11/25/13	
Mill Brook / Irasville Weather	Mad River Massage	WR_MillBk_715 / WR_IrasvilleMet_715				MIL
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	5/3/13	11/25/13	Battery failure from 5/15-5/16 and 5/19 - 5/28
Water Temperature	15-min	°C	DTS-12	5/3/13	11/25/13	Battery failure from 5/15-5/16 and 5/19 - 5/28
Stage	15-min	m	Isco Module	5/3/13	11/25/13	Battery failure from 5/15-5/16 and 5/19 - 5/28
Air Temperature	30-min	°C	iButton thermochron	5/2/13	11/25/13	Logger full from 6/13-6/18, 7/31-8/18,9/30-10/
Precipitation	0.01 in	in	Onset Rain Gauge	5/2/13	11/25/13	Logger bad until 6/13
Sugarbush Weather	Sugarbush	WR_SgrbshMet_2170				SUG
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Air Temperature	30-min	°C	iButton thermochron	5/2/13	11/25/13	
Precipitation	0.01 in	in	Onset Rain Gauge	5/2/13	11/25/13	Logger full from 6/13-8/18, 9/29-10/22
Folsom Brook	River House Condos	WR_FolsomBk_750				FOL
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	7/11/13	11/25/13	
Water Temperature	15-min	°C	DTS-12	7/11/13	11/25/13	
Stage	15-min	m	Isco Module	6/20/13	11/25/13	
Freeman Brook / Warren Weathe	Perot Property	WR_FreeBk_895 / WR_WarrenMet_895				FRE
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	5/31/13	11/26/13	
Water Temperature	15-min	°C	DTS-12	5/31/13	11/26/13	
Stage	15-min	m	Isco Module	5/31/13	11/26/13	
Air Temperature	30-min	°C	iButton thermochron	5/3/13	11/26/13	Logger lost on 6/13, logger full 7/23- 8/18, after
Precipitation	0.01 in	in	Onset Rain Gauge	5/3/13	11/26/13	No rain data - repeated failures
East Warren Weather	Wade Property	WR_EWarrenMet_1880				WDE
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Air Temperature	30-min	°C	iButton thermochron	5/2/13	11/25/13	
Precipitation	0.01 in	in	Onset Rain Gauge	5/2/13	11/25/13	

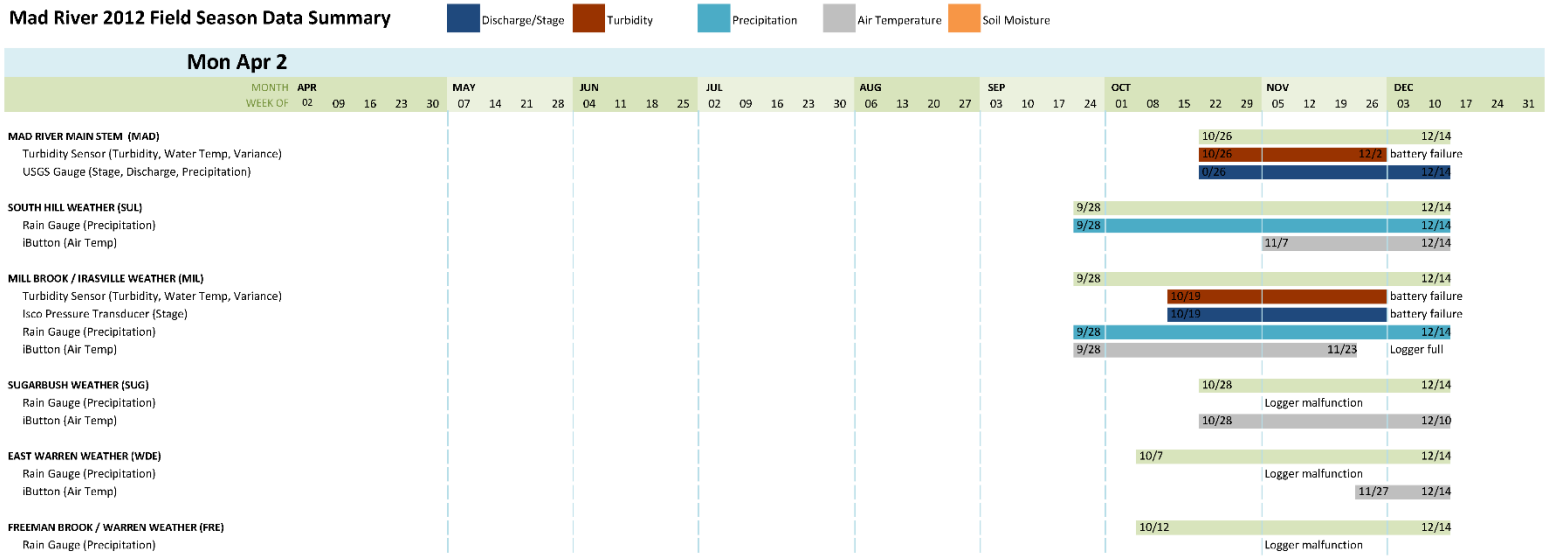

Mad River 2013 Field Season Data Summary
Discharge/Stage
Turbidity
Precipitation
Air Temperature
Soil Moisture
Mon Apr 1
MONTH
WEEK OF
APR 01 08 15 22 29
MAY 06 13 20 27
JUN 03 10 17 24
JUL 01 08 15 22 29
AUG 05 12 19 26
SEP 02 09 16 23 30
OCT 07 14 21 28
NOV 04 11 18 25
DEC 02 09 16 23 30
MAD RIVER MAIN STEM / MORETOWN WEATHER (MAD)
Turbidity Sensor (Turbidity, Water Temp, Variance)
USGS Gauge (Stage, Discharge)
Rain Gauge (Precipitation)
iButton (Air Temp)
SOUTH HILL WEATHER (SUL)
Rain Gauge (Precipitation)
iButton (Air Temp)
FAYSTON WEATHER (JFR)
Rain Gauge (Precipitation)
iButton (Air Temp)
WAITSFIELD WEATHER (WEM)
Rain Gauge (Precipitation)
Met Station (Air Temp, Wind, Solar Rad.)
Soil Moisture Sensors (Water Content, Ground Temp)
SHEPARD BROOK (SHP)
Turbidity Sensor (Turbidity, Water Temp, Variance)
Isco Pressure Transducer (Stage)
HIGH BRIDGE BROOK (HBR)
Turbidity Sensor (Turbidity, Water Temp, Variance)
Levelogger Pressure Transducer (Stage)
Isco Pressure Transducer (Stage)
MILL BROOK / IRASVILLE WEATHER (MIL)
Turbidity Sensor (Turbidity, Water Temp, Variance)
Isco Pressure Transducer (Stage)
Rain Gauge (Precipitation)
iButton (Air Temp)
SUGARBUSH WEATHER (SUG)
Rain Gauge (Precipitation)
iButton (Air Temp)
FOLSOM BROOK (FOL)
Turbidity Sensor (Turbidity, Water Temp, Variance)
Isco Pressure Transducer (Stage)
EAST WARREN WEATHER (WDE)
Rain Gauge (Precipitation)
iButton (Air Temp)
FREEMAN BROOK / WARREN WEATHER (FRE)
Turbidity Sensor (Turbidity, Water Temp, Variance)
Isco Pressure Transducer (Stage)
Rain Gauge (Precipitation)
4/8
11/25
5/2
8/10
Logger malfunction
7/31
Full
8/18
9/30
Logger Full
10/22
5/3
6/6
Slugs
6/13
7/14
Logger Malfunctic
8/6
7/24
6/20
7/11
5/31
7/3
Storm
6/18
5/15
Battery
5/28
5/2
Logger Malfunction
6/13 - New Gauge
Fell off post
Blocked
11/26
No rain data due to lost gauge, and repeated logger failures

Mad River Watershed Data Summary **2012 Field Season**

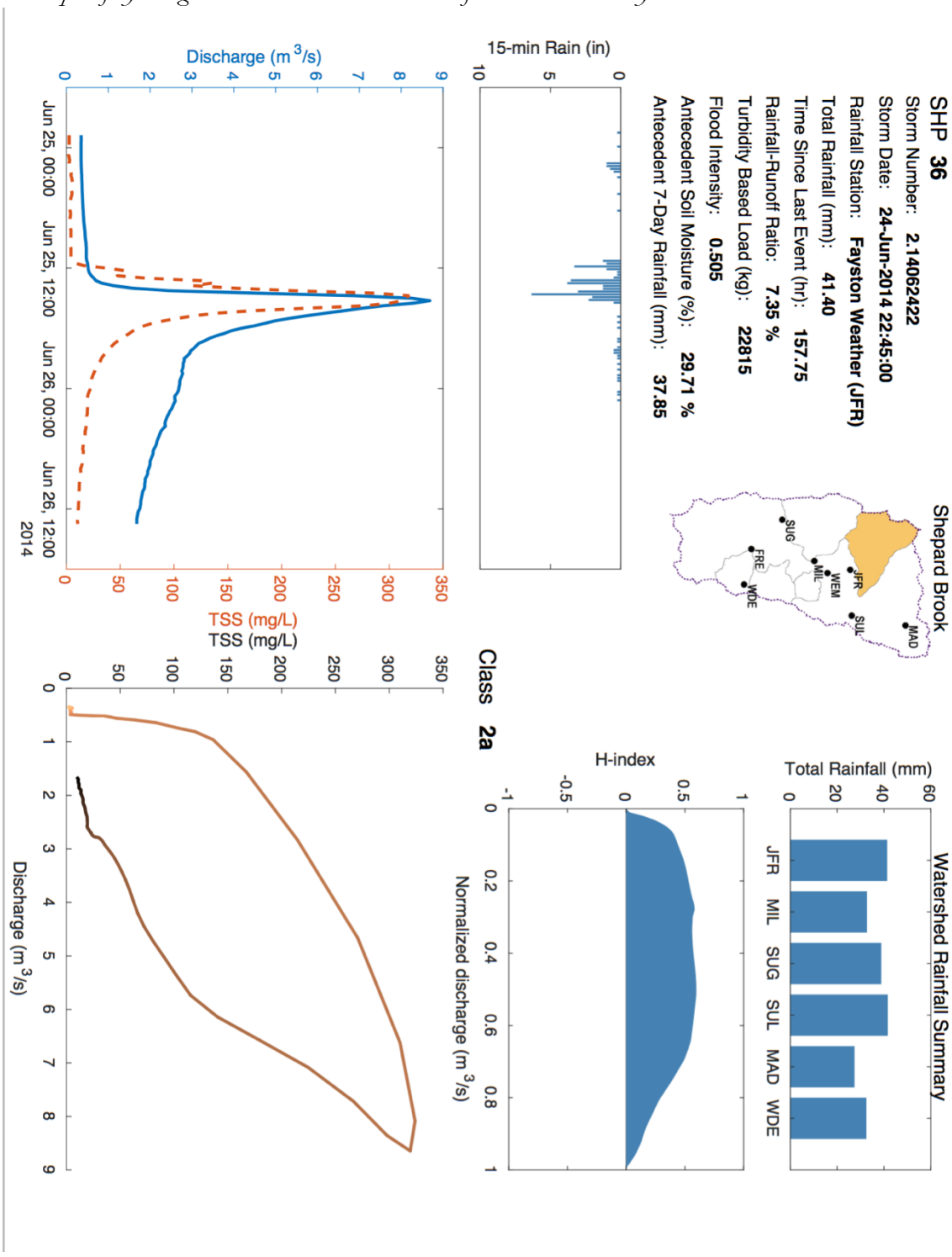
Last Updated: 2/27/16

Site Name	Site Location	Site Code				Short Code
Mad River / Moretown Weather	**Sainsbury Property**	**WR_MadRvr_0428800 / WR_MoretownMet_544**				**MAD**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	10/26/12	12/14/12	Battery Failure from 11/19 - 11/20, after 12/2/12
Water Temperature	15-min	°C	DTS-12	10/26/12	12/14/12	Battery Failure from 11/19 - 11/20, after 12/2/12
Stage	15-min	ft	USGS Gauge	N/A	N/A	
Discharge	15-min	cfs	USGS Gauge	N/A	N/A	
Precipitation	0.01 in	in	USGS Gauge	N/A	N/A	
South Hill Weather	**Sullivan Property**	**WR_SHillMet_1410**				**SUL**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Precipitation	0.01 in	in	Onset Rain Gauge	9/28/12	12/14/12	
Air Temperature	30-min	°C	iButton thermochron	11/7/12	12/14/12	
Mill Brook / Irasville Weather	**Mad River Massage**	**WR_MillBk_715 / WR_IrasvilleMet_715**				**MIL**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Turbidity	15-min	NTU	DTS-12	10/5/12	12/2/12	Battery Failure 10/14 - 10/18, bad data 10/22 - 10/30
Water Temperature	15-min	°C	DTS-12	10/5/12	12/2/12	
Stage	15-min	m	Isco Module	10/12/12	12/2/12	
Air Temperature	30-min	°C	iButton thermochron	9/28/12	12/14/12	Logger full after 11/23/12
Precipitation	0.01 in	in	Onset Rain Gauge	9/28/12	12/14/12	
Sugarbush Weather	**Sugarbush**	**WR_SgrbshMet_2170**				**SUG**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Air Temperature	30-min	°C	iButton thermochron	10/28/12	12/14/12	Logger full after 12/10/12
Precipitation	0.01 in	in	Onset Rain Gauge	10/28/12	12/14/12	No data - logger error
Freeman Brook /Warren Weather	**Perot Property**	**WR_FreeBk_895 / WR_WarrenMet_895**				**FRE**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Air Temperature	30-min	°C	iButton thermochron	10/12/12	12/14/12	Logger full after 11/23/12
Precipitation	0.01 in	in	Onset Rain Gauge	10/12/12	12/14/12	Rain gauge did not function
East Warren Weather	**Wade Property**	**WR_EWarrenMet_1880**				**WDE**
Parameter	**Measurement Interval**	**Units**	**Equipment**	**Date Start**	**Date End**	**Notes**
Air Temperature	30-min	°C	iButton thermochron	10/7/12	12/14/12	No data until 11/27
Precipitation	0.01 in	in	Onset Rain Gauge	10/7/12	12/14/12	No data - logger error

Mad River 2012 Field Season Data Summary



Example of hydrological event data dashboard used for storm event analysis



APPENDIX B: STAGE-DISCHARGE RELATIONSHIPS

Stage-discharge rating curves were utilized to estimate discharge at all five study tributaries in the Mad River watershed. Two different methods were used to generate rating curves. For tributaries with long-term monitoring (Mill Brook, Shepard Brook, and Folsom Brook) HEC-RAS models were developed and calibrated. Discharge for High Bridge Brook and Freeman Brook was estimated using a drainage-area ratio method based on discharge at the Mad River outlet.

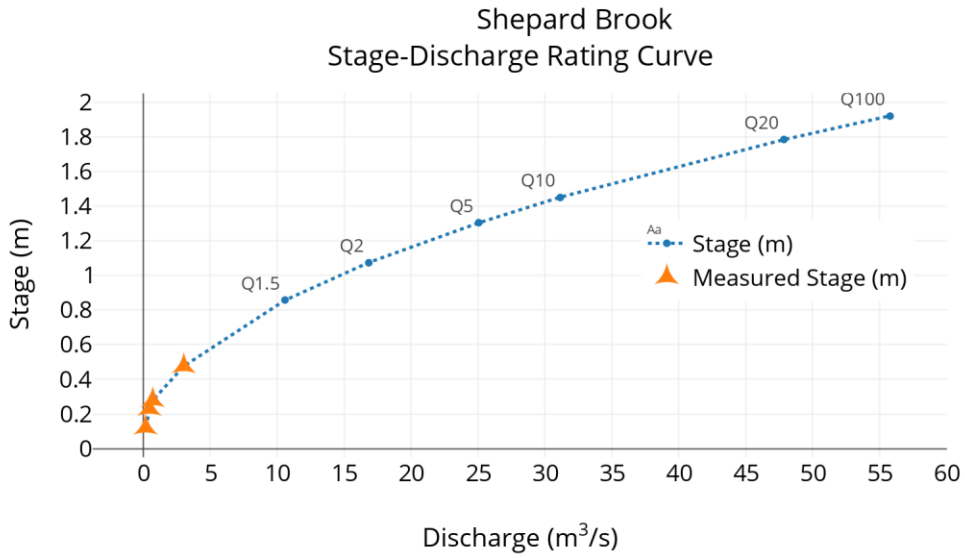
HEC-RAS model based rating curves

A HEC-RAS model (version 4.1) was developed for the stream reach containing the gauging station, and used to generate a theoretical rating curve. The model was then calibrated to measured discharge by adjusting Manning's n coefficient in the HEC-RAS model. Surveys were performed for all three tributaries in summer 2014 by Alex Morton, supported by Beverley Wemple, Kristen Underwood, Scott Hamshaw, and Jordan Duffy. HEC-RAS models were built by Alex Morton and Jordan Duffy. Discharge measurements were made by Kristen Underwood during 2013 and 2014 using the velocity-area method. Because discharge measurements were restricted to only wadeable conditions, the model calibration at higher flows may not be very reliable. Model calibration was performed by Scott Hamshaw.

Drainage-area ratio rating curves

Stage measurements were compared to time delayed measurements of instantaneous discharge at the Mad River outlet and a regression model was fit. The time delay was determined by observing the typical time lag between peak flow in the tributary to that of the Mad River at the USGS gauge. Log-linear regression and quadratic models were explored to find the best model fit.

Shepard Brook Rating Curve

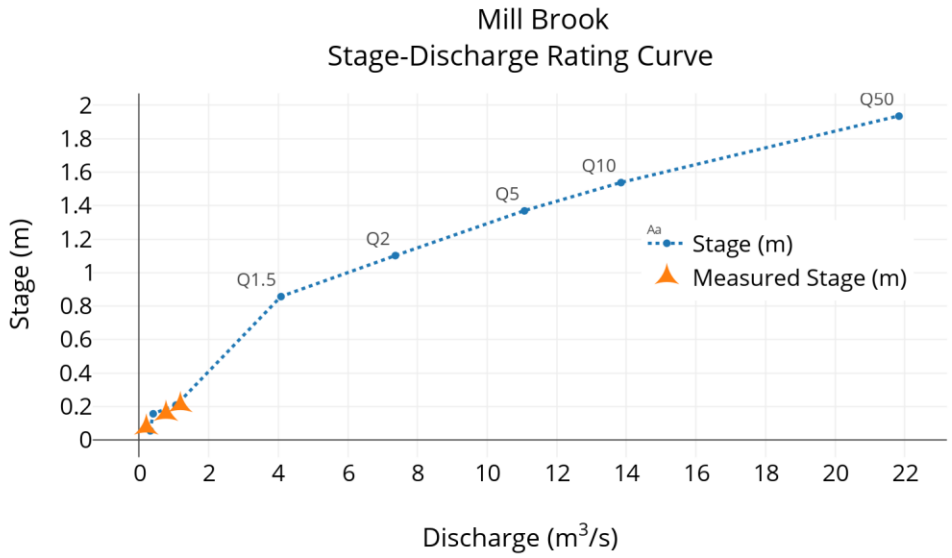


Shepard Brook Discharge Measurements

Flow measurements by Kristen Underwood

Date	Adjusted Sonde Stage (m)	Measured Discharge (ft^3/s)	Measured Discharge (m^3/s)
11/6/13	0.231	16.6	0.469
11/9/13	0.28	25.2	0.713
11/18/13	0.4765	107.1	3.032
7/23/14	0.122	6.6	0.187

Mill Brook Rating Curve

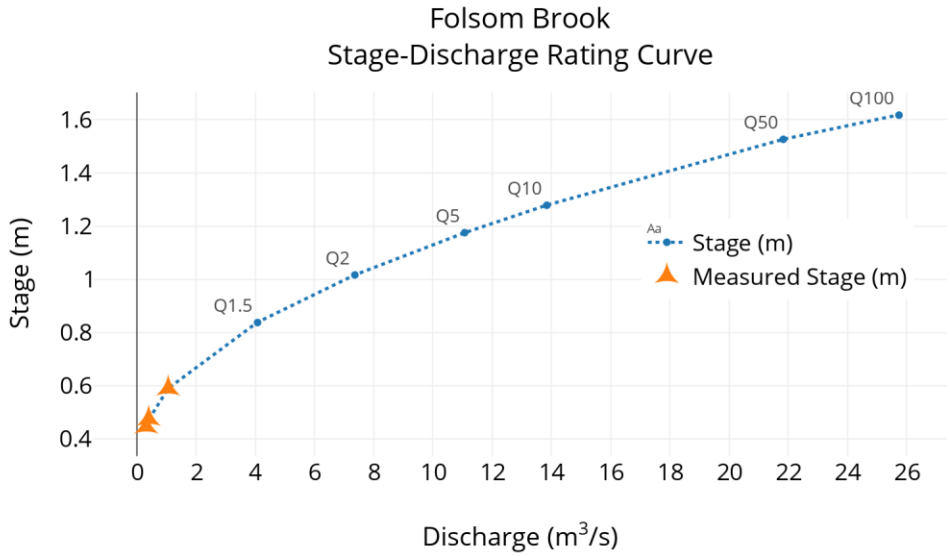


Mill Brook Discharge Measurements

Flow measurements made Kristen Underwood

Date	Adjusted Sonde Stage (m)	Measured Discharge (ft^3/s)	Measured Discharge (m^3/s)
11/6/13	0.157	27.28	0.773
11/9/13	0.209	41.84	1.185
7/23/14	0.075	7.32	0.207

Folsom Brook Rating Curve



Folsom Brook Discharge Measurements

Flow measurements by Kristen Underwood

Date	Adjusted Sonde Stage (m)	Measured Discharge (ft^3/s)	Measured Discharge (m^3/s)
11/2/13	0.475	14.3	0.40
11/9/13	0.446	11.4	0.32
11/18/13	0.589	37.4	1.06

High Bridge Brook Rating Curve

Log-linear regression model:

$$Q = BCF \cdot a \cdot H^b + x$$

$$Q\ (cfs) = 1.0894 \cdot 27.12 \cdot H^{1.5318} - 0.3$$

Freeman Brook Rating Curve

Quadratic regression model:

$$Q = a_1 H^2 + a_2 H + a_3$$

$$Q\ (cfs) = -4.8H^2 + 83H - 10$$

APPENDIX C: CODE FOR MATLAB FUNCTIONS

Streambank Cross-section Analysis Scripts

```
% program to view bank cross sections as raw point cloud data and as
% derived bank ground surface

clc;
close all;
clear all;

UAS = load('../Data/UAS/SB/XS1_UAS_4may2016');
TLS = load('../Data/UAS/SB/XS1_TLS_4may2016');
binSizeUAS = 0.10; %in meters
binSizeTLS = 0.05;
xsUASData = sortrows(UAS.xsData,1);
xsTLSData = sortrows(TLS.xsData,1);
xUAS = xsUASData(:,1);
xTLS = xsTLSData(:,1);
elevUAS = xsUASData(:,2);
elevTLS = xsTLSData(:,2);

startXS = 14;
endXS = max(xUAS);
ind = find(xUAS > endXS | xUAS < startXS);
xUAS(ind) = []; elevUAS(ind)=[];
ind = find(xTLS > endXS | xTLS < startXS);
xTLS(ind) = []; elevTLS(ind)=[];
count = 1;
for i = startXS:binSizeUAS:endXS
    ind = find((xUAS>=i) & (xUAS<(i+binSizeUAS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevUAS(ind));
        minZUAS(count) = M;
        temp = xUAS(ind);
        LUAS(count) = temp(I);
        count = count+1;
    end

end


count = 1;

for i = startXS:binSizeTLS:endXS
    ind = find((xTLS>=i) & (xTLS<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevTLS(ind));
        minZTLS(count) = M;
        temp = xTLS(ind);
        LTLS(count) = temp(I);
```

```
        count = count+1;
    end

end

figure()
plot(xTLS,elevTLS,'.c')
hold on
plot(xUAS,elevUAS,'.b','MarkerSize',10)
%axis('equal')
xlabel('Distance Along XS (m)');
ylabel('Elevation (m)');
hold off;

figure()
plot(LUAS,minZUAS,'--k','LineWidth',2)
hold on
plot(LTLS,minZTLS,'-b','LineWidth',2')
%axis('equal')
xlabel('Distance Along XS (m)');
ylabel('Elevation (m)');
hold off;

% program to estiamte bank surface from point cloud data along
% cross-sections and compare surfaces from two separate surveys

clc;
close all;
clear all;

UAS = load('../Data/UAS/SB/XS1_UAS_4may2016');
TLS = load('../Data/UAS/SB/XS1_TLS_4may2016');
binSizeUAS = 0.2; %in meters
binSizeTLS = .05;
startXS = 14;
endXS = 21;
xCorrTLS = 0;
yCorrTLS = 0.00;
xCorrUAS = 0.00;
yCorrUAS = 0;

xsUASData = sortrows(UAS.xsData,1);
xsTLSData = sortrows(TLS.xsData,1);
xUAS = xsUASData(:,1) + xCorrUAS;
xTLS = xsTLSData(:,1) + xCorrTLS;
elevUAS = xsUASData(:,2) + yCorrUAS;
elevTLS = xsTLSData(:,2) + yCorrTLS;

ind = find(xUAS > endXS | xUAS < startXS);
xUAS(ind) = []; elevUAS(ind)=[];
ind = find(xTLS > endXS | xTLS < startXS);
xTLS(ind) = []; elevTLS(ind)=[];
count = 1;
for i = startXS:binSizeUAS:endXS
```

```
    ind = find((xUAS>=i) & (xUAS<(i+binSizeUAS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevUAS(ind));
        minZUAS(count) = M;
        temp = xUAS(ind);
        LUAS(count) = temp(I);
        count = count+1;
    end

end


count = 1;

for i = startXS:binSizeTLS:endXS
    ind = find((xTLS>=i) & (xTLS<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevTLS(ind));
        minZTLS(count) = M;
        temp = xTLS(ind);
        LTLS(count) = temp(I);
        count = count+1;
    end

end
count = 1;
for i = startXS:binSizeUAS:endXS
    vertError(count) = interp1(LUAS,minZUAS,i)-interp1(LTLS,minZTLS,i);
    count = count+1;
end
meanVertError = mean(vertError,'omitNaN')
medianVertError = median(vertError,'omitNaN')
stdevVertError = std(vertError,'omitNaN')
rmseVert = sqrt(mean((vertError).^2,'omitNaN'))

startZ = min([interp1(LUAS,minZUAS,startXS+binSizeUAS)
interp1(LTLS,minZTLS,startXS+binSizeUAS)],[],'omitNaN');
endZ = max([interp1(LUAS,minZUAS,endXS-binSizeUAS) interp1(LTLS,minZTLS,endXS-
binSizeUAS)]);
count = 1;
for i = startZ:binSizeTLS:endZ
    ind = find((elevTLS>=i) & (elevTLS<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = max(xTLS(ind));
        maxXTLS(count) = M;
        temp = elevTLS(ind);
        YTLS(count) = temp(I);
        count = count+1;
    end
end
count = 1;
```

```
for i = startZ:binSizeUAS:endZ
    ind = find((elevUAS>=i) & (elevUAS<(i+binSizeUAS)));
    tf = isempty(ind);
    if tf==0
        [M I] = max(xUAS(ind));
        maxXUAS(count) = M;
        temp = elevUAS(ind);
        YUAS(count) = temp(I);
        count = count+1;
    end
end

count = 1;
for i = startZ:binSizeUAS:endZ
     horizError(count) = interp1(YTLS,maxXTLS,i)-interp1(YUAS,maxXUAS,i);
     count = count+1;
end
meanHorizError = mean(horizError,'omitNaN')
medianHorizError = median(horizError,'omitNaN')
stdevHorizError = std(horizError,'omitNaN')
rmseHoriz = sqrt(mean((horizError).^2,'omitNaN'))

ind = find(xTLS > endXS | xTLS < startXS);
xTLS(ind) = []; elevTLS(ind)=[];
ind = find(xUAS > endXS | xUAS < startXS);
xUAS(ind) = []; elevUAS(ind)=[];
xUAS = xUAS-startXS;
xTLS = xTLS-startXS;
LUAS = LUAS-startXS;
LTLS = LTLS-startXS;
maxXTLS = maxXTLS-startXS;
maxXUAS = maxXUAS-startXS;

figure()
plot(xUAS,elevUAS,'.k','LineWidth',2,'MarkerSize',15)
hold on;
plot(xTLS,elevTLS,'.m','MarkerSize',15)
plot(maxXTLS,YTLS,'-b','LineWidth',2)
plot(maxXUAS,YUAS,'--g','LineWidth',2)
hold off
title('Vertical bin derived bank profiles');
xlabel('Distance along XS (m)');
ylabel('Elevation (m)');
legend('raw UAS data','raw TLS data','UAS bank profile','TLS bank
profile','Location','SouthEast');
axis('equal')

figure()
plot(xUAS,elevUAS,'.k','LineWidth',2,'MarkerSize',15)
hold on;
plot(xTLS,elevTLS,'.m','MarkerSize',15)
plot(LUAS,minZUAS,'--g','LineWidth',2,'MarkerSize',15)
plot(LTLS,minZTLS,'-b','LineWidth',2')
axis('equal')
```

```
title('UAS Comparison MR-D');
xlabel('Distance along XS (m)');
ylabel('Elevation (m)');
legend('Raw UAS Data Flight 1','Raw UAS Data Flight 2','UAS Bank Profile
Flight 1','UAS Bank Profile Flight 2','Location','SouthEast');
hold off;
%fig2plotly()

% program to calculate erosion along streambank cross-section as change in
% area. Compares changes in area between two separate surveys

clc;
close all;
clear all;

UAS1 = load('../Data/UAS/NHR/Area2_XS3_UAS_12-22-15');
UAS2 = load('../Data/UAS/NHR/Area2_XS3_UAS_4-27-16');
TLS1 = load('../Data/UAS/NHR/Area2_XS3_TLS_12-14-15');
TLS2 = load('../Data/UAS/NHR/Area2_XS3_TLS_4-27-16');
binSizeUAS = 0.18; %in meters
binSizeTLS = .15;
startXS = 21.0;
endXS = 34;
xCorrTLS = 0;
yCorrTLS = 0;
xCorrUAS = 0;
yCorrUAS = 0;

xsUASData1 = sortrows(UAS1.xsData,1);
xsUASData2 = sortrows(UAS2.xsData,1);
xsTLSData1 = sortrows(TLS1.xsData,1);
xsTLSData2 = sortrows(TLS2.xsData,1);
xUAS1 = xsUASData1(:,1) + xCorrUAS;
xUAS2 = xsUASData2(:,1) + xCorrUAS;
xTLS1 = xsTLSData1(:,1) + xCorrTLS;
xTLS2 = xsTLSData2(:,1) + xCorrTLS;
elevUAS1 = xsUASData1(:,2) + yCorrUAS;
elevUAS2 = xsUASData2(:,2) + yCorrUAS;
elevTLS1 = xsTLSData1(:,2) + yCorrTLS;
elevTLS2 = xsTLSData2(:,2) + yCorrTLS;

ind = find(xUAS1 > endXS | xUAS1 < startXS);
xUAS1(ind) = []; elevUAS1(ind)=[];
ind = find(xUAS2 > endXS | xUAS2 < startXS);
xUAS2(ind) = []; elevUAS2(ind)=[];
ind = find(xTLS1 > endXS | xTLS1 < startXS);
xTLS1(ind) = []; elevTLS1(ind)=[];
ind = find(xTLS2 > endXS | xTLS2 < startXS);
xTLS2(ind) = []; elevTLS2(ind)=[];
count = 1;
for i = startXS:binSizeUAS:endXS
    ind = find((xUAS1>=i) & (xUAS1<(i+binSizeUAS)));
    tf = isempty(ind);
    if tf==0
```

```
        [M I] = min(elevUAS1(ind));
        minZUAS1(count) = M;
        temp = xUAS1(ind);
        LUAS1(count) = temp(I);
        count = count+1;
    end

end


count = 1;

for i = startXS:binSizeTLS:endXS
    ind = find((xTLS1>=i) & (xTLS1<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevTLS1(ind));
        minZTLS1(count) = M;
        temp = xTLS1(ind);
        LTLS1(count) = temp(I);
        count = count+1;
    end

end
count = 1;

for i = startXS:binSizeTLS:endXS
    ind = find((xTLS2>=i) & (xTLS2<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevTLS2(ind));
        minZTLS2(count) = M;
        temp = xTLS2(ind);
        LTLS2(count) = temp(I);
        count = count+1;
    end

end
count = 1;

for i = startXS:binSizeUAS:endXS
    ind = find((xUAS2>=i) & (xUAS2<(i+binSizeUAS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevUAS2(ind));
        minZUAS2(count) = M;
        temp = xUAS2(ind);
        LUAS2(count) = temp(I);
        count = count+1;
    end

end

count = 1;
```

```
for i = startXS:binSizeUAS:endXS
    vertErrorUAS(count) = interp1(LUAS2,minZUAS2,i)-interp1(LUAS1,minZUAS1,i);
    count = count+1;
end
ind = isnan(vertErrorUAS);
vertErrorUAS(ind) = [];
meanVertRetreatUAS = mean(vertErrorUAS,'omitNaN')
medianVertRetreatUAS = median(vertErrorUAS,'omitNaN')
stdevVertRetreatUAS = std(vertErrorUAS,'omitNaN')
temp = vertErrorUAS < 0;
erosionVertAreaUAS = sum(vertErrorUAS(temp)*binSizeUAS)
depositionVertAreaUAS = sum(vertErrorUAS(~temp)*binSizeUAS)
netRetreatVertAreaUAS = meanVertRetreatUAS*(endXS-startXS)

startZ = max([interp1(LUAS1,minZUAS1,startXS+binSizeUAS)
interp1(LTLS1,minZTLS1,startXS+binSizeUAS)],[],'omitNaN');
endZ = min([interp1(LUAS1,minZUAS1,endXS-binSizeUAS)
interp1(LTLS1,minZTLS1,endXS-binSizeUAS)]);
%endZ = 108;
count = 1;
for i = startZ:binSizeTLS:endZ
    ind = find((elevTLS1>=i) & (elevTLS1<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = max(xTLS1(ind));
        maxXTLS1(count) = M;
        temp = elevTLS1(ind);
        YTLS1(count) = temp(I);
        count = count+1;
    end
end
count = 1;
for i = startZ:binSizeTLS:endZ
    ind = find((elevTLS2>=i) & (elevTLS2<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = max(xTLS2(ind));
        maxXTLS2(count) = M;
        temp = elevTLS2(ind);
        YTLS2(count) = temp(I);
        count = count+1;
    end
end
count = 1;
for i = startZ:binSizeUAS:endZ
    ind = find((elevUAS1>=i) & (elevUAS1<(i+binSizeUAS)));
    tf = isempty(ind);
    if tf==0
        [M I] = max(xUAS1(ind));
        maxXUAS1(count) = M;
        temp = elevUAS1(ind);
        YUAS1(count) = temp(I);
        count = count+1;
    end
```

```
end
count = 1;
for i = startZ:binSizeUAS:endZ
    ind = find((elevUAS2>=i) & (elevUAS2<(i+binSizeUAS)));
    tf = isempty(ind);
    if tf==0
        [M I] = max(xUAS2(ind));
        maxXUAS2(count) = M;
        temp = elevUAS2(ind);
        YUAS2(count) = temp(I);
        count = count+1;
    end
end
count = 1;
for i = startZ:binSizeTLS:endZ
     horizErrorTLS(count) = interp1(YTLS1,maxXTLS1,i)-
interp1(YTLS2,maxXTLS2,i);
     count = count+1;
end
count = 1;
for i = startZ:binSizeUAS:endZ
     horizErrorUAS(count) = interp1(YUAS1,maxXUAS1,i)-
interp1(YUAS2,maxXUAS2,i);
     count = count+1;
end
ind = isnan(horizErrorTLS);
horizErrorTLS(ind) = [];
meanRetreat = mean(horizErrorTLS,'omitNaN')
medianRetreat = median(horizErrorTLS,'omitNaN')
stdevRetreat = std(horizErrorTLS,'omitNaN')
temp = horizErrorTLS < 0;
erosionArea = sum(horizErrorTLS(temp)*binSizeTLS)
depositionArea = sum(horizErrorTLS(~temp)*binSizeTLS)
netRetreatArea = meanRetreat*(endZ-startZ) % square meters

ind = isnan(horizErrorUAS);
horizErrorUAS(ind) = [];
meanRetreatUAS = mean(horizErrorUAS,'omitNaN')
medianRetreatUAS = median(horizErrorUAS,'omitNaN')
stdevRetreatUAS = std(horizErrorUAS,'omitNaN')
temp = horizErrorUAS < 0;
erosionAreaUAS = sum(horizErrorUAS(temp)*binSizeUAS)
depositionAreaUAS = sum(horizErrorUAS(~temp)*binSizeUAS)
netRetreatAreaUAS = meanRetreatUAS*(endZ-startZ)
percError = abs(netRetreatAreaUAS-netRetreatArea)/netRetreatArea

% ind = find(xTLS > endXS | xTLS < startXS);
% xTLS1(ind) = []; elevTLS(ind)=[];
% ind = find(xUAS > endXS | xUAS < startXS);
% xUAS(ind) = []; elevUAS(ind)=[];
xUAS1 = xUAS1-startXS;
xUAS2 = xUAS2-startXS;
xTLS1 = xTLS1-startXS;
maxXTLS1 = maxXTLS1-startXS;
```

```
maxXUAS1 = maxXUAS1-startXS;
maxXTLS2 = maxXTLS2-startXS;
maxXUAS2 = maxXUAS2-startXS;

figure()
plot(xTLS1,elevTLS1,'.c','MarkerSize',14)
hold on
plot(xTLS2,elevTLS2,'.m','MarkerSize',14)
hold off
legend('TLS Date 1','TLS Date 2');

figure()
plot(xUAS1,elevUAS1,'.c','MarkerSize',14)
hold on
plot(xUAS2,elevUAS2,'.m','MarkerSize',14)
hold off
legend('UAS Date 1','UAS Date 2');

figure()
plot(maxXTLS1,YTLS1,'-b','LineWidth',2)
hold on
plot(maxXUAS1,YUAS1,'--g','LineWidth',2)
plot(maxXTLS2,YTLS2,'-k','LineWidth',2)
plot(maxXUAS2,YUAS2,'--m','LineWidth',2)
hold off
title('UAS and TLS Measured Bank Horizontal Retreat');
xlabel('Distance along XS (m)');
ylabel('Elevation (m)');
legend('TLS Date 1','UAS Date 1','TLS Date 2','UAS Date 2');
%axis('equal')
fig2plotly()

figure()
plot(LUAS1,minZUAS1,'--g','LineWidth',2)
hold on
plot(LTLS1,minZTLS1,'-b','LineWidth',2')
plot(LUAS2,minZUAS2,'--m','LineWidth',2)
plot(LTLS2,minZTLS2,'-k','LineWidth',2')
axis('equal')
title('UAS and TLS Vertical Erosion Measured Retreat');
xlabel('Distance along XS (m)');
ylabel('Elevation (m)');
legend('UAS Date 1','TLS Date 1','UAS Date 2','TLS Date 2');
hold off;

% program to compare four surveys along streambank cross section

clc;
close all;
clear all;

UAS1 = load('../Data/UAS/MR-D/Area2_XS1_UAS_22apr2015');
UAS2 = load('../Data/UAS/MR-D/Area2_XS1_UAS_22jun2015');
UAS3 = load('../Data/UAS/MR-D/Area2_XS1_UAS_10nov2015');
```

```
UAS4 = load('../Data/UAS/MR-D/Area2_XS1_UAS_18may2016');
binSizeUAS = 0.2; %in meters
binSizeTLS = 0.1;
startXS = 8;
endXS = 14;
xCorrTLS = 0;
yCorrTLS = 0;
xCorrUAS = 0;
yCorrUAS =0;

xsUAS1Data = sortrows(UAS1.xsData,1);
xsUAS2Data = sortrows(UAS2.xsData,1);
xsUAS3Data = sortrows(UAS3.xsData,1);
xsUAS4Data = sortrows(UAS4.xsData,1);
xUAS1 = xsUAS1Data(:,1);
xUAS2 = xsUAS2Data(:,1);
xUAS3 = xsUAS3Data(:,1);
xUAS4 = xsUAS4Data(:,1);
elevUAS1 = xsUAS1Data(:,2);
elevUAS2 = xsUAS2Data(:,2);
elevUAS3 = xsUAS3Data(:,2)+.05;
elevUAS4 = xsUAS4Data(:,2);

ind = find(xUAS1 > endXS | xUAS1 < startXS);
xUAS1(ind) = []; elevUAS1(ind)=[];
ind = find(xUAS2 > endXS | xUAS2 < startXS);
xUAS2(ind) = []; elevUAS2(ind)=[];
ind = find(xUAS3 > endXS | xUAS3 < startXS);
xUAS3(ind) = []; elevUAS3(ind)=[];
ind = find(xUAS4 > endXS | xUAS4 < startXS);
xUAS4(ind) = []; elevUAS4(ind)=[];
count = 1;
for i = startXS:binSizeUAS:endXS
    ind = find((xUAS1>=i) & (xUAS1<(i+binSizeUAS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevUAS1(ind));
        minZUAS1(count) = M;
        temp = xUAS1(ind);
        LUAS1(count) = temp(I);
        count = count+1;
    end
end

count = 1;

for i = startXS:binSizeUAS:endXS
    ind = find((xUAS2>=i) & (xUAS2<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevUAS2(ind));
        minZUAS2(count) = M;
        temp = xUAS2(ind);
        LUAS2(count) = temp(I);
```

```
        count = count+1;
    end

end


count = 1;

for i = startXS:binSizeUAS:endXS
    ind = find((xUAS3>=i) & (xUAS3<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevUAS3(ind));
        minZUAS3(count) = M;
        temp = xUAS3(ind);
        LUAS3(count) = temp(I);
        count = count+1;
    end

end


count = 1;

for i = startXS:binSizeUAS:endXS
    ind = find((xUAS4>=i) & (xUAS4<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevUAS4(ind));
        minZUAS4(count) = M;
        temp = xUAS4(ind);
        LUAS4(count) = temp(I);
        count = count+1;
    end

end

figure()
plot(LUAS1-startXS,minZUAS1,'--k','LineWidth',2)
hold on;
plot(LUAS2-startXS,minZUAS2,'-.m','LineWidth',2)
plot(LUAS3-startXS,minZUAS3,':g','LineWidth',2)
plot(LUAS4-startXS,minZUAS4,'-b','LineWidth',2')
axis('equal')
title('Comparison of UAS Data');
xlabel('Distance along XS (m)');
ylabel('Elevation (m)');
legend('Spring 2015','Summer 2015','Fall 2015','Spring 2016');
hold off;
fig2plotly()

% program to compare ground survey control points to other surveys
% generated from point cloude data along streambank cross-sections
```

```
clc;
close all;
clear all;


UAS = load('../Data/UAS/SB/XS1_UAS_4may2016');
GPS = load('../Data/UAS/SB/XS1_GPS_4may2016');
TLS = load('../Data/UAS/SB/XS1_TLS_4may2016');
site = 'SB 2016';

xsGPSData = sortrows(GPS.xsData,1);
xGPS = xsGPSData(:,1);
elevGPS = xsGPSData(:,2);
numGPS = length(xGPS);
binSizeUAS = 0.20;
binSizeTLS = .05;

switch site
    case 'SB 2015'
        str1 = 'Site: SB Fall 2015';
        startGPS = 6;
        endGPS = 12;
        xCorrTLS = 0.1;
        yCorrTLS = 0;
        xCorrUAS = 0;
        yCorrUAS = 0;
        xlims = [14 21];
        startXS = 14; endXS = 21;
    case 'SB 2016'
        str1 = 'Site: SB Spring 2016';
        startGPS = 4;
        endGPS = numGPS;
        xCorrTLS = 0;
        yCorrTLS = 0;
        xCorrUAS = 0;
        yCorrUAS = 0;
        xlims = [14 21];
        startXS = 14; endXS = 21;
    case 'NHR 2016'
        str1 = 'Site: NHR Spring 2016';
        startGPS = 6;
        endGPS = numGPS;
        xCorrTLS = 0;
        yCorrTLS = 0;
        xCorrUAS = 0;
        yCorrUAS = 0;
        xlims = [8 20];
        startXS = 8;
        endXS = 20;
    case 'NHR 2015'
        str1 = 'Site: NHR Fall 2015';
        startGPS = 1;
        endGPS = numGPS;
        xCorrTLS = 0;
```

```
    yCorrTLS = 0;
    xCorrUAS = 0;
    yCorrUAS = 0;
    xlims = [8 20];
    startXS = 8; endXS = 20;
case 'MR-A 2015'
    str1 = 'Site: MR-A Fall 2015';
    startGPS = 2;
    endGPS = 14;
    xCorrTLS = 0;
    yCorrTLS = 0;
    xCorrUAS = 0;
    yCorrUAS = 0;
    xlims = [0 8];
    startXS = 0;
    endXS = 8;
case 'MR-A 2016'
    str1 = 'Site: MR-A Spring 2016';
    startGPS = 3;
    endGPS = 11;
    xCorrTLS = 0;
    yCorrTLS = 0;
    xCorrUAS = 0;
    yCorrUAS = 0;
    xlims = [0 8];
    startXS = 0;
    endXS = 8;
case 'MR-B'
    str1 = 'Site: MR-B Fall 2015';
    startGPS = 2;
    endGPS = 5;
    xCorrTLS = 0;
    yCorrTLS = 0;
    xCorrUAS = 0;
    yCorrUAS = 0;
    xlims = [6 16];
    startXS = 6; endXS = 16;
case 'MR-C'
    str1 = 'Site: MR-C Fall 2015';
    startGPS = 3;
    endGPS = 9;
    xCorrTLS = .2;
    yCorrTLS = -.05;
    xCorrUAS = 0;
    yCorrUAS = 0;
    xlims = [8 16];
    startXS = 8;
    endXS = 16;
case 'MR-D'
    str1 = 'Site: MR-D Fall 2015';
    startGPS = 20;
    endGPS = 30;
    xCorrTLS = 0;
    yCorrTLS = -.1;
```

```
        xCorrUAS = 0;
        yCorrUAS = .2;
        xlims = [70 80];
        startXS =70;
        endXS = 80;
end

xsUASData = sortrows(UAS.xsData,1);

xsTLSData = sortrows(TLS.xsData,1);

xUAS = xsUASData(:,1) + xCorrUAS;
xTLS = xsTLSData(:,1) + xCorrTLS;
elevUAS = xsUASData(:,2) + yCorrUAS;
elevTLS = xsTLSData(:,2) + yCorrTLS;

count = 1;
for i = startXS:binSizeUAS:endXS
    ind = find((xUAS>=i) & (xUAS<(i+binSizeUAS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevUAS(ind));
        minZUAS(count) = M;
        temp = xUAS(ind);
        LUAS(count) = temp(I);
        count = count+1;
    end

end

count = 1;

for i = startXS:binSizeTLS:endXS
    ind = find((xTLS>=i) & (xTLS<(i+binSizeTLS)));
    tf = isempty(ind);
    if tf==0
        [M I] = min(elevTLS(ind));
        minZTLS(count) = M;
        temp = xTLS(ind);
        LTLS(count) = temp(I);
        count = count+1;
    end

end
% startGPS = 1;
% endGPS = numGPS;

for i = startGPS:endGPS
    for j = 1:length(LUAS)
        distUAS(j) = sqrt((LUAS(j)-xGPS(i))^2 + (minZUAS(j)-elevGPS(i))^2);
    end
    nearestUAS(i) = min(distUAS);
    clear distUAS
    for j = 1:length(LTLS)
```

```
        distTLS(j) = sqrt((LTLS(j)-xGPS(i))^2 + (minZTLS(j)-elevGPS(i))^2);
    end
    nearestTLS(i) = min(distTLS);
    clear distTLS
end


meanUAS = mean(nearestUAS)
medianUAS = median(nearestUAS)
stdevUAS = std(nearestUAS)
RMSEUAS = sqrt(mean((nearestUAS).^2))
meanTLS = mean(nearestTLS)
medianTLS = median(nearestTLS)
stdevTLS = std(nearestTLS)
RMSETLS = sqrt(mean((nearestTLS).^2))



ind = find(xTLS > endXS | xTLS < startXS);
xTLS(ind) = []; elevTLS(ind)=[];
ind = find(xUAS > endXS | xUAS < startXS);
xUAS(ind) = []; elevUAS(ind)=[];
ind = find(xGPS > endXS | xGPS < startXS);
xGPS(ind) = []; elevGPS(ind)=[];
xUAS = xUAS-startXS;
xTLS = xTLS-startXS;
xGPS = xGPS-startXS;
LUAS = LUAS-startXS;
LTLS = LTLS-startXS;

figure()
plot(xTLS,elevTLS,'.c')
hold on
plot(xUAS,elevUAS,'.b','MarkerSize',10)
plot(xGPS,elevGPS,'*r');
plot(LUAS,minZUAS,'--m');
plot(LTLS,minZTLS,'--k');
title(str1);
xlabel('Distance Along XS (m)');
ylabel('Elevation (m)');
hold off;
%fig2plotly()
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program to match TSS samples to turbidity sensor data
% and generate power law regressions
% Scott Hamshaw
% Created: 19-Mar-2016
% Last Modified:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% NOTES: Enter SiteCode desired (MAD = Mad River, FOL = Folsom Brook, FRE =
% Freeman Brook, HBR = High Bridge Brook, SHP = Shepard Brook, MIL = Mill
% Brook, DOW = Dowsville Brook
% Enter water quality parameters desired (TSS, TP, SRP, TDP, NOX, NH3, TN,
TDN)

clc; clear all; close all;
disp('Program Started');
tic;
addpath('../Stats');
addpath('../Plotting');
c = @cmu.colors;
set(0,'DefaultAxesFontSize',14)

% Choose Site to run
SiteCode = 'MAD';

% Choose Parameter to run
WQParam = 'TSS'

disp(strcat('Site selected for analysis = ',SiteCode));
% Setup for Mad River
switch SiteCode
    case 'MAD'
        load('../Data/Mad/SensorData/MAD_Sensor_2012-2015_all');
        load('../Data/Mad/WQSamples/MAD_WQ_2012-2015');
        str1 = 'Mad River';
    case 'FOL'
        load('../Data/Mad/SensorData/FOL_Sensor_2013-2015_all');
        load('../Data/Mad/WQSamples/FOL_WQ_2013-2015');
        str1 = 'Folsom Brook';
    case 'FRE'
        load('../Data/Mad/SensorData/FRE_Sensor_2013_all');
        load('../Data/Mad/WQSamples/FRE_WQ_2013');
        str1 = 'Freeman Brook';
    case 'MIL'
        load('../Data/Mad/SensorData/MIL_Sensor_2012-2015_all');
        load('../Data/Mad/WQSamples/MIL_WQ_2013-2015');
        str1 = 'Mill Brook';
    case 'SHP'
        load('../Data/Mad/SensorData/SHP_Sensor_2013-2015_all');
        load('../Data/Mad/WQSamples/SHP_WQ_2013-2015');
        str1 = 'Shepard Brook';
```

```
    case 'HBR'
        load('../Data/Mad/SensorData/HBR_Sensor_2013_all');
        load('../Data/Mad/WQSamples/HBR_WQ_2013');
        str1 = 'High Bridge Brook';
end

switch WQParam
    case 'TSS'
        WQVar = TSS;
        str3 = 'TSS (mg/L)'; % set parameter title and units for y-axis
    case 'TP'
        WQVar = TP*1000;
        str3 = 'TP (\mug/L)'; % set parameter title and units for y-axis
    case 'TDP'
        WQVar = TDP*1000;
        str3 = 'TDP (\mug/L)';
end

str2 = 'Turbidity (NTU)'; % set parameter title and units for x-axis

% sort date chronologically to make sure samples match up correctly
[B,I] = sort(numDateWQ);
numDateWQ = numDateWQ(I); matDateWQ = matDateWQ(I);
textDateWQ=textDateWQ(I); WQVar = WQVar(I);
WQExclude = WQExclude(I); WQHydroLimb = WQHydroLimb(I);
WQSampleType = WQSampleType(I); WQStormNum = WQStormNum(I);

turbidityTS = timeseries(turbidity,textDate)


% exclude bad data points and those missing turbidity data
ind = logical(logical(WQExclude==1)+logical(WQExclude==3)); %ind = [];
textDateWQ(ind)=[]; WQVar(ind)=[];numDateWQ(ind)=[];matDateWQ(ind)=[];
WQStormNum(ind)=[];WQSampleType(ind)=[];WQHydroLimb(ind)=[];
TSSTS = timeseries(WQVar,textDateWQ,'Name','WQ Samples');
turbMatchedTS = resample(turbidityTS,textDateWQ);

figure(1)
loglog(turbMatchedTS.Data,WQVar,'s','MarkerFaceColor','blue','MarkerEdgeColor'
,'black','MarkerSize',8);
title({str1, strcat(WQParam,' vs. Turbidity')},'FontSize',12);
xlabel(str2,'FontSize',12); ylabel(str3,'FontSize',12);
xlim([0.1 10000]); ylim([0.1 10000]);
ytic=get(gca,'YTick'); % Grab the y tick values
yticlab=num2str(ytic'); % Convert to string mx
set(gca,'YTickLabel',yticlab) % Re-label
xtic=get(gca,'XTick'); % Grab the y tick values
xticlab=num2str(xtic'); % Convert to string mx
set(gca,'XTickLabel',xticlab) % Re-label
hold off

[a,b,n,SF,rsq_adj] = powerFit(turbMatchedTS.Data,WQVar);

figure(2); % Plot TSS vs. Turbidity with Power Law Curve Fit on Log-Log scale
```

```
%subplot(1,3,1)
loglog(turbMatchedTS.Data,WQVar,'o','MarkerFaceColor',c('teal'),'MarkerEdgeCol
or',c('teal'),'MarkerSize',6);
%title({str1, strcat('Power Law Fit of ',WQParam,' vs. Turbidity')});
title(str1,'FontSize',26)
xlabel(str2,'FontSize',22); ylabel(str3,'FontSize',22);
hold on;
x = min(turbMatchedTS.Data):1:max(turbMatchedTS.Data);
y = SF*a*x.^b;
plot(x,y,'--k','LineWidth',2)
% str = {strcat(WQParam,sprintf(' = %.4f{} x
%.4f{}turbidity^{%.3f}',SF,a,b)),sprintf('n = %.0f{}',n),...
%    sprintf('r^{2}-adj = %.3f{}',rsq_adj)};
%
annotation(figure(2),'textbox',[0.5,0.15,0.40,0.15],'String',str,'EdgeColor','
none',...
%    'HorizontalAlignment','right','VerticalAlignment','baseline');
grid off;
xlim([0.1 10000]); ylim([0.1 10000]);
ytic=get(gca,'YTick'); % Grab the y tick values
yticlab=num2str(ytic'); % Convert to string mx
set(gca,'YTickLabel',yticlab) % Re-label
xtic=get(gca,'XTick'); % Grab the y tick values
xticlab=num2str(xtic'); % Convert to string mx
set(gca,'XTickLabel',xticlab,'FontSize',20) % Re-label
set(gca,'YTickLabel',yticlab,'FontSize',20) % Re-label

% separate rising limb vs. falling limb vs. baseflow samples
ind = find(WQHydroLimb==1); % rising limb samples
ind2 = find(WQHydroLimb==2); % falling limb Samples
ind3 = find(WQHydroLimb==3); % baseflow Samples
risingTurb = turbMatchedTS.Data(ind); risingTSS = WQVar(ind);
fallingTurb = turbMatchedTS.Data(ind2); fallingTSS = WQVar(ind2);
baseTurb = turbMatchedTS.Data(ind3); baseTSS = WQVar(ind3);

figure(3) % Plot TSS vs. Turbidity on log-log scale by hydrograph limb
location
loglog(risingTurb,risingTSS,'o','MarkerFaceColor','blue','MarkerEdgeColor','bl
ack','MarkerSize',8);
hold on;
loglog(fallingTurb,fallingTSS,'s','MarkerFaceColor','green','MarkerEdgeColor',
'black','MarkerSize',8);
loglog(baseTurb,baseTSS,'d','MarkerFaceColor','yellow','MarkerEdgeColor','blac
k','MarkerSize',8);
title(strcat(WQParam,' vs. Turbidity by Location on Hydrograph'));
legend('Rising Limb','Falling Limb','Baseflow','Location','SouthEast');
xlabel(str2); ylabel(str3);
xlim([0.1 10000]); ylim([0.1 10000]);
ytic=get(gca,'YTick'); % Grab the y tick values
yticlab=num2str(ytic'); % Convert to string mx
set(gca,'YTickLabel',yticlab) % Re-label
xtic=get(gca,'XTick'); % Grab the y tick values
xticlab=num2str(xtic'); % Convert to string mx
set(gca,'XTickLabel',xticlab) % Re-label
```

```
% separate grab samples from storm Isco samples
ind = find(WQSampleType==2); % grab samples
ind2 = find(WQSampleType==1); % storm Isco Samples
grabTurb = turbMatchedTS.Data(ind); grabTSS = WQVar(ind);
stormTurb = turbMatchedTS.Data(ind2); stormTSS = WQVar(ind2);

figure(4); % Plot TSS vs. Turbidity on log-log scale by sample type
loglog(stormTurb,stormTSS,'o','MarkerFaceColor','blue','MarkerEdgeColor','blac
k','MarkerSize',8);
hold on;
loglog(grabTurb,grabTSS,'s','MarkerFaceColor','green','MarkerEdgeColor','black
','MarkerSize',8);
title(strcat(WQParam,' vs. Turbidity by Sample Type'));
legend('Isco Storm Samples','Grab Samples','Location','SouthEast');
xlabel(str2); ylabel(str3);
xlim([0.1 10000]); ylim([0.1 10000]);
ytic=get(gca,'YTick'); % Grab the y tick values
yticlab=num2str(ytic'); % Convert to string mx
set(gca,'YTickLabel',yticlab) % Re-label
xtic=get(gca,'XTick'); % Grab the y tick values
xticlab=num2str(xtic'); % Convert to string mx
set(gca,'XTickLabel',xticlab) % Re-label

% separate data into different storm numbers
ind2 = isfinite(WQStormNum);
ind3 = unique(WQStormNum(ind2));
% if min(WQStormNum(ind2))>1
%     WQStormNum = WQStormNum-(min(WQStormNum(ind2)-1))
% end
storms = zeros(length(ind3),3); % initialize variable
count = 0;
for i = 1:length(ind3) % compute start and stop rows for each storm
    ind = find(WQStormNum==ind3(i));
    storms(i,1) = i; storms(i,2) = min(ind); storms(i,3) = max(ind);
end
stormDates = cellstr(datestr(numDateWQ(storms(:,2)),2)); % Get text strings of
each storm date
figure(5)
cc=hsv(length(ind3));
for i =1:length(ind3);

    loglog(turbMatchedTS.Data(storms(i,2):storms(i,3)),WQVar(storms(i,2):storm
s(i,3)),'o',...
        'MarkerFaceColor',cc(i,:),'MarkerEdgeColor','black','MarkerSize',8,'Di
splayName',stormDates{i});
    hold on
end
title(strcat(WQParam,' vs. Turbidity by Individual Storm'));
legend('Location','SouthEast');
xlabel(str2); ylabel(str3);
xlim([0.1 10000]); ylim([0.1 10000]);
ytic=get(gca,'YTick'); % Grab the y tick values
yticlab=num2str(ytic'); % Convert to string mx
```

```
set(gca,'YTickLabel',yticlab) % Re-label
xtic=get(gca,'XTick'); % Grab the y tick values
xticlab=num2str(xtic'); % Convert to string mx
set(gca,'XTickLabel',xticlab) % Re-label
hold off
disp('program complete');
toc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program to plot storm event analysis
% and generate storm statistics
% Scott Hamshaw
% 07-Nov-2014
% Revised: 23-May-2016
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% NOTES: hydrology site codes: [1 - MAD, 2 - SHP, 3 - HBR, 4 - MIL, 5 - FOL, 6
- FRE]
% met site codes: [ 1 = MIL, 2 - SUG, 3 - SUL, 4 - WDE, 5 - JFR, 6 - FRE, 7 -
MAD, 8 - WEM ]
% If progam is run with plots generated, limiting to maximum of 50 storms
% is advised

clc; close all; clear all;
disp('Program started...');
tic;
addpath('../TimeSeries');
addpath('../Plotting');
load sedRatingCurveParams;
rainData = load('../Data/Mad/MetData/Mad_AllRainData');
load('../Data/Mad/MetData/WEM_SoilMoist_2013-2015_15Min.mat');
MapUnderlay = imread('../Plotting/Images/UnderlayMapr.jpg');

%%%% User set parameters %%%%%%%%%%%%%%%%%%%
SiteCode = 'MIL'; % set site code number
plotYes = 1; % set to 1 for plotting storms or 0 for no plotting
savePlots = 0; % set to 1 to save plots to pdfs
saveData = 0; % set to 1 to save data as .csv
exportPlots = 0; % set to 1 to save individual hysteresis plots

if plotYes==1
    disp('Program set to display storm plots');
else
    disp('Program set to NOT display plots');
end

% load site parameters
rainStaCodes = {'MIL';'SUG';'SUL';'WDE';'JFR';'FRE';'MAD';'WEM'};
rainStaLabels = {'Irasville Weather (MIL)';'Sugarbush Weather (SUG)';...
    'South Hill Weather (SUL)';'East Warren Weather (WDE)';'Fayston Weather
(JFR)';...
    'Warren Weather (FRE)';'Moretown Weather (MAD)';'Waitsfield Weather
(WEM)'};
```

```
hystClassLabels =
{'1A';'1B';'1C';'2A';'2B';'2C';'2D';'2E';'3A';'3B';'3C';'4';'5A';'5B';'Complex
'};
switch SiteCode
    case 'MAD'
        load('../Data/Mad/SensorData/MAD_Sensor_2012-2015_15Min');
        load ('../Data/Mad/EventData/MAD_Online_2012-2015.mat');
        %load ('../Data/Mad/EventData/MAD_Borg_Deploy.mat');
        load('Output Data/MAD_stormHystClass.mat');
        str1 = 'Mad River';
        a = MAD.a; SF=MAD.SF; b=MAD.b;
        a = 2.1433; SF = 1.4933; b=0.7640;
        discharge = discharge/35.31; % convert discharge from cfs to m^3/s
        convFact = 1000;
        site = 1;
        staPriority = [8 1 5 7 6 2 4 3];
        catchArea = 360008350; %square meters
        smoothingParam = [3 21];
        TribMap = imread('../Plotting/Images/MADMap.jpg'); % set path of map
    case 'FOL'
        load('../Data/Mad/SensorData/FOL_Sensor_2013-2015_15Min');
        load ('../Data/Mad/EventData/FOL_Events_2013-2015.mat');
        %load ('../Data/Mad/EventData/FOL_Phillips_Deploy.mat');
        load('Output Data/FOL_stormHystClass.mat');
        str1 = 'Folsom Brook';
        a = FOL.a; SF=FOL.SF; b=FOL.b;
        convFact = 1000;
        site = 5;
        staPriority = [4 1 8 2 6 5 7 3];
        catchArea = 18197700; %square meters
        smoothingParam = [4 11];
        TribMap = imread('../Plotting/Images/FOLMap.jpg'); % set path of map
    case 'FRE'
        load('../Data/Mad/SensorData/FRE_Sensor_2013_15Min');
        load ('../Data/Mad/EventData/FRE_Events_2013.mat');
        load('Output Data/FRE_stormHystClass.mat');
        str1 = 'Freeman Brook';
        a = FRE.a; SF=FRE.SF; b=FRE.b;
        convFact = 1000;
        site = 6;
        staPriority = [4 6 1 8 2 5 7 3];
        p = [-4.8018,83.01,-27];
        discharge = polyval(p,stage*3.2808)/35.31;
        catchArea = 16962300; %square meters
        smoothingParam = [4 11];
        TribMap = imread('../Plotting/Images/FREMap.jpg'); % set path of map
    case 'MIL'
        load('../Data/Mad/SensorData/MIL_Sensor_2012-2015_15Min');
        load ('../Data/Mad/EventData/MIL_Events_2012-2015.mat');
        %load ('../Data/Mad/EventData/MIL_Phillips_Deploy.mat');
        load('Output Data/MIL_stormHystClass.mat');
        str1 = 'Mill Brook';
        a = MIL.a; SF=MIL.SF; b=MIL.b;
        convFact = 1000;
```

```
        site = 4;
        staPriority = [2 1 8 5 6 4 7 3];
        catchArea = 49823000; %square meters
        smoothingParam = [4 11];
        TribMap = imread('../Plotting/Images/MILMap.jpg'); % set path of map
    case 'SHP'
        load('../Data/Mad/SensorData/SHP_Sensor_2013-2015_15Min');
        load ('../Data/Mad/EventData/SHP_Events_2013-2015.mat');
        %load ('../Data/Mad/EventData/SHP_Phillips_Deploy.mat');
        load('Output Data/SHP_stormHystClass.mat');
        str1 = 'Shepard Brook';
        a = SHP.a; SF=SHP.SF; b=SHP.b;
        convFact = 1000;
        site = 2;
        staPriority = [5 8 1 2 3 7 4 6];
        catchArea = 44611100; %square meters
        smoothingParam = [4 11];
        TribMap = imread('../Plotting/Images/SHPMap.jpg'); % set path of map
    case 'HBR'
        load('../Data/Mad/SensorData/HBR_Sensor_2013_15Min');
        load ('../Data/Mad/EventData/HBR_Events_2013.mat');
        load('Output Data/HBR_stormHystClass.mat');
        str1 = 'High Bridge Brook';
        a = HBR.a; SF=HBR.SF; b=HBR.b;
        convFact = 1000;
        site = 3;
        staPriority = [8 1 4 6 5 3 7 2];
        discharge = (1.0376.*27.12.*(stage*3.28).^2.1277)/35.31;
        catchArea = 8643600; %square meters
        smoothingParam = [4 11];
        TribMap = imread('../Plotting/Images/HBRMap.jpg'); % set path of map
end
%%%% Data pre-processing
ind = stormHystClass(:,4) == 99;
stormHystClass(ind,4) = 14;
%convert precip from in to mm
rainData.allRainData = rainData.allRainData .* 25.4;

% calculate number of storm events
numEvents = length(x)/2;
% determine start and stop of windows to extract for storm event analysis
% and convert to datetime arrays
ind = 1:2:numEvents*2;
startDates = datetime(x(ind),'ConvertFrom','datenum');
ind = 2:2:numEvents*2;
endDates = datetime(x(ind),'ConvertFrom','datenum');

% initialize variables to NaNs
[totRain,peakFlow,timeToPeak,totLoad,baseflow,rainDuration,...
    flowDuration,stormFlow,basinLag,lastEventTime,maxRain,...
    peakTSS,TSSTimeToPeak,floodIntensity,rainStaCV,avgRainInt,weightAvgRainAmt
,...
    rainAmt14day, rainAmt10day, rainAmt7day, rainAmt3day, rainAmt1day,...
    VWC1,VWC2,VWC3,VWC4,HI,meanTSS,totFlow,TSSPeakRainDiff...
```

```
    rainRunoffRatio] = deal(NaN(numEvents,1));
% initialize datetime arrays
[peakFlowTime,sedPeakTime,rainStart,rainEnd,sedStart,sedEnd,flowStart,...
    flowEnd,rainCM,flowCM] =
deal(repmat(datetime(2000,1,1,0,0,0),numEvents,1));

% store site number and storm numbers
site = site * ones(numEvents,1);
% set storm numbers (format siteNum.YYMMDDHH
stormNum = site + ( startDates.Month*1000 + startDates.Day*10
+startDates.Hour/10 + (startDates.Year - 2000)*100000)/10000000;

% store start & end times
startTime = startDates; endTime = endDates;
% store month
stormMonth = startDates.Month;

% Loop through storm events
for i = 1:numEvents

    % isolate data for storm event i
    IscoData = dataExtractorNum([startDates(i) endDates(i)],[numDate turbidity
discharge],1);
    numDateCrop = datetime(IscoData(:,1),'ConvertFrom','datenum');
    turbidityCrop= IscoData(:,2); dischargeCrop = IscoData(:,3);
    if i >= 3 && i <= 18
        if strcmp(SiteCode,'HBR')
            TSSData = dataExtractorNum([startDates(i)+1/24
endDates(i)+1/24],[numDate turbidity],1);
            turbidityCrop = TSSData(:,2);
        end
    end

    MetData = dataExtractorNum([startDates(i)
endDates(i)],[rainData.rainNumDate rainData.allRainData],1);
    rainDateCrop = datetime(MetData(:,1),'ConvertFrom','datenum');
    rainAmtCrop = MetData(:,2:end); % MIL SUG SUL WDE JFR FRE MAD WEM

    SoilMoistData = dataExtractorNum([startDates(i) endDates(i)],[rainNumDate
WC1 WC2 WC3 WC4],1);
    soilMoistDateCrop = datetime(SoilMoistData(:,1),'ConvertFrom','datenum');
    soilMoistCrop = SoilMoistData(:,2:end); % WC1 WC2 WC3 WC4

    if ~isempty(soilMoistCrop)
    % average soil moisture for first hour of storm
    VWC1(i) = mean(soilMoistCrop(1:4,1));
    VWC2(i) = mean(soilMoistCrop(1:4,2));
    VWC3(i) = mean(soilMoistCrop(1:4,3));
    VWC4(i) = mean(soilMoistCrop(1:4,4));
    end
    % compute rain totals at all gauges and find reference station to use
    % using pre-determined priority station priority list
    sumRainAmtCrop = sum(rainAmtCrop);
    siteNaN = isnan(sumRainAmtCrop);
```

```
    siteInd = find(siteNaN == 0);
    sitesOnline = ismember(staPriority,siteInd);
    sitesOnline = staPriority(sitesOnline);
    if isempty(sitesOnline)
    refSta = 7;
    else
    refSta = sitesOnline(1);
    end

    % check if no rain recorded at reference station, then use next best
    % rain gauge that recorded precip
    if sumRainAmtCrop(refSta)==0

        [~,refSta] = max(sumRainAmtCrop);
    end


    % calculate CV of rain gauges if more than 2 online
    if length(sitesOnline) >= 3
        rainStaCV(i) =
std(sumRainAmtCrop(sitesOnline))/mean(sumRainAmtCrop(sitesOnline));
    end

    % find rain start/stop dates, and duration
    ind = find(rainAmtCrop(:,refSta),1,'first');
    if isempty(ind)
        [~,indNewSta] = max(sumRainAmtCrop);
        ind = find(rainAmtCrop(:,indNewSta),1,'first');
        if isempty(ind)
            rainStart(i) = rainDateCrop(1);
        else
            rainStart(i) = rainDateCrop(ind);
        end
        ind = find(rainAmtCrop(:,indNewSta),1,'last');
        if isempty(ind)
            rainEnd(i) = rainDateCrop(2);
        else
            rainEnd(i) = rainDateCrop(ind);
        end
    else
        rainStart(i) = rainDateCrop(ind);
        ind = find(rainAmtCrop(:,refSta),1,'last');
        rainEnd(i) = rainDateCrop(ind);
    end

    dt = rainEnd(i)- rainStart(i);
    rainDuration(i) = hours(dt); % in hours


    % calculate total rainfall and rain center of mass
    cumRain = cumsum(rainAmtCrop(:,refSta),'omitnan');
    totRain(i) = sum(rainAmtCrop(:,refSta),'omitnan');
    ind = find(cumRain >= totRain(i)/2,1,'first');
    rainCM(i) = rainDateCrop(ind);
```

```
    % find max hourly rain rate
    rainRate = rainAmtCrop (1:end-3,refSta)+rainAmtCrop(2:end-
2,refSta)+rainAmtCrop(3:end-1,refSta)...
        +rainAmtCrop(4:end,refSta);
    maxRain(i) = max(rainRate);
    % calculate average hourly rainfall intensity
    if rainDuration(i) > 0
    avgRainInt(i) = totRain(i)/rainDuration(i);
    else
        avgRainInt(i) = totRain(i)/0.25;
    end

    %calculate weighted average of rainfall using Thiessen Polygon method
    rainTot = sum(rainAmtCrop(:,sitesOnline));
    rainGaugeWeighting = autoThiessenPoly(sitesOnline);
    weightAvgRainAmt(i) = sum(rainTot.*rainGaugeWeighting);

    %calculate antecedent rainfall
    longMetData = dataExtractorNum([startDates(i)-14
endDates(i)],[rainData.rainNumDate rainData.allRainData],1);
    longRainDateCrop = datetime(longMetData(:,1),'ConvertFrom','datenum');
    longRainAmtCrop = longMetData(:,2:end); % MIL SUG SUL WDE JFR FRE MAD WEM

    longSumRainAmtCrop = sum(longRainAmtCrop);

    siteNaN = isnan(longSumRainAmtCrop);
    siteInd = find(siteNaN == 0);
    longSitesOnline = ismember(staPriority,siteInd);
    longSitesOnline = staPriority(longSitesOnline);

    if isempty(longSitesOnline)
        [~,longSitesOnline] = min(sum(isnan(longRainAmtCrop)));
    end

    rainTot = sum(longRainAmtCrop(:,longSitesOnline),'omitnan');
    rainGaugeWeighting = autoThiessenPoly(longSitesOnline);
    rainAmt14day(i) = sum(rainTot.*rainGaugeWeighting);

    ind = find(longRainDateCrop>(startDates(i)-10));
    longRainAmtCrop2 = longRainAmtCrop(ind,:);
    rainTot = sum(longRainAmtCrop2(:,longSitesOnline),'omitnan');
    rainAmt10day(i) = sum(rainTot.*rainGaugeWeighting);

    ind = find(longRainDateCrop>(startDates(i)-7));
    longRainAmtCrop2 = longRainAmtCrop(ind,:);
    rainTot = sum(longRainAmtCrop2(:,longSitesOnline),'omitnan');
    rainAmt7day(i) = sum(rainTot.*rainGaugeWeighting);

    ind = find(longRainDateCrop>(startDates(i)-3));
    longRainAmtCrop2 = longRainAmtCrop(ind,:);
    rainTot = sum(longRainAmtCrop2(:,longSitesOnline),'omitnan');
    rainAmt3day(i) = sum(rainTot.*rainGaugeWeighting);

    ind = find(longRainDateCrop>(startDates(i)-1));
```

```
    longRainAmtCrop2 = longRainAmtCrop(ind,:);
    rainTot = sum(longRainAmtCrop2(:,longSitesOnline),'omitnan');
    rainAmt1day(i) = sum(rainTot.*rainGaugeWeighting);

    % find flow start of storm flow
    temp = diff(dischargeCrop);
    ind = find(temp > 0,1,'first');
    flowStart(i) = numDateCrop(ind);
    baseflow(i) = dischargeCrop(ind);

    % find end of storm flow
    ind2 = length(numDateCrop);
    flowEnd(i) = numDateCrop(ind2);
    baseflowEnd = dischargeCrop(ind2);
    % extract storm flow (baseflow separation)
    baseFlow = interp1([datenum(numDateCrop(ind))
datenum(numDateCrop(ind2))],...
        [baseflow(i)
baseflowEnd],datenum(numDateCrop(ind)):1/96:datenum(numDateCrop(ind2)));
    temp = (dischargeCrop(ind:ind2)-baseFlow')*(15*60);
    temp2 = cumsum(temp);
    stormFlow(i) = sum(temp);
    ind3 = find(temp2 >= stormFlow(i)/2,1,'first');
    flowCM(i) = numDateCrop(ind3+ind);
    dt = numDateCrop(ind2)-numDateCrop(ind);
    flowDuration(i) = hours(dt);
    totFlow(i) = sum(dischargeCrop(ind:ind2)*(15*60));

    % find peak flow and time to peak flow
    [peakFlow(i),ind4] = max(dischargeCrop);
    peakFlowTime(i) = numDateCrop(ind4);
    dt = peakFlowTime(i)-flowStart(i);
    timeToPeak(i) = hours(dt);
    dt = flowCM(i)-rainCM(i);
    basinLag(i) = hours(dt);
    % calculate flood intensity and rainfall-runoff ratio
    floodIntensity(i) = (peakFlow(i)-baseflow(i))/timeToPeak(i);
    rainRunoffRatio(i) = stormFlow(i)/(catchArea*(totRain(i)/1000));

    % last event time
    if i ==1
        lastEventTime(i) = 0;
    else
        dt = rainStart(i)-rainEnd(i-1);
        if hours(dt) >= 0
            lastEventTime(i) = hours(dt);
        else
            lastEventTime(i)=0;
        end
    end


    TSS =(SF*a*(turbidityCrop).^b); % turbidity based estimate of TSS (mg/L)
    meanTSS(i) = mean(TSS);
```

```
     % find start of sediment flow
    temp = diff(TSS);
    ind = find(temp > 0,1,'first');
    if isempty(ind)
        ind = 1;
    end
    sedStart(i) = numDateCrop(ind);
    sedEnd(i) = flowEnd(i);

    % find tss  and time to peak tss
    [peakTSS(i),ind4] = max(TSS);
    sedPeakTime(i) = numDateCrop(ind4);
    dt = peakFlowTime(i)-sedStart(i);
    TSSTimeToPeak(i) = hours(dt);
    dt = sedPeakTime(i)-rainCM(i);
    TSSPeakRainDiff(i) = hours(dt);
    TSS = TSS./1000;
    Load = TSS(ind:end).*dischargeCrop(ind:end)*convFact*(15*60)*(1/10^6); %
time series of sediment load (kg/15-min)

    totLoad(i) = sum(Load,'omitnan');

  if startDates(i) < flowStart(i)
        ind5 = find(numDateCrop >= flowStart(i));
  else
      ind5 = 1:length(TSS);
  end

    % smooth TSS data for use in hysteresis plots
    if length(TSS(ind5))>=smoothingParam(2)
     tssSmooth = sgolayfilt(TSS(ind5),smoothingParam(1),smoothingParam(2));
    else
        tssSmooth = TSS(ind5);
    end
    % smooth discharge data
    if length(dischargeCrop(ind5))>=smoothingParam(2)
        dischargeSmooth =
sgolayfilt(dischargeCrop(ind5),smoothingParam(1),smoothingParam(2));
    else
        dischargeSmooth = dischargeCrop(ind5);
    end

    % calculate HI
    plotHI = hystInd(dischargeSmooth,tssSmooth);
    HI(i) = mean(plotHI);

    if exportPlots == 1
        % call function to generate hysteresis plots
        [A,B,C,D] =
hysteresisPlots(dischargeSmooth,tssSmooth,stormNum(i),site(i),i,plotHI);
        BW28x28(i,:) = A;
        Grayscale28x28(i,:) = B;
        HI28x28(i,:) = C;
```

```
        RGB28x28(i,:) = D;
    end

if plotYes == 1
    scrsz = get(groot,'ScreenSize');

    h = figure('Position',[50 50 1100 860]);
    %h = figure('Position',[1 scrsz(4)/4 scrsz(3)/2 2*scrsz(4)/3]);
    annoFontSize = 11;
    str1b=strjoin({SiteCode,'\bf',num2str(i)});
    t =
annotation('textbox',[0.05,0.95,.4,.05],'String',str1b,'LineStyle','none');
    t.FontSize =  14;
    str1a=strjoin({'Storm Date: ','\bf',datestr(numDateCrop(1))});
    t =
annotation('textbox',[0.05,0.89,.4,.05],'String',str1a,'LineStyle','none');
    t.FontSize =  annoFontSize;
    str2=strjoin({'Storm Number: ','\bf',num2str(stormNum(i),9)});
    t =
annotation('textbox',[0.05,0.92,.4,.05],'String',str2,'LineStyle','none');
    t.FontSize =  annoFontSize;
    str3a=strjoin({'Rainfall Station: ','\bf',rainStaLabels{refSta}});
    t =
annotation('textbox',[0.05,0.86,.4,.05],'String',str3a,'LineStyle','none');
    t.FontSize =  annoFontSize;
    str3=strjoin({'Total Rainfall (mm): ','\bf',sprintf('
%.2f{}',totRain(i))});
    t =
annotation('textbox',[0.05,0.83,.4,.05],'String',str3,'LineStyle','none');
    t.FontSize =  annoFontSize;
    str4=strjoin({'Time Since Last Event (hr):
','\bf',num2str(lastEventTime(i))});
    t =
annotation('textbox',[0.05,0.80,.4,.05],'String',str4,'LineStyle','none');
    t.FontSize =  annoFontSize;
    str5 = strjoin({'Rainfall-Runoff Ratio: ','\bf',sprintf('
%.2f{}',rainRunoffRatio(i)*100),'%'});
    t =
annotation('textbox',[0.05,0.77,.4,.05],'String',str5,'LineStyle','none');
    t.FontSize =  annoFontSize;
    str6 = strjoin({'Turbidity Based Load (kg): ','\bf',sprintf('
%.0f{}',totLoad(i))});
    t =
annotation('textbox',[0.05,0.74,.4,.05],'String',str6,'LineStyle','none');
    t.FontSize =  annoFontSize;
    str7 = strjoin({'Flood Intensity: ','\bf',sprintf('
%.3f{}',floodIntensity(i))});
    t =
annotation('textbox',[0.05,0.71,.4,.05],'String',str7,'LineStyle','none');
    t.FontSize =  annoFontSize;
    str8 = strjoin({'Antecedent Soil Moisture (%): ','\bf',sprintf('
%.2f{}',VWC2(i)),'%'});
    t =
annotation('textbox',[0.05,0.68,.4,.05],'String',str8,'LineStyle','none');
```

```
    t.FontSize =  annoFontSize;
    str9 = strjoin({'Antecedent 7-Day Rainfall (mm): ','\bf',sprintf('
%.2f{}',rainAmt7day(i))});
    t =
annotation('textbox',[0.05,0.65,.4,.05],'String',str9,'LineStyle','none');
    t.FontSize =  annoFontSize;
    str10 = strjoin({'Class ','\bf',hystClassLabels{stormHystClass(i,4)}});
    t =
annotation('textbox',[0.51,0.47,.1,.05],'String',str10,'LineStyle','none');
    t.FontSize =  14;

    t =
annotation('textbox',[0.39,0.95,.2,.05],'String',str1,'LineStyle','none');
    t.FontSize =  12;

    set(gca,'Visible','off');
    posVect4 = [0.32,0.66,0.28,0.33];
    subplot('Position',posVect4)
    image(MapUnderlay);
    axis image;
    axis off;
    posVect3 = [0.32,0.66,0.28,0.33];
    subplot('Position',posVect3)
    image(TribMap);
    axis image;
    axis off;

    posVect1 = [0.05,0.05,0.4,0.4];
    subplot('Position',posVect1)
    yyaxis left
    plot(numDateCrop,dischargeCrop,'LineStyle','-','LineWidth',2);
    ylabel('Discharge (m^{3}/s)');
    text(datenum(peakFlowTime(i)),peakFlow(i),num2str(peakFlow(i),'%
10.1f'),'HorizontalAlignment','left')
    yyaxis right
    plot(numDateCrop,TSS,'LineStyle','--','LineWidth',2);
    ylabel('TSS (mg/L)');
    text(datenum(sedPeakTime(i)),peakTSS(i),num2str(peakTSS(i),'%
10.1f'),'HorizontalAlignment','right')

    posVect5 = [0.05,0.49,0.4,0.15];
    subplot('Position',posVect5);
    bar(datenum(rainDateCrop),rainAmtCrop(:,refSta),'EdgeColor','none','FaceCo
lor',[.27 .51 .70]);
    ylabel('15-min Rain (in)');
    set(gca,'Ydir','reverse','xtick',[]);
    grid off;

    t = annotation('textbox',[0.68,0.95,.3,.05],'String','Watershed Rainfall
Summary','LineStyle','none');
    t.FontSize =  12;
    posVect6 = [0.65,0.82,0.30,0.15];
    subplot('Position',posVect6);
```

```
    bar(sumRainAmtCrop(sitesOnline),'EdgeColor','none','FaceColor',[.27 .51
.70]);
    ylabel('Total Rainfall (mm)');
    set(gca,'XTickLabel',rainStaCodes(sitesOnline))
    grid off;

    posVect2 = [0.55,0.05,0.4,0.4];
    subplot('Position',posVect2);
    plot_dir(dischargeSmooth,tssSmooth);
    ylabel('TSS (mg/L)');
    xlabel('Discharge (m^{3}/s)');
    grid off;

    subplot('Position',[0.65,0.52,0.30,0.25]);
    t = linspace(0,1,length(plotHI));
    area(t,plotHI,'FaceColor',[.27 .51 .70],'EdgeColor','none');
    ylabel('H-index');
    xlabel('Normalized discharge (m^{3}/s)');
    ylim([-1 1]);

    if savePlots == 1
        strsave = strcat('Output Data/StormDashboard_',sprintf('
%.8f',stormNum(i)),'.pdf');
        h.PaperOrientation = 'landscape';
        print(h,'-dpdf',strsave,'-bestfit');
    end

end % end if


if saveData == 1
    dataTSOut(i,1) = {numDateCrop};
    dataTSOut(i,2) = {dischargeCrop};
    dataTSOut(i,3) = {TSS};
    dataTSOut(i,4) = {rainAmtCrop(:,refSta)};
end
if plotYes == 1
    %close(h);
end
disp(strcat('Storm_', num2str(i),'_of_', num2str(numEvents), '_complete'))
end % storm event loop

if exportPlots == 1
    save(strcat('Output Data/',SiteCode,'_BW28x28_Output.mat'),'BW28x28');
    save(strcat('Output
Data/',SiteCode,'_Grayscale28x28_Output.mat'),'Grayscale28x28');
    save(strcat('Output Data/',SiteCode,'_HI28x28_Output.mat'),'HI28x28');
    save(strcat('Output Data/',SiteCode,'_RGB28x28_Output.mat'),'RGB28x28');
end

disp('Storm Event Analysis Complete');
if saveData == 1
dataOut =
table(startTime,endTime,rainStart,rainCM,rainEnd,site,stormNum,flowStart,...
```

```
    peakFlowTime,flowCM,flowEnd,sedStart,sedPeakTime,sedEnd,stormMonth,...
    lastEventTime,basinLag,totRain,maxRain,avgRainInt,weightAvgRainAmt,rainDur
ation,rainStaCV,totFlow,stormFlow,peakFlow,...
    rainAmt14day, rainAmt10day, rainAmt7day, rainAmt3day,
rainAmt1day,VWC1,VWC2,VWC3,VWC4,...
    flowDuration,timeToPeak,baseflow,floodIntensity,rainRunoffRatio,peakTSS,me
anTSS,TSSTimeToPeak,TSSPeakRainDiff,totLoad,HI);
writetable(dataOut,'Output Data/OutputStormDataTable.csv');
save('Output Data/OutputStormTSData','dataTSOut');
disp('Data saved');
end
toc

function [A,B,C,D] = hysteresisPlots (Q,C,stormNum,site,stormInd,HI)
% normalize data then interpolate along line to ensure consistent spacing
% of points
Qnorm = (Q - min(Q))./(max(Q)-min(Q));
Cnorm = (C - min(C))./(max(C)-min(C));
Qnorm_dense = Qnorm(1);
Cnorm_dense = Cnorm(1);
for i = 1:length(Qnorm)-1
    dist = Qnorm(i+1) - Qnorm(i);
    if dist > 0.001
        dq = Qnorm(i):0.001:Qnorm(i+1);
        Cq = interp1(Qnorm(i:(i+1)),Cnorm(i:(i+1)),dq);
       Qnorm_dense = horzcat(Qnorm_dense,dq(2:end));
       Cnorm_dense = horzcat(Cnorm_dense,Cq(2:end));
    elseif dist < -0.001
        dq = Qnorm(i):-0.001:Qnorm(i+1);
        Cq = interp1(Qnorm(i:(i+1)),Cnorm(i:(i+1)),dq);
       Qnorm_dense = horzcat(Qnorm_dense,dq(2:end));
       Cnorm_dense = horzcat(Cnorm_dense,Cq(2:end));
    else
        Qnorm_dense = horzcat(Qnorm_dense,Qnorm(i+1));
        Cnorm_dense = horzcat(Cnorm_dense,Cnorm(i+1));
    end

end
Qnorm_dense = horzcat(Qnorm_dense,Qnorm(1));
Cnorm_dense = horzcat(Cnorm_dense,Cnorm(1));

        % save grayscale 28 x 28 pixel and 72 x 72 pixel .bmp images
        h_fig = figure(1);

        plot(Qnorm_dense,Cnorm_dense,'-w','linewidth',35);
         h_fig.Color = 'k';
         h_fig.InvertHardcopy = 'off';
         set(gca,'Visible','off');
        strsave = strcat('Output Plots/BW28x28_',sprintf('
%.8f',stormNum),'.bmp');
        strsave2 = strcat('Output Plots/BW72x72_',sprintf('
%.8f',stormNum),'.bmp');
        F = getframe(gcf);
        print(h_fig,'-opengl','-dbmp256',strsave2,'-r72');
```

```
        I = frame2im(F);
        newI = imresize(I,[28 28]);
        imwrite(newI,strsave);

        A = reshape(newI(:,:,1),1,numel(newI(:,:,1)));

        clf(h_fig);
        t = length(Qnorm_dense);
        for i = 1:length(Qnorm_dense)-1
            tDist(i) = sqrt((Qnorm_dense(i)-Qnorm_dense(i+1))^2 +
(Cnorm_dense(i)-Cnorm_dense(i+1))^2);
        end
        sumDist = sum(tDist);
        cumDist = cumsum(tDist);
        ind = find(cumDist > 0.10*sumDist,1);
        ind1 = find(cumDist > 0.20*sumDist,1);
        ind2 = find(cumDist > 0.30*sumDist,1);
        ind3 = find(cumDist > 0.40*sumDist,1);
        ind4 = find(cumDist > 0.50*sumDist,1);
        ind5 = find(cumDist > 0.60*sumDist,1);
        ind6 = find(cumDist > 0.70*sumDist,1);
        ind7 = find(cumDist > 0.80*sumDist,1);
        ind8 = find(cumDist > 0.90*sumDist,1);
%         plot(Qnorm_dense(1:ind),Cnorm_dense(1:ind),'-w','linewidth',35);
%         hold on
%         plot(Qnorm_dense(ind:ind1),Cnorm_dense(ind:ind1),'-','color',[0.9
0.9 0.9],'linewidth',35);
%         plot(Qnorm_dense(ind1:ind2),Cnorm_dense(ind1:ind2),'-','color',[0.8
0.8 0.8],'linewidth',35);
%         plot(Qnorm_dense(ind2:ind3),Cnorm_dense(ind2:ind3),'-','color',[0.7
0.7 0.7],'linewidth',35);
%         plot(Qnorm_dense(ind3:ind4),Cnorm_dense(ind3:ind4),'-','color',[0.6
0.6 0.6],'linewidth',35);
%         plot(Qnorm_dense(ind4:ind5),Cnorm_dense(ind4:ind5),'-','color',[0.5
0.5 0.5],'linewidth',35);
%         plot(Qnorm_dense(ind5:ind6),Cnorm_dense(ind5:ind6),'-','color',[0.4
0.4 0.4],'linewidth',35);
%         plot(Qnorm_dense(ind6:ind7),Cnorm_dense(ind6:ind7),'-','color',[0.3
0.3 0.3],'linewidth',35);
%         plot(Qnorm_dense(ind7:ind8),Cnorm_dense(ind7:ind8),'-','color',[0.2
0.2 0.2],'linewidth',35);
%         plot(Qnorm_dense(ind8:end),Cnorm_dense(ind8:end),'-','color',[0.1
0.1 0.1],'linewidth',35);

        plot_dir2(Qnorm_dense,Cnorm_dense);
        set(gca,'Visible','off');
        strsave = strcat('Output Plots/Grayscale28x28_',sprintf('
%.8f',stormNum),'.bmp');
        strsave2 = strcat('Output Plots/Grayscale72x72_',sprintf('
%.8f',stormNum),'.bmp');
        F = getframe(gcf);
        print(h_fig,'-opengl','-dbmp256',strsave2,'-r72');
        I = frame2im(F);
        newI = imresize(I,[28 28]);
```

```
        imwrite(newI,strsave);

        B = reshape(newI(:,:,1),1,numel(newI(:,:,1)));


         clf(h_fig);
        t = linspace(0,1,length(HI));
        area(t,HI,'FaceColor','w','EdgeColor','none');
        ylim([-1 1]);
        set(gca,'Visible','off');
        strsave = strcat('Output Plots/HI28x28_',sprintf('
%.8f',stormNum),'.bmp');
        strsave2 = strcat('Output Plots/HI72x72_',sprintf('
%.8f',stormNum),'.bmp');
        F = getframe(gcf);
        print(h_fig,'-opengl','-dbmp256',strsave2,'-r72');
        I = frame2im(F);
        newI = imresize(I,[28 28]);
        imwrite(newI,strsave);

        C = reshape(newI(:,:,1),1,numel(newI(:,:,1)));

        clf(h_fig);

        map = colormap(hot(257));
        h_fig.Color = 'w';
        t = length(Qnorm_dense);
        for i = 1:length(Qnorm_dense)-1
            tDist(i) = sqrt((Qnorm_dense(i)-Qnorm_dense(i+1))^2 +
(Cnorm_dense(i)-Cnorm_dense(i+1))^2);
        end
        sumDist = sum(tDist);
        cumDist = cumsum(tDist);
        ind = find(cumDist > 0.33*sumDist,1);
        ind3 = find(cumDist > 0.66*sumDist,1);
        c = @cmu.colors;

        plot(Qnorm_dense(1:ind),Cnorm_dense(1:ind),'-r','linewidth',35);
        hold on
        plot(Qnorm_dense(ind:ind3),Cnorm_dense(ind:ind3),'-g','linewidth',35);
        plot(Qnorm_dense(ind3:end),Cnorm_dense(ind3:end),'-b','linewidth',35);
        hold off
        set(gca,'Visible','off');
        strsave = strcat('Output Plots/RGB28x28_',sprintf('
%.8f',stormNum),'.bmp');
        strsave2 = strcat('Output Plots/RGB72x72_',sprintf('
%.8f',stormNum),'.bmp');
        F = getframe(gcf);
        print(h_fig,'-opengl','-dbmp256',strsave2,'-r72');
        I = frame2im(F);
        newI = imresize(I,[28 28]);
        imwrite(newI,strsave);
        D = reshape(newI,1,numel(newI));
        clf(h_fig);
```

```
end
```

```
%%%%%%%%%%
% Driver to create training data set and train and test an RBM classifier on
hysteresis plots

% set parameters
numTrials = 5; % number of trials to train and fine-tune classifier
numSets = 5; % number of times to generate new training/test data set
batchsize = 14;
numHidNodes=30;
nnType = 'RBM'; % set to RBM or DBN
dataSet = 'select+rand'; % set to one of 'select+rand',
'synth+rand','cleanonly','tribsonly','all'

addpath('Output Data');
addpath('../ANNs');

% pre-initialize variables
trainError = NaN(1,numTrials*numSets);
testError = NaN(1,numTrials*numSets);
trainDataSummary = NaN(numSets,15);
testDataSummary = NaN(numSets,15);
testConfMatrix = NaN(16,16,numSets*numTrials);
trainConfMatrix = NaN(16,16,numSets*numTrials);
trainClassAccuracy = NaN(numSets*numTrials,15);
testClassAccuracy = NaN(numSets*numTrials,15);
meanTest1OffAccuracy = NaN(1,numSets*numTrials);
meanTrain1OffAccuracy = NaN(1,numSets*numTrials);
% loop through numSets
count = 1;
for idxSet = 1:numSets

    switch dataSet
        case 'all'
            [batchdata,batchtargets,batchstormnums,testbatchdata,testbatchtarg
ets,testbatchstormnums]...
                =
RBMpreprocessGrayscale(0.66,batchsize,15,0,0,'Gray_All_25_50_50');

        case 'cleanonly'
        case 'tribsonly'
        case 'synth+rand'
            [batchdata,batchtargets,batchstormnums,testbatchdata,testbatchtarg
ets,testbatchstormnums]...
                =
RBMpreprocessSynthGrayscale(0.50,batchsize,15,0,0,'Gray_SynthRand_50_50_50');
        case 'select+rand'
            [batchdata,batchtargets,batchstormnums,testbatchdata,testbatchtarg
ets,testbatchstormnums]...
                =
RBMpreprocessSelectGrayscaleBalanced(0.50,batchsize,15,1,0,'Gray_SelRand_25_50
_50');
```

```
         case 'select+randColor'
            [batchdata,batchtargets,batchstormnums,testbatchdata,testbatchtarg
ets,testbatchstormnums]...
                =
RBMpreprocessSelectColorBalanced(0.50,batchsize,15,1,0,'Color_SelRand_25_50_50
');
    end
    trainDataSummary(idxSet,:) = sum(sum(batchtargets,3));
    testDataSummary(idxSet,:) = sum(sum(testbatchtargets,3));
    fprintf(1,'Using training data set realization number %d...\n', idxSet);
    % loop through numTrials
    for idxTrial = 1:numTrials
        % create and pre-train DBNN/RBM
        fprintf(1,'Pre-training network on GPU, trial number %d...\n',
idxTrial);
        DN = deeptrain_GPU(batchdata,[numHidNodes 25 50],batchsize);

        % fine tune classifider
        switch nnType
            case 'RBM'
                fprintf(1,'fine tuning RBM classifier, trial number %d...\n',
idxTrial);
                [w1,wClass,test_err,test_crerr,train_err,train_crerr,confusion
MatrixTrain,...
                    confusionMatrixTest,mean1OffTrainError,mean1OffTestError,m
eanTestError,meanTrainError,...
                    testPredProbs, trainPredProbs] ...
                    =
finetune_RBM_classifier(DN,batchdata,testbatchdata,batchtargets,testbatchtarge
ts);
            case 'DBN'

        end
        testError(count) = meanTestError;
        trainError(count) = meanTrainError;
        testConfMatrix(:,:,count) = confusionMatrixTest;
        trainConfMatrix(:,:,count) = confusionMatrixTrain;
        trainClassAccuracy(count,:) = confusionMatrixTrain(16,1:15);
        testClassAccuracy(count,:) = confusionMatrixTest(16,1:15);
        meanTest1OffAccuracy(count) = mean1OffTestError;
        meanTrain1OffAccuracy(count) = mean1OffTrainError;
        count = count+1;
    end % end numTrials loop

end % end numSets loop
% calculate average training and testing accuracy
meanTrainAccuracy = 1 - mean(trainError)
meanTestAccuracy = 1- mean(testError)
meanTrainClassAccuracy = mean(trainClassAccuracy,'omitnan')
meanTestClassAccuracy = mean(trainClassAccuracy,'omitnan')


mean1OffTrainAccuracy = 1-mean(mean1OffTrainError)
mean1OffTestAccuracy = 1-mean(mean1OffTestError)
```

```
figure(1);
plot_L1(DN,numHidNodes);

DN.L{1}.vishid = w1(1:end-1,:);
figure(2);
plot_L1(DN,numHidNodes);

disp('program completed');



function
[batchdata,batchtargets,batchstormnums,testbatchdata,testbatchtargets,testbatc
hstormnums]...
    =
RBMpreprocessGrayscale(testDataPortion,batchsize,K,onlyCleanData,onlyTribs,fna
me)

addpath('Output Data');

load trainSelectStorms

% prepare training data
digitdata = [];
targets   = [];
cleanPattern = [];
stormNum = [];

for n = 1:size(trainStormNumbers,1)
    [X,hystClass] = subsetHystPlots28Gray(trainStormNumbers(n,:));
    randomorder = randperm(14);
    digitdata = [digitdata; X(randomorder,:)];
    targetVect = zeros(size(trainStormNumbers,2),K);
    t = 1:14;
    ind = sub2ind(size(targetVect),t,hystClass);
    targetVect(ind) = 1;
    targets = [targets; targetVect(randomorder,:)];
    stormNums = trainStormNumbers(n,:)';
    stormNum = [stormNum; stormNums(randomorder)];
end

digitdata = digitdata/255;

traindigitdata = digitdata(1:28,:);
traintargets = targets(1:28,:);
trainstormNum = stormNum(1:28,:);


% prepare testing data

digitdata = [];
targets   = [];
```

```
cleanPattern = [];
stormNum = [];
K = 15;
    % Folsom Brook Data
    load FOL_Grayscale28x28_Output; load FOL_stormHystClass;
    targetClass = stormHystClass(:,4);
    targetVect = zeros(size(Grayscale28x28,1),K);
    ind = sub2ind(size(targetVect), (1:size(Grayscale28x28,1))', targetClass);
    targetVect(ind)=1;
    cleanPattern = [cleanPattern; stormHystClass(:,5)];
    stormNum = [stormNum; stormHystClass(:,1)];
    digitdata = [digitdata; Grayscale28x28];
    targets = [targets; targetVect];
    % Shepard Brook Data
    load SHP_Grayscale28x28_Output; load SHP_stormHystClass;
    targetClass = stormHystClass(:,4);
    targetVect = zeros(size(Grayscale28x28,1),K);
    ind = sub2ind(size(targetVect), (1:size(Grayscale28x28,1))', targetClass);
    targetVect(ind)=1;
    cleanPattern = [cleanPattern; stormHystClass(:,5)];
    stormNum = [stormNum; stormHystClass(:,1)];
    digitdata = [digitdata; Grayscale28x28];
    targets = [targets; targetVect];
    % Mill Brook Data
    load MIL_Grayscale28x28_Output; load MIL_stormHystClass;
    targetClass = stormHystClass(:,4);
    targetVect = zeros(size(Grayscale28x28,1),K);
    ind = sub2ind(size(targetVect), (1:size(Grayscale28x28,1))', targetClass);
    targetVect(ind)=1;
    cleanPattern = [cleanPattern; stormHystClass(:,5)];
    stormNum = [stormNum; stormHystClass(:,1)];
    digitdata = [digitdata; Grayscale28x28];
    targets = [targets; targetVect];
    % Freeman Brook Data
    load FRE_Grayscale28x28_Output; load FRE_stormHystClass;
    targetClass = stormHystClass(:,4);
    targetVect = zeros(size(Grayscale28x28,1),K);
    ind = sub2ind(size(targetVect), (1:size(Grayscale28x28,1))', targetClass);
    targetVect(ind)=1;
    cleanPattern = [cleanPattern; stormHystClass(:,5)];
    stormNum = [stormNum; stormHystClass(:,1)];
    digitdata = [digitdata; Grayscale28x28];
    targets = [targets; targetVect];
    % Mad River Data
    if onlyTribs == 0
        load MAD_Grayscale28x28_Output; load MAD_stormHystClass;
        targetClass = stormHystClass(:,4);
        targetVect = zeros(size(Grayscale28x28,1),K);
        ind = sub2ind(size(targetVect), (1:size(Grayscale28x28,1))',
targetClass);
        targetVect(ind)=1;
        cleanPattern = [cleanPattern; stormHystClass(:,5)];
        stormNum = [stormNum; stormHystClass(:,1)];
        digitdata = [digitdata; Grayscale28x28];
```

```
        targets = [targets; targetVect];
    end
    % High Bridge Brook Data
    load HBR_Grayscale28x28_Output; load HBR_stormHystClass;
    targetClass = stormHystClass(:,4);
    targetVect = zeros(size(Grayscale28x28,1),K);
    ind = sub2ind(size(targetVect), (1:size(Grayscale28x28,1))', targetClass);
    targetVect(ind)=1;
    cleanPattern = [cleanPattern; stormHystClass(:,5)];
    stormNum = [stormNum; stormHystClass(:,1)];
    digitdata = [digitdata; Grayscale28x28];
    targets = [targets; targetVect];


digitdata = double(digitdata);
if onlyCleanData == 1
    ind = find(cleanPattern == 0);
    count = 1;
    for i = 1:length(ind)
        digitdata(ind(count),:)=NaN;
        targets(ind(count),:)=NaN;
        stormNum(ind(count))=NaN;
        count = count+1;
    end
    digitdata(any(isnan(digitdata),2),:)=[];
    targets(any(isnan(targets),2),:)=[];
    stormNum(any(isnan(stormNum),2),:)=[];

end

 ind = find(targets(:,15) == 1);
    count = 1;
    for i = 1:length(ind)
        digitdata(ind(count),:)=NaN;
        targets(ind(count),:)=NaN;
        stormNum(ind(count))=NaN;
        count = count+1;
    end
    digitdata(any(isnan(digitdata),2),:)=[];
    targets(any(isnan(targets),2),:)=[];
    stormNum(any(isnan(stormNum),2),:)=[];

[Lia,Locb] = ismember(stormNum,trainstormNum);
ind = find(Locb ~= 0);



count = 1;
    for i = 1:length(ind)
        digitdata(ind(count),:)=NaN;
        targets(ind(count),:)=NaN;
        stormNum(ind(count))=NaN;
        count = count+1;
    end
```

```
    digitdata(any(isnan(digitdata),2),:)=[];
    targets(any(isnan(targets),2),:)=[];
    stormNum(any(isnan(stormNum),2),:)=[];

digitdata = digitdata/255;

% Randomize order
testNum = size(digitdata,1);
randomorder = randperm(testNum);
tempDigitData = digitdata(randomorder,:);
tempTargetData = targets(randomorder,:);
tempStormNum = stormNum(randomorder,:);

numRandTrain = 250;

traindigitdata = [traindigitdata; tempDigitData(1:numRandTrain,:)];
    traintargets = [traintargets; tempTargetData(1:numRandTrain,:)];
    trainstormNum = [trainstormNum; tempStormNum(1:numRandTrain,:)];

    testDigitData = tempDigitData(numRandTrain+1:end,:);
    testTargetData = tempTargetData(numRandTrain+1:end,:);
    testStormNum = tempStormNum(numRandTrain+1:end,:);

    trainNum = numel(trainstormNum);
    testNum = size(testDigitData, 1);
    fprintf(1, 'Size of the training dataset = %5d \n', trainNum);
fprintf(1, 'Size of the testing dataset = %5d \n', testNum);
% batch training data
numbatches   = floor(trainNum/batchsize);
numdims      = size(traindigitdata, 2);
batchdata    = zeros(batchsize, numdims, numbatches);
batchtargets = zeros(batchsize, K, numbatches);
batchstormnums = zeros(batchsize,1,numbatches);
for b=1:numbatches
    batchdata(:,:,b)  = traindigitdata(1+(b-1)*batchsize:b*batchsize, :);
    batchtargets(:,:,b) = traintargets(1+(b-1)*batchsize:b*batchsize, :);
    batchstormnums(:,:,b) = trainstormNum(1+(b-1)*batchsize:b*batchsize);
end


% batch testing data
randomorder = randperm(testNum);
numbatches   = floor(testNum/batchsize);
numdims      = size(testDigitData, 2);
testbatchdata    = zeros(batchsize, numdims, numbatches);
testbatchtargets = zeros(batchsize, K, numbatches);
testbatchstormnums = zeros(batchsize,1,numbatches);
for b=1:numbatches
    testbatchdata(:,:,b)  = testDigitData(randomorder(1+(b-
1)*batchsize:b*batchsize), :);
    testbatchtargets(:,:,b) = testTargetData(randomorder(1+(b-
1)*batchsize:b*batchsize), :);
    testbatchstormnums(:,:,b) = testStormNum(randomorder(1+(b-
1)*batchsize:b*batchsize), :);
```

```
end
% save dataset as a 3D matrix
temp = datevec(now);
fname = ['../Data/Mad/Hyst/' fname num2str(temp(1)) num2str(temp(2))
num2str(temp(3)) '_' num2str(temp(4)) num2str(temp(5)) num2str(temp(6),2)];
save(fname,
'batchdata','batchtargets','batchstormnums','testbatchdata','testbatchtargets'
,'testbatchstormnums');

end


function [DN] = deeptrain_GPU(batchdata,layers,batchsize)
% Version 1.000
%
% Code provided by Ruslan Salakhutdinov and Geoff Hinton
Modified by Scott Hamshaw
%
% Permission is granted for anyone to copy, use, modify, or distribute this
% program and accompanying programs and documents for any purpose, provided
% this copyright notice is retained and prominently displayed, along with
% a note saying that the original programs are available from our
% web page.
% The programs and documents are distributed without any warranty, express or
% implied.  As the programs were written for research purposes only, they have
% not been tested to the degree that would be advisable in any important
% application.  All use of these programs is entirely at the user's own risk.

% Version 1.100
%
% Updated by Computational Cognitive Neuroscience Lab
% University of Padova
% ccnl.psy.unipd.it
%
% Implementation on graphic processors (GPUs) using MATLAB Parallel Computing
Toolbox

% Program modified by Scott Hamshaw

tic
% DEEP NETWORK SETUP
% (parameters and final network weights will be saved in structure DN)
DN.layersize   = layers;            % network architecture
DN.nlayers     = length(DN.layersize);
DN.maxepochs   = 200;                    % unsupervised learning epochs
DN.batchsize   = batchsize;                   % mini-batch size
sparsity       = 1;                     % set to 1 to encourage sparsity on
third layer
spars_factor   = 0.05;                  % how much sparsity?
epsilonw_GPU   = gpuArray(0.1);         % learning rate (weights)
epsilonvb_GPU  = gpuArray(0.1);         % learning rate (visible biases)
epsilonhb_GPU  = gpuArray(0.1);         % learning rate (hidden biases)
weightcost_GPU = gpuArray(0.0002);      % decay factor
init_momentum  = 0.5;                   % initial momentum coefficient
```

```
final_momentum = 0.9;                   % momentum coefficient

% load training dataset
% fname = ['Hyst_Grayscale_select_b14.mat'];
% load(fname);
fprintf(1,'\nUnsupervised training of a deep belief net\n');
DN.err = zeros(DN.maxepochs, DN.nlayers, 'single');
tic();

for layer = 1:DN.nlayers

    % for the first layer, input data are raw images
    % for next layers, input data are preceding hidden activations
    fprintf(1,'Training layer %d...\n', layer);
    if layer == 1
        data_GPU = gpuArray(single(batchdata));
    else
        data_GPU  = batchposhidprobs;
    end

    % initialize weights and biases
    numhid  = DN.layersize(layer);
    [numcases, numdims, numbatches] = size(data_GPU);
    numcases_GPU     = gpuArray(numcases);
    vishid_GPU       = gpuArray(0.1*randn(numdims, numhid, 'single'));
    hidbiases_GPU    = gpuArray(zeros(1,numhid, 'single'));
    visbiases_GPU    = gpuArray(zeros(1,numdims, 'single'));
    vishidinc_GPU    = gpuArray(zeros(numdims, numhid, 'single'));
    hidbiasinc_GPU   = gpuArray(zeros(1,numhid, 'single'));
    visbiasinc_GPU   = gpuArray(zeros(1,numdims, 'single'));
    batchposhidprobs = gpuArray(zeros(DN.batchsize, numhid, numbatches,
'single'));

    for epoch = 1:DN.maxepochs
        errsum = 0;
        for mb = 1:numbatches
            data_mb = data_GPU(:, :, mb);  % select one slice (mini-batch)
            rbm_GPU;  % learn an RBM with 1-step contrastive divergence
            errsum = errsum + err;
            if epoch == DN.maxepochs
                batchposhidprobs(:, :, mb) = poshidprobs_GPU;
            end
            if sparsity && (layer == 3)
                poshidact = sum(poshidprobs_GPU);
                Q = poshidact/DN.batchsize;
                if mean(Q) > spars_factor
                    hidbiases_GPU = hidbiases_GPU - epsilonhb_GPU*(Q-
spars_factor);
                end
            end
        end
        DN.err(epoch, layer) = errsum;
    end
    % save learned weights
```

```
    DN.L{layer}.hidbiases  = gather(hidbiases_GPU);
    DN.L{layer}.vishid     = gather(vishid_GPU);
    DN.L{layer}.visbiases  = gather(visbiases_GPU);
    DN.L{layer}.batchposhidprobs = gather(batchposhidprobs);

end

DN.learningtime = toc();
fprintf(1, '\nElapsed time: %d \n', DN.learningtime);
fname = 'DBN_Grayscale_select_14.mat';
% save final network and parameters
save (fname, 'DN');
hidrecbiases = DN.L{1,1}.hidbiases;
visbiases = DN.L{1,1}.visbiases;
vishid = DN.L{1,1}.vishid;
save mnistvhclassify vishid hidrecbiases visbiases;
penrecbiases = DN.L{1,2}.hidbiases;
hidgenbiases = DN.L{1,2}.visbiases;
hidpen = DN.L{1,2}.vishid;
save mnisthpclassify hidpen penrecbiases hidgenbiases;
penrecbiases2 = DN.L{1,3}.hidbiases;
hidgenbiases2 = DN.L{1,3}.visbiases;
hidpen2 = DN.L{1,3}.vishid;
save mnisthp2classify hidpen2 penrecbiases2 hidgenbiases2;
toc
end

% Version 1.000
%
% Code provided by Geoff Hinton and Ruslan Salakhutdinov
% Modified by Scott Hamshaw
%
% Permission is granted for anyone to copy, use, modify, or distribute this
% program and accompanying programs and documents for any purpose, provided
% this copyright notice is retained and prominently displayed, along with
% a note saying that the original programs are available from our
% web page.
% The programs and documents are distributed without any warranty, express or
% implied.  As the programs were written for research purposes only, they have
% not been tested to the degree that would be advisable in any important
% application.  All use of these programs is entirely at the user's own risk.

% This program trains Restricted Boltzmann Machine in which
% visible, binary, stochastic pixels are connected to
% hidden, binary, stochastic feature detectors using symmetrically
% weighted connections. Learning is done with 1-step Contrastive
Divergence.
% The program assumes that the following variables are set externally:
% maxepoch  -- maximum number of epochs
% numhid    -- number of hidden units
% batchdata -- the data that is divided into batches (numcases numdims
numbatches)

% Version 1.100
```

```
%
% Updated by Computational Cognitive Neuroscience Lab
% University of Padova
% ccnl.psy.unipd.it
%
% Implementation on graphic processors (GPUs) using MATLAB Parallel Computing
Toolbox

momentum_GPU    = gpuArray(init_momentum);

%%%%%%%%% START POSITIVE PHASE %%%%%%%%%
poshidprobs_GPU = 1./(1 + exp(-data_mb * vishid_GPU - repmat(hidbiases_GPU,
numcases, 1)));
posprods_GPU    = data_mb' * poshidprobs_GPU;
poshidact_GPU   = sum(poshidprobs_GPU);
posvisact_GPU   = sum(data_mb);
%%%%%%%%% END OF POSITIVE PHASE  %%%%%%%%%
poshidstates_GPU = poshidprobs_GPU > rand(numcases, numhid);

%%%%%%%%% START NEGATIVE PHASE  %%%%%%%%%
negdata_GPU     = 1./(1 + exp(-poshidstates_GPU * vishid_GPU' -
repmat(visbiases_GPU, numcases, 1)));
neghidprobs_GPU = 1./(1 + exp(-negdata_GPU * vishid_GPU       -
repmat(hidbiases_GPU, numcases, 1)));
negprods_GPU    = negdata_GPU' * neghidprobs_GPU;
neghidact_GPU   = sum(neghidprobs_GPU);
negvisact_GPU   = sum(negdata_GPU);
%%%%%%%%% END OF NEGATIVE PHASE %%%%%%%%%

err = gather(sqrt(sum(sum((data_mb - negdata_GPU).^2))));
if epoch > 5,
    momentum_GPU = gpuArray(final_momentum);
end

%%%%%%%%% UPDATE WEIGHTS AND BIASES %%%%%%%%%
vishidinc_GPU  = momentum_GPU * vishidinc_GPU  + epsilonw_GPU*( (posprods_GPU-
negprods_GPU)/numcases_GPU - weightcost_GPU * vishid_GPU);
visbiasinc_GPU = momentum_GPU * visbiasinc_GPU +
(epsilonvb_GPU/numcases_GPU)*(posvisact_GPU-negvisact_GPU);
hidbiasinc_GPU = momentum_GPU * hidbiasinc_GPU +
(epsilonhb_GPU/numcases_GPU)*(poshidact_GPU-neghidact_GPU);
vishid_GPU     = vishid_GPU + vishidinc_GPU;
visbiases_GPU  = visbiases_GPU + visbiasinc_GPU;
hidbiases_GPU  = hidbiases_GPU + hidbiasinc_GPU;
%%%%%%%%% END OF UPDATES %%%%%%%%%

function
[w1,w_class,test_err,test_crerr,train_err,train_crerr,confusionMatrixTrain,...
    confusionMatrixTest,mean1OffTrainError,mean1OffTestError,meanTestError,mea
nTrainError,...
    testPredProbs, trainPredProbs] = finetune_RBM_classifier...
    (DN,batchdata,testbatchdata,batchtargets,testbatchtargets)
% Version 1.000
%
```

```
% Code provided by Ruslan Salakhutdinov and Geoff Hinton
% Modified by Scott Hamshaw
%
% Permission is granted for anyone to copy, use, modify, or distribute this
% program and accompanying programs and documents for any purpose, provided
% this copyright notice is retained and prominently displayed, along with
% a note saying that the original programs are available from our
% web page.
% The programs and documents are distributed without any warranty, express or
% implied.  As the programs were written for research purposes only, they have
% not been tested to the degree that would be advisable in any important
% application.  All use of these programs is entirely at the user's own risk.

% This program fine-tunes an autoencoder with backpropagation.
% Weights of the autoencoder are going to be saved in mnist_weights.mat
% and trainig and test reconstruction errors in mnist_error.mat
% You can also set maxepoch, default value is 200 as in our paper.

maxepoch=200;
fprintf(1,'\nTraining discriminative model on MNIST by minimizing cross
entropy error. \n');

%load MNIST_data_125.mat;

%load Hyst_Grayscale_cleanonly_b25.mat;

%load DBN_Grayscale_clean_25_25_50.mat;
hidrecbiases = DN.L{1,1}.hidbiases;
visbiases = DN.L{1,1}.visbiases;
vishid = DN.L{1,1}.vishid;


[numcases,numdims,numbatches]=size(batchdata);
N=numcases;
K = 15;

%%%% PREINITIALIZE WEIGHTS OF THE DISCRIMINATIVE
MODEL%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

w1=[vishid; hidrecbiases];
%w2=[hidpen; penrecbiases];
%w3=[hidpen2; penrecbiases2];
w_class = 0.1*randn(size(w1,2)+1,K);


%%%%%%%%%% END OF PREINITIALIZATIO OF
WEIGHTS  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

l1=size(w1,1)-1;
l2=size(w1,2);
%l3=size(w3,1)-1;
%l4=size(w_class,1)-1;
l5=K;
test_err=[];
```

```
train_err=[];


for epoch = 1:maxepoch

%%%%%%%%%%%%%%%%%%% COMPUTE TRAINING MISCLASSIFICATION ERROR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
err=0;
err_cr=0;
counter=0;
[numcases,numdims,numbatches]=size(batchdata);
N=numcases;

confusionMatrixTest = zeros(K+1,K+1);
confusionMatrixTrain = zeros(K+1,K+1);
trainPredProbs = [];
 for batch = 1:numbatches
  data = batchdata(:,:,batch);
  target = batchtargets(:,:,batch);
  data = [data ones(N,1)];
  w1probs = 1./(1 + exp(-data*w1)); w1probs = [w1probs  ones(N,1)];
  %w2probs = 1./(1 + exp(-w1probs*w2)); w2probs = [w2probs ones(N,1)];
  %w3probs = 1./(1 + exp(-w2probs*w3)); w3probs = [w3probs  ones(N,1)];
  targetout = exp(w1probs*w_class);
  targetout = targetout./repmat(sum(targetout,2),1,K);
  trainPredProbs = [trainPredProbs; targetout];
  [I,J]=max(targetout,[],2);
  [I1,J1]=max(target,[],2);
  counter=counter+length(find(J==J1));
  err_cr = err_cr- sum(sum( target(:,1:end).*log(targetout))) ;
  % create matrix of error by class type
  for i = 1:numcases
     confusionMatrixTrain(J(i),J1(i)) = confusionMatrixTrain(J(i),J1(i))+1;
  end

 end


 confusionMatrixTrain(1:K,K+1) =
diag(confusionMatrixTrain(1:K,1:K))./sum(confusionMatrixTrain(1:K,1:K),2);
 confusionMatrixTrain(K+1,1:K) =
diag(confusionMatrixTrain(1:K,1:K))'./sum(confusionMatrixTrain(1:K,1:K),1);


 train_err(epoch)=(numcases*numbatches-counter);
 train_crerr(epoch)=err_cr/numbatches;

%%%%%%%%%%%%%% END OF COMPUTING TRAINING MISCLASSIFICATION ERROR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%% COMPUTE TEST MISCLASSIFICATION ERROR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
err=0;
err_cr=0;
```

```
counter=0;
[testnumcases,testnumdims,testnumbatches]=size(testbatchdata);
N=testnumcases;
testPredProbs = [];
for batch = 1:testnumbatches
  data = testbatchdata(:,:,batch);
  target = testbatchtargets(:,:,batch);
  data = [data ones(N,1)];
  w1probs = 1./(1 + exp(-data*w1)); w1probs = [w1probs  ones(N,1)];
  %w2probs = 1./(1 + exp(-w1probs*w2)); w2probs = [w2probs ones(N,1)];
  %w3probs = 1./(1 + exp(-w2probs*w3)); w3probs = [w3probs  ones(N,1)];
  targetout = exp(w1probs*w_class);
  targetout = targetout./repmat(sum(targetout,2),1,K);
  testPredProbs = [testPredProbs; targetout];
  [I,J]=max(targetout,[],2);
  [I1,J1]=max(target,[],2);
  counter=counter+length(find(J==J1));
  err_cr = err_cr- sum(sum( target(:,1:end).*log(targetout))) ;

   % create matrix of error by class type

    for i = 1:numcases
     confusionMatrixTest(J(i),J1(i)) = confusionMatrixTest(J(i),J1(i)) +1;
     end

end


 confusionMatrixTest(1:K,K+1) =
diag(confusionMatrixTest(1:K,1:K))./sum(confusionMatrixTest(1:K,1:K),2);
 confusionMatrixTest(K+1,1:K) =
diag(confusionMatrixTest(1:K,1:K))'./sum(confusionMatrixTest(1:K,1:K),1);

 test_err(epoch)=(testnumcases*testnumbatches-counter);
 test_crerr(epoch)=err_cr/testnumbatches;
 fprintf(1,'Before epoch %d Train # misclassified: %d (from %d). Test #
misclassified: %d (from %d) \t \t \n',...
            epoch,train_err(epoch),numcases*numbatches,test_err(epoch),testnum
cases*testnumbatches);

%%%%%%%%%%%%% END OF COMPUTING TEST MISCLASSIFICATION ERROR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 for batch = 1:numbatches
 fprintf(1,'epoch %d batch %d\r',epoch,batch);

 data = batchdata(:,:,batch);
 targets = batchtargets(:,:,batch);

%%%%%%%%%%%%%% PERFORM CONJUGATE GRADIENT WITH 3 LINESEARCHES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  max_iter=3;

  if epoch<6  % First update top-level weights holding other weights fixed.
```

```
    N = size(data,1);
    XX = [data ones(N,1)];
    w1probs = 1./(1 + exp(-XX*w1)); %w1probs = [w1probs  ones(N,1)];
    %w2probs = 1./(1 + exp(-w1probs*w2)); w2probs = [w2probs ones(N,1)];
    %w3probs = 1./(1 + exp(-w2probs*w3)); %w3probs = [w3probs  ones(N,1)];

    VV = (w_class(:)')';
    Dim = [l2; l5];
    [X, fX] =
minimize(VV,'CG_CLASSIFY_INIT_RBM',max_iter,Dim,w1probs,targets,K);
    w_class = reshape(X,l2+1,l5);

  else
    %VV = [w1(:)' w2(:)' w3(:)' w_class(:)'];
    VV = [w1(:)' w_class(:)'];
    %Dim = [l1; l2; l3; l4; l5];
    Dim = [l1; l2; l5];
    [X, fX] = minimize(VV,'CG_CLASSIFY_RBM',max_iter,Dim,data,targets,K);

    w1 = reshape(X(1:(l1+1)*l2),l1+1,l2);
    xxx = (l1+1)*l2;
    %w2 = reshape(X(xxx+1:xxx+(l2+1)*l3),l2+1,l3);
    %xxx = xxx+(l2+1)*l3;
    %w3 = reshape(X(xxx+1:xxx+(l3+1)*l4),l3+1,l4);
    %xxx = xxx+(l3+1)*l4;
    w_class = reshape(X(xxx+1:xxx+(l2+1)*l5),l2+1,l5);

  end
%%%%%%%%%%%%%% END OF CONJUGATE GRADIENT WITH 3 LINESEARCHES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 end

 save weights_RBM_Grayscale_synth_25 w1 w_class
 save error_RBM_Grayscale_synth_25 test_err test_crerr train_err train_crerr
confusionMatrixTrain confusionMatrixTest;


end

%meanTestError = test_err(end)/(testnumbatches*testnumcases)
%meanTrainError = train_err(end)/(numbatches*numcases)
confMatrix1OffInd = [1 2 3 4 9 13 14 18 28 35 49 52 53 68 69 70 72 76 85 86 87
...
    103 104 118 119 120 129 137 138 153 154 155 170 171 188 205 222];
mean1OffTrainError =
sum(sum(confusionMatrixTrain(confMatrix1OffInd)))/sum(sum(confusionMatrixTrain
(1:14,1:14)));
mean1OffTestError =
sum(sum(confusionMatrixTest(confMatrix1OffInd)))/sum(sum(confusionMatrixTest(1
:14,1:14)));
meanTestError =
sum(diag(confusionMatrixTest(1:14,1:14)))/sum(sum(confusionMatrixTest(1:14,1:1
4)))
```

```
meanTrainError =
sum(diag(confusionMatrixTrain(1:14,1:14)))/sum(sum(confusionMatrixTrain(1:14,1
:14)))

% Version 1.000
%
% Code provided by Ruslan Salakhutdinov and Geoff Hinton
% Modified by Scott Hamshaw
%
% Permission is granted for anyone to copy, use, modify, or distribute this
% program and accompanying programs and documents for any purpose, provided
% this copyright notice is retained and prominently displayed, along with
% a note saying that the original programs are available from our
% web page.
% The programs and documents are distributed without any warranty, express or
% implied.  As the programs were written for research purposes only, they have
% not been tested to the degree that would be advisable in any important
% application.  All use of these programs is entirely at the user's own risk.


function [f, df] = CG_CLASSIFY_INIT_RBM(VV,Dim,w1probs,target,K);
l1 = Dim(1);
l5 = Dim(2);
N = size(w1probs,1);
% Do decomversion.
  w_class = reshape(VV,l1+1,l5);
  w1probs = [w1probs  ones(N,1)];

  targetout = exp(w1probs*w_class);
  targetout = targetout./repmat(sum(targetout,2),1,K);
  f = -sum(sum( target(:,1:end).*log(targetout))) ;
IO = (targetout-target(:,1:end));
Ix_class=IO;
dw_class =  w1probs'*Ix_class;

df = [dw_class(:)'];


% Version 1.000
%
% Code provided by Ruslan Salakhutdinov and Geoff Hinton
% Modified by Scott Hamshaw
%
% Permission is granted for anyone to copy, use, modify, or distribute this
% program and accompanying programs and documents for any purpose, provided
% this copyright notice is retained and prominently displayed, along with
% a note saying that the original programs are available from our
% web page.
% The programs and documents are distributed without any warranty, express or
% implied.  As the programs were written for research purposes only, they have
% not been tested to the degree that would be advisable in any important
% application.  All use of these programs is entirely at the user's own risk.
```

```
function [f, df] = CG_CLASSIFY_INIT(VV,Dim,w3probs,target,K);
l1 = Dim(1);
l2 = Dim(2);
N = size(w3probs,1);
% Do decomversion.
  w_class = reshape(VV,l1+1,l2);
  w3probs = [w3probs  ones(N,1)];

  targetout = exp(w3probs*w_class);
  targetout = targetout./repmat(sum(targetout,2),1,K);
  f = -sum(sum( target(:,1:end).*log(targetout))) ;
IO = (targetout-target(:,1:end));
Ix_class=IO;
dw_class =  w3probs'*Ix_class;

df = [dw_class(:)'];
```

```
%%% Program to perform data pre-processing

addpath('Output Data');
addpath('../TimeSeries');
addpath('../Plotting');

%rainData = load('../Data/Mad/MetData/Mad_AllRainData_Hr');
rainData = load('../Data/Mad/MetData/MAD_AvgRain_Hr.mat');
load('../Data/Mad/MetData/WEM_SoilMoist_2013-2015_15Min.mat');
SiteCode = 'MIL'

load sedRatingCurveParams;

switch SiteCode
    case 'MAD'
        load('../Data/Mad/SensorData/MAD_Sensor_2012-2015_Hourly');
        load ('../Data/Mad/EventData/MAD_Online_2012-2015_Daily.mat');
        str1 = 'Mad River';
        a = MAD.a; SF=MAD.SF; b=MAD.b;
        discharge = discharge/35.31; % convert discharge from cfs to m^3/s
        convFact = 1000;
        site = 1;
        staPriority = [8 1 5 7 6 2 4 3];
        catchArea = 360008350; %square meters
        smoothingParam = [3 21];
%     case 'FOL'
%         load('../Data/Mad/SensorData/FOL_Sensor_2013-2015_15Min');
%         load ('../Data/Mad/EventData/FOL_Events_2013-2015.mat');
%         str1 = 'Folsom Brook';
%         a = FOL.a; SF=FOL.SF; b=FOL.b;
%         convFact = 1000;
%         site = 5;
%         staPriority = [4 1 8 2 6 5 7 3];
%         catchArea = 18197700; %square meters
%         smoothingParam = [4 11];
%     case 'FRE'
%         load('../Data/Mad/SensorData/FRE_Sensor_2013_15Min');
%         load ('../Data/Mad/EventData/FRE_Events_2013.mat');
%         a = FRE.a; SF=FRE.SF; b=FRE.b;
%         convFact = 1000;
%         site = 6;
%         staPriority = [4 6 1 8 2 5 7 3];
%         p = [-4.8018,83.01,-27];
%         discharge = polyval(p,stage*3.2808)/35.31;
%         catchArea = 16962300; %square meters
%         smoothingParam = [4 11];
    case 'MIL'
        load('../Data/Mad/SensorData/MIL_Sensor_2012-2015_HR');
        load ('../Data/Mad/EventData/MIL_Online_2012-2015.mat');
        str1 = 'Mill Brook';
```

```
        a = MIL.a; SF=MIL.SF; b=MIL.b;
        convFact = 1000;
        site = 4;
        staPriority = [2 1 8 5 6 4 7 3];
        catchArea = 49823000; %square meters
        smoothingParam = [4 11];
    case 'SHP'
        load('../Data/Mad/SensorData/SHP_Sensor_2013-2015_HR');
        load ('../Data/Mad/EventData/SHP_Online_2013-2015.mat');
        str1 = 'Shepard Brook';
        a = SHP.a; SF=SHP.SF; b=SHP.b;
        convFact = 1000;
        site = 2;
        staPriority = [5 8 1 2 3 7 4 6];
        catchArea = 44611100; %square meters
        smoothingParam = [4 11];
%     case 'HBR'
%         load('../Data/Mad/SensorData/HBR_Sensor_2013_15Min');
%         load ('../Data/Mad/EventData/HBR_Events_2013.mat');
%         str1 = 'High Bridge Brook';
%         a = HBR.a; SF=HBR.SF; b=HBR.b;
%         convFact = 1000;
%         site = 3;
%         staPriority = [8 1 4 6 5 3 7 2];
%         discharge = (1.0376.*27.12.*(stage*3.28).^2.1277)/35.31;
%         catchArea = 8643600; %square meters
%         smoothingParam = [4 11];
end

TSS =(SF*a*(turbidity).^b); % turbidity based estimate of TSS (mg/L)

%convert precip from in to mm

% rainData.allRainData = rainData.allRainData .* 25.4;
% rainAmt = NaN(size(rainData.allRainData,1),1);
% for i = 1:size(rainData.allRainData,1)
%     siteNaN = isnan(rainData.allRainData(i,:));
%
%     siteInd = find(siteNaN == 0);
%     sitesOnline = ismember(staPriority,siteInd);
%     sitesOnline = staPriority(sitesOnline);
%     rainTot = sum(rainData.allRainData(i,sitesOnline),1,'omitnan');
%     rainGaugeWeighting = autoThiessenPoly(sitesOnline);
%     rainAmt(i) = sum(rainTot.*rainGaugeWeighting);
% end

% calculate number of storm events
numEvents = length(x)/2;
% determine start and stop of windows to extract for storm event analysis
% and convert to datetime arrays
ind = 1:2:numEvents*2;
startDates = datetime(x(ind),'ConvertFrom','datenum');
ind = 2:2:numEvents*2;
endDates = datetime(x(ind),'ConvertFrom','datenum');
```

```
IscoData = dataExtractorNum([startDates endDates],[numDate discharge TSS
waterTemp],1);
numDate = IscoData(:,1); discharge = IscoData(:,2); TSS= IscoData(:,3);
waterTemp = IscoData(:,4);

PrecipData = dataExtractorNum([startDates endDates],[rainData.rainNumDate
rainData.rainAmt],1);
rainData.rainNumDate = PrecipData(:,1); rainAmt = PrecipData(:,2);

PrecipData = dataExtractorNum([startDates endDates],[rainNumDate WC2 WC3 WC1
WC4],1);
wcNumDate = PrecipData(:,1); WC2 = PrecipData(:,2); WC3 = PrecipData(:,3); WC1
= PrecipData(:,4); WC4 = PrecipData(:,5);

rainNumDate = rainData.rainNumDate;

% match rain amounts to stream data time steps
rainAmtMatched = zeros(length(numDate),1);
vwc1Matched = zeros(length(numDate),1);
vwc2Matched = zeros(length(numDate),1);
vwc3Matched = zeros(length(numDate),1);
vwc4Matched = zeros(length(numDate),1);
for i = 1:length(numDate)
    datelow = numDate(i)-1/1440;
    datehigh = numDate(i) + 1/1440;
    ind = find(rainNumDate > datelow & rainNumDate < datehigh);
    TF = isempty(ind);
    if TF == 0
        if numel(ind) > 1
            ind = ind(1);
        end
        rainAmtMatched(i) = rainAmt(ind);
    end
    ind = find(wcNumDate > datelow & wcNumDate < datehigh);
    TF = isempty(ind);
    if TF == 0
        if numel(ind) > 1
            ind = ind(1);
        end
        vwc1Matched(i) = WC1(ind);
        vwc2Matched(i) = WC2(ind);
        vwc3Matched(i) = WC3(ind);
        vwc4Matched(i) = WC4(ind);
    end
end

% to remove later
% [n, bin] = histc(numDate, unique(numDate));
% multiple = find(n > 1);
% index    = find(ismember(bin, multiple));
```

```
% numDate(index)
t = 1:length(TSS);
t = t/96;

[fh ah] = scrollPlot3(14,t,[rainAmtMatched discharge vwc2Matched],[],'on');
title(ah(3),{'Mill Brook','Precipitation, Discharge, and SM'})
grid(ah(1)); grid(ah(2)); grid(ah(3));
ylabel(ah(1),'Turbidity (NTU)');
ylabel(ah(2),'Discharge (cfs)');
xlabel(ah(3),'Time (Days)');
xlabel(ah(1),'Time (Days)');
xlabel(ah(2),'Time (Days)');
ylabel(ah(3),'15-Min Rainall (in)');
h =findobj('Type','line');
set(h(1),'Color','red','LineWidth',2);
set(h(2),'Color','blue','LineWidth',2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Driver program for running recurrent CPN w/time series data
% Scott Hamshaw
% Created 19-Sep-2013
% Last Revised
%%%%%%%%%%%%%%%%%%%%%%%%

% NOTES:
addpath('../TimeSeries');
addpath('../ANNs');
addpath('../Normalization');
addpath('../Stats');

close all;
disp('Program started...');
tic;

siteCode = 'MAD'
switch siteCode
    case 'MAD'

        load('Output Data/MAD_Hr_rCPNdata.mat');
        load('../Data/Mad/EventData/MAD_Events_2013-2015');

        trainStartInd = 3589;   % 7/31/13 11:00
        testStartInd = 10660;   % 4/28/15 00:00
        testEndInd = 16048;     % 12/8/15 12:00;
        trainStartStormInd = 25;
        testStartStormInd = 96;
        testEndStormInd = 143;

    case 'SHP'

        load('Output Data/SHP_Hr_rCPNdata.mat');
        load('../Data/Mad/EventData/SHP_Events_2013-2015');

        trainStartInd = 468;   % 7/31/13 11:00
```

```
        testStartInd = 7392;   % 4/28/15 00:00
        testEndInd = 12757;     % 12/8/15 12:00;
        trainStartStormInd = 5;
        testStartStormInd = 63;
        testEndStormInd = 102;

    case 'MIL'
        load('Output Data/MIL_Hr_rCPNdata.mat');
        load('../Data/Mad/EventData/MIL_Events_2012-2015');

        trainStartInd = 2802;   % 7/31/13 11:00
        testStartInd = 10000;   % 4/28/15 00:00
        testEndInd = 15053;     % 12/8/15 12:00;
        trainStartStormInd = 36;
        testStartStormInd = 107;
        testEndStormInd = 154;
end


ind = find(vwc1Matched == 0);
vwc1Matched(ind) = NaN; vwc2Matched(ind) = NaN; vwc3Matched(ind) = NaN;
vwc4Matched(ind) = NaN;
ind = find(waterTemp >= 25);
waterTemp(ind) = NaN;

% determine start and stop of windows to extract for storm events
% and convert to datetime arrays
numEvents = length(x)/2;
ind = 1:2:numEvents*2;
startDates = datetime(x(ind),'ConvertFrom','datenum');
ind = 2:2:numEvents*2;
endDates = datetime(x(ind),'ConvertFrom','datenum');

for i = 1:numEvents-1
    if startDates(i+1) <= endDates(i)
        startDates(i+1) = endDates(i)+1/24;
    end
end


%discharge = (log(discharge)+1)*10;
stormflow = discharge - baseflow;
ind = find(stormflow <0);
stormflow (ind) = 0;

% Setup output classes and convert prediciton variable to classificaiton
K = 100;
targetClasses = linspace(1,K)';
targetQClasses = logspace(0,2.5,K)'; targetQClasses =
round(targetQClasses*10)/10;
% targetQClasses = linspace(1,ceil(max(discharge)),K);
targetBaseflowClasses = logspace(0,1.5,K)'; targetBaseflowClasses =
round(targetBaseflowClasses*10)/10;
% targetBaseflowClasses = linspace(1,ceil(max(baseflow)),K);
```

```
targetTSSClasses = logspace(0,3.25,K)'; targetTSSClasses =
round(targetTSSClasses*10)/10;
% targetTSSClasses = linspace(1,ceil(max(TSS)),K);
targetDischarge = roundtowardvec(discharge,targetQClasses,'round');
[~, targetDischargeClassNum] = ismember(targetDischarge, targetQClasses);
targetBaseflow = roundtowardvec(baseflow,targetBaseflowClasses,'round');
[~, targetBaseflowClassNum] = ismember(targetBaseflow, targetBaseflowClasses);
targetStormflow = roundtowardvec(stormflow,targetQClasses,'round');
[~, targetStormflowClassNum] = ismember(targetStormflow, targetQClasses);
targetTSS = roundtowardvec(TSS,targetTSSClasses,'round');
[~, targetTSSClassNum] = ismember(targetTSS, targetTSSClasses);



% create lagged time values of variables
lagDischarge = timelagdata(discharge,12);
lagPrecip = timelagdata(rainAmtMatched,24);
lagVWC1 = timelagdata(vwc1Matched,12);
lagVWC2 = timelagdata(vwc2Matched,12);
lagVWC3 = timelagdata(vwc3Matched,12);
lagVWC4 = timelagdata(vwc4Matched,12);
lagTemp = timelagdata(waterTemp,12);
lagBaseflow = timelagdata(baseflow,12);
lagStormflow = timelagdata(stormflow,12);
lagTSS = timelagdata(TSS,12);

alpha=0.7;  %Kohonen weight layer learning constant
beta=0.1;   %Grosberg weight layer learning constant
Thresh=0.00001; %Network will train until this threshold is met
maxIter = 100;
numRecNodes = 1;

% %% stormflow rCPN
% % build input data matrices
% data1 = [lagPrecip(:,2:6) lagVWC1(:,2:3) targetStormflowClassNum]; % +1
recurrent nodes
% trainData1 = data1(trainStartInd:testStartInd-1,:);
% testData1 = data1(testStartInd:testEndInd,1:7);
% typeKey1 = [1 1 1 1 1 2 2 3];
%
% tempData =
dataExtractorNum([startDates(trainStartStormInd:testStartStormInd-1)
endDates(trainStartStormInd:testStartStormInd-1)],...
%     [numDate(trainStartInd:testStartInd-1) trainData1
targetStormflow(trainStartInd:testStartInd-1)
rainAmtMatched(trainStartInd:testStartInd-1)],1);
%     numDateTrainCrop = tempData(:,1);
%     trainData1 = tempData(:,2:end-2);
%     targetStormflowTrainCrop = tempData(:,end-1);
%     rainAmtTrainCrop = tempData(:,end-1);
%
```

```
%     tempData =
dataExtractorNum([startDates(testStartStormInd:testEndStormInd)
endDates(testStartStormInd:testEndStormInd)],...
%     [numDate(testStartInd:testEndInd) testData1
targetStormflow(testStartInd:testEndInd)
rainAmtMatched(testStartInd:testEndInd)],1);
%     numDateTestCrop = tempData(:,1);
%     testData1 = tempData(:,2:end-2);
%     targetStormflowTestCrop = tempData(:,end-1);
%     rainAmtTestCrop = tempData(:,end-1);
%
%
% [predictTrain, predictTest,~,~] = rCPNv2(trainData1,testData1,1,1,...
%     2,1,alpha,beta,Thresh,0,typeKey1,50,1,400,1,'_weights_stormflow');
% ind = find(predictTest == 0);
% predictTest(ind) = 1;
% for i = 1:length(predictTrain)
%     estTrainStormflow(i) = targetQClasses(predictTrain(i));
% end
% for i = 1:length(predictTest)
%     estTestStormflow(i) = targetQClasses(predictTest(i));
% end
% disp('Training Data - Stormflow');
% evalANN(estTrainStormflow',targetStormflowTrainCrop,numDateTrainCrop,...
%     rainAmtTrainCrop,'Stormflow (cms)','Training Data - Stormflow',1);
%
% disp('Testing Data - Stormflow');
% evalANN(estTestStormflow',targetStormflowTestCrop,numDateTestCrop,...
%     rainAmtTestCrop,'Stormflow (cms)','Testing Data - Stormflow',0);
%
% estStormflow = zeros(length(numDate),1);
% for i = 1:length(estStormflow)
%     datelow = numDate(i)-1/1440;
%     datehigh = numDate(i) + 1/1440;
%     if i < testStartInd
%         ind = find(numDateTrainCrop(2:end) > datelow &
numDateTrainCrop(2:end) < datehigh);
%     else
%         ind = find(numDateTestCrop > datelow & numDateTestCrop < datehigh);
%     end
%     TF = isempty(ind);
%
%     if TF == 0
%         if numel(ind) > 1
%             ind = ind(1);
%         end
%         if i < testStartInd
%             estStormflow(i) = estTrainStormflow(ind);
%         else
%             estStormflow(i) = estTestStormflow(ind);
%         end
%     end
%
% end
```

```
%
% lagEstStormflow = timelagdata(estStormflow,12);
%
% %% baseflow rCPN
% data2 = [lagTemp(:,6) lagPrecip(:,12) vwc4Matched lagVWC1(:,5)
targetBaseflowClassNum]; % +1 recurrent node
% trainData2 = data2(trainStartInd:testStartInd-1,:);
% testData2 = data2(testStartInd:testEndInd,1:4);
% typeKey2 = [1 2 3 4 5];
%
% [predictTrain, predictTest,~,~] = rCPNv2(trainData2,testData2,3,1,...
%     2,1,alpha,beta,Thresh,0,typeKey2,25,1,400,1,'_weights_baseflow');
% ind = find(predictTest == 0);
% predictTest(ind) = 1;
% ind = find(predictTrain == 0);
% predictTrain(ind) = 1;
% for i = 1:length(predictTrain)
%     estTrainBaseflow(i) = targetBaseflowClasses(predictTrain(i));
% end
% for i = 1:length(predictTest)
%     estTestBaseflow(i) = targetBaseflowClasses(predictTest(i));
% end
%
% disp('Training Data - Baseflow');
% evalANN(estTrainBaseflow',targetBaseflow(trainStartInd:testStartInd-
1),numDate(trainStartInd:testStartInd-1),...
%     rainAmtMatched(trainStartInd:testStartInd-1),'Baseflow (cms)','Training
Data - Baselow',1);
%
% disp('Testing Data - Baseflow');
%
evalANN(estTestBaseflow',targetBaseflow(testStartInd:testEndInd),numDate(testS
tartInd:testEndInd),...
%     rainAmtMatched(testStartInd:testEndInd),'Baseflow (cms)','Testing Data -
Baselow',0);
%
% disp('Training Data - Discharge');
% evalANN((estTrainBaseflow'+estStormflow(trainStartInd+1:testStartInd-
1)),targetDischarge(trainStartInd:testStartInd-
1),numDate(trainStartInd:testStartInd-1),...
%     rainAmtMatched(trainStartInd:testStartInd-1),'Discharge (cms)','Training
Data - Discharge',1);
%
% disp('Testing Data - Discharge');
%
evalANN((estTestBaseflow'+estStormflow(testStartInd:testEndInd)),targetDischar
ge(testStartInd:testEndInd),numDate(testStartInd:testEndInd),...
%     rainAmtMatched(testStartInd:testEndInd),'Discharge (cms)','Testing Data
- Discharge',0);
%
% estBaseflow = [repmat(0,trainStartInd-1,1);
baseflow(trainStartInd:testStartInd-1); ...
%     repmat(1,numRecNodes,1); estTestBaseflow';repmat(0,16515-testEndInd,1)];
% lagEstBaseflow = timelagdata(estBaseflow,12);
```

```
%
% estDischarge = [repmat(0,trainStartInd-1,1);
discharge(trainStartInd:testStartInd-1); ...
%     repmat(1,numRecNodes,1);
(estTestBaseflow'+estStormflow(testStartInd:testEndInd));repmat(0,16515-
testEndInd,1)];
% lagEstDischarge = timelagdata(estDischarge,12);
% %% TSS with baseflow and stormflow rCPN
%
%
% data3 = [estDischarge lagEstDischarge(:,1:2) lagPrecip(:,6:7) lagVWC1(:,4)
targetTSSClassNum]; % +1 recurrent node
% trainData3 = data3(trainStartInd:testStartInd-1,:);
% testData3 = data3(testStartInd:testEndInd,1:6);
% typeKey3 = [1 1 1 2 2 3 4];
%
% [predictTrain, predictTest,~,~] = rCPNv2(trainData2,testData2,3,1,...
%     2,1,alpha,beta,Thresh,0,typeKey2,100,2,maxIter,1,'_weights_TSS');
%
% for i = 1:length(predictTrain)
%     estTrainTSS(i) = targetTSSClasses(predictTrain(i));
% end
% for i = 1:length(predictTest)
%     estTestTSS(i) = targetTSSClasses(predictTest(i));
% end
%
% disp('Training Data - TSS');
% evalANN(estTrainTSS',targetTSS(trainStartInd:testStartInd-
1),numDate(trainStartInd:testStartInd-1),...
%     rainAmtMatched(trainStartInd:testStartInd-1),'TSS (mg/L)','Training Data
- TSS',1);
%
% disp('Testing Data - TSS');
%
evalANN(estTestTSS',targetTSS(testStartInd:testEndInd),numDate(testStartInd:te
stEndInd),...
%     rainAmtMatched(testStartInd:testEndInd),'TSS (mg/L)','Training Data -
TSS',0);

%%
data4 = [waterTemp vwc3Matched lagVWC1(:,3:4) lagPrecip(:,5:8)
targetDischargeClassNum]; % +2 recurrent node
trainData4 = data4(trainStartInd:testStartInd-1,:);
testData4 = data4(testStartInd:testEndInd,1:8);
typeKey4 = [1 2 3 3 5 5 5 5 6];

[predictTrain, predictTest,~,~] = rCPNv2(trainData4,testData4,1,1,...
    2,1,alpha,beta,Thresh,0,typeKey4,100,2,40,1,'_weights_discharge');

ind = find(predictTest == 0);
predictTest(ind) = 1;
ind = find(predictTrain == 0);
predictTrain(ind) = 1;
for i = 1:length(predictTrain)
```

```
    estTrainDischarge(i) = targetQClasses(predictTrain(i));
end
for i = 1:length(predictTest)
    estTestDischarge(i) = targetQClasses(predictTest(i));
end

disp('Training Data - Discharge');
evalANN(estTrainDischarge',targetDischarge(trainStartInd:testStartInd-
1),numDate(trainStartInd:testStartInd-1),...
    rainAmtMatched(trainStartInd:testStartInd-1),'Discharge (cms)','Training
Data - Dischaerge',1);

disp('Testing Data - Discharge');
evalANN(estTestDischarge',targetDischarge(testStartInd:testEndInd),numDate(tes
tStartInd:testEndInd),...
    rainAmtMatched(testStartInd:testEndInd),'Discharge (cms)','Testing Data -
Discharge',0);

estDischarge = [repmat(0,trainStartInd-1,1);
discharge(trainStartInd:testStartInd-1); ...
    repmat(1,numRecNodes,1); estTestDischarge';repmat(0,16515-testEndInd,1)];
lagEstDischarge = timelagdata(estDischarge,12);

save('MAD_Q3','discharge','estTestDischarge','estTrainDischarge','numDate','es
tDischarge','rainAmtMatched','vwc1Matched','vwc3Matched','waterTemp')
%%
data5 = [estDischarge lagEstDischarge(:,1) lagPrecip(:,6:8) lagVWC1(:,1:2)
lagTemp(:,12) targetTSSClassNum]; % +2 recurrent node
trainData5 = data5(trainStartInd:testStartInd-1,:);
testData5 = data5(testStartInd:testEndInd,1:8);

typeKey5 = [1 1 2 2 2 3 3 4 5];

[predictTrain, predictTest,~,~] = rCPNv2(trainData5,testData5,1,1,...
    2,1,alpha,beta,Thresh,0,typeKey5,500,2,40,1,'_weights_TSS');

ind = find(predictTest == 0);
predictTest(ind) = 1;
for i = 1:length(predictTrain)
    estTrainTSS(i) = targetTSSClasses(predictTrain(i));
end
for i = 1:length(predictTest)
    estTestTSS(i) = targetTSSClasses(predictTest(i));
end

disp('Training Data - TSS');
evalANN(estTrainTSS',targetTSS(trainStartInd:testStartInd-
1),numDate(trainStartInd:testStartInd-1),...
    rainAmtMatched(trainStartInd:testStartInd-1),'TSS (mg/L)','Training Data -
TSS',1);

disp('Testing Data - TSS');
evalANN(estTestTSS',targetTSS(testStartInd:testEndInd),numDate(testStartInd:te
stEndInd),...
```

```
    rainAmtMatched(testStartInd:testEndInd),'TSS (mg/L)','Testing Data -
TSS',0);


save('MAD_TSS2','estTrainTSS','estTestTSS','TSS','discharge','estTestDischarge
','estTrainDischarge','numDate','estDischarge','rainAmtMatched','vwc1Matched',
'vwc3Matched','waterTemp')
%%

% plot



% testRegPredict = -74.765 +
0.463*lagPrecip(testStartInd:testEndInd,6)+0.661*lagPrecip(testStartInd:testEn
dInd,6) ...
%     +0.354*vwc1Matched(testStartInd:testEndInd) +
2.268*vwc2Matched(testStartInd:testEndInd)...
%     +0.216*vwc3Matched(testStartInd:testEndInd)
+0.083*vwc4Matched(testStartInd:testEndInd)...
%     +0.401*waterTemp(testStartInd:testEndInd);

% prep training data
% data = rainAmtMatched(trainStartInd:testStartInd-1);
% data2 = discharge(trainStartInd:testStartInd-1);
%
% lag_output1 = timelagdata(data(:,size(data,2)),lags);
% Nind = find(isnan(lag_output1)); %% find indeces with NaNs (due to time lag)
% lag_output1(Nind) = 0;
% lag_output2 = timelagdata(data2(:,size(data,2)),2);
% Nind = find(isnan(lag_output2)); %% find indeces with NaNs (due to time lag)
% lag_output2(Nind) = 0;

% trainData = [lag_output1 vwc2Matched(trainStartInd:testStartInd-1)
vwc3Matched(trainStartInd:testStartInd-1)...
%     vwc4Matched(trainStartInd:testStartInd-1)
vwc1Matched(trainStartInd:testStartInd-1) ...
%     waterTemp(trainStartInd:testStartInd-1)
targetClass(trainStartInd:testStartInd-1)];


% Set Coefficients and Constants
alpha=0.5;  %Kohonen weight layer learning constant
beta=0.1;   %Grosberg weight layer learning constant
Thresh=0.00001; %Network will train until this threshold is met
numWIN = 1; %Num of K-winners for IDW interpolation
lag_outputs = 2; %number of recurrent nodes for target data
seed = 1*ones(lag_outputs,1); % initialization values for recurrent nodes
norm = 2; %normalization mode  1=L2_
numHidNodes = 158;
wijInitMode = 1; % 1 for random, 2 for set equal to normalized training data
maxIterations = 25;
trainMode = 1; %1 = random training data, 2 = sequential training data
fname = 'Weights_rCPN_150' % filename for saved weights;
```

```
%%
%[TRAIN,INTERP,winnode]=CounterProp_recurrent(raw,test,numWIN,lag_outputs,norm
,seed,)
% [train, interp,Wij,Wjk] = rCPNv2(trainData,testData,numWIN,lag_outputs,...
%     norm,seed,alpha,beta,Thresh,0,typeKey,numHidNodes,wijInitMode,maxIter,tr
ainMode,fname);
% %% Evaluation
% t = 1:length(interp);
% t = t/24;
% figure()
% subplot(3,1,2:3)
% h = plot(t,testTarget,t,interp);
% xlabel('Time (days)');
% ylabel('Turbidity (NTU)');
% legend('Observed Turbidity','Estimated Turbidity','Location','Northwest')
% set(h(2),'LineWidth',1.5,'LineStyle','--','Color','m')
% set(h(1),'LineWidth',1.5,'Color','k')
% subplot(3,1,1)
% bar(t,rainAmtMatched(testStartInd:testEndInd));
% set(gca,'Ydir','reverse');
% ylabel('15-Minute Rainfall (in)');
% xlabel('Time (days)');
%
% figure()
% x = 0.1:max(testTarget);
% loglog(testTarget,interp,'.');
% xlim([0.1 10000]);
% ylim([0.1 10000]);
%
%
% hold on
% plot(x,x);
% xlabel('Observed Turbidity (NTU)','FontSize',12);
% ylabel('Predicted Turbidity (NTU)','FontSize',12);
% str1 = {'Mill Brook - 8/1/13 - 9/23/13', 'Sugarbush Rain Gauge'};
% title(str1,'FontSize',12);
%
% figure()
% x = 0:1:max(trainTarget);
% loglog(trainTarget(3:end),train,'.');
% %loglog(trainTarget,train,'.');
%
% figure()
% t = 1:length(train);
% t = t/24;
% subplot(3,1,2:3)
% h = plot(t(2:end),trainTarget(1:end-3),t(2:end),train(1:end-1));
% xlabel('Time (days)');
% ylabel('Turbidity (NTU)');
% legend('Observed Turbidity','Estimated Turbidity','Location','Northwest')
% set(h(2),'LineWidth',1.5,'LineStyle','--','Color','m')
% set(h(1),'LineWidth',1.5,'Color','k')
% subplot(3,1,1)
% bar(t,rainAmtMatched(trainStartInd+2:testStartInd-1));
```

```
% set(gca,'Ydir','reverse');
% ylabel('15-Minute Rainfall (in)');
% xlabel('Time (days)');
%
% RMSE = rmse(testTarget,interp)
% Corr= corr(testTarget,interp)
% Rsquare = rsquare(testTarget,interp)
% %save('FOL_Estimates7.mat','numDate','discharge','turbidity','interp');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Recurrent Counterpropogation Network Function
% Scott Hamshaw
% Created: 19-Oct-2013
% Last Revised: Sept-2017
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% NOTES: Input file must be formatted with last column representing output.
% Rows represent samples and columns represent variables
%
% INPUTS:
% train = training data (n samples/patterns by N+1 variables). last column
% represents target output data
% predict = test/prediction data (n samples patterns by N variables)
% numWin = number of k-winners for interpolation mode
% numRecNode = number of recurrent nodes (number of previous estimated outputs
to
% use as input data)
% normMode = normalization method (1 = none, 2 = L2, 3 = unit sphere)
% initSeed = initial seed to use for recurrent nodes
% alpha = Kohonen weights learning rate
% beta = Grossberg weights learning rate
% trainThresh = training threshold for RMSE
% trainMode = skip training if = 1
%
% OUTPUTS:
%
function [trainLoc,interp,Wij,Wjk] =
rCPNv2(train,predict,numWin,numRecNode,...
    normMode,initSeed,alpha,beta,trainThresh,trainMode,typeKey,numHidNodes,wij
InitMode,...
    maxIterations,trainSort,fname)

trainRaw = train; % store original training data
predictRaw = predict; % store original prediction data
[~,N] = size(train);
[M2,~] = size(predict);
numInVar = N-1; % number of input variables

%% Data Pre-processing
if numRecNode >0
    % Generate lagged values of target data for training
    lagPredict = timelagdata(train(:,N),numRecNode); % call function to time
lag target data
```

```
    ind = find(isnan(lagPredict)); % find indeces with NaNs created by time
lag
    [Nrow,Ncol] = ind2sub(size(lagPredict),ind); % convert indeces
    Ncol = unique(Ncol);
    lagPredict = lagPredict(:,Ncol); % retain only time lagged data
    Nrow = unique(Nrow);
    train = [train(:,1:numInVar) lagPredict train(:,N)]; % create training
data set with

    train(Nrow,:) = []; %%% remove rows that contain NaNs (due to time lag)
end
% add mean of target output to prediction data for normalization purposes
for i = 1:numRecNode
    predict = [predict ones(M2,1)*mean(trainRaw(:,N))];
end

if numRecNode ==0
    train = [train(:,1:numInVar) train(:,N)];
end

% Normalize Data based on menu choice
% normMode = menu('Select Desired Data Normalization for use in CPNN',...
%     'No Normalization / Data Already Normalized','L2 Normalization (use
Euclidean Distance)',...
%     'L2 Normalization then Unit Sphere Normalization (use dot product)');

switch normMode
    case 1 % no normalization
        normTrain = train(:,size(train,2));
        normPredict = predict;
    case 2 % L2 normalization (returns data without target/output column
        disp('Using L2 Normalization')
        [normTrain,normPredict,max_target,min_target]=L2_Norm(train,predict,ty
peKey);
    case 3 % unit sphere (returns data without target/output column but with
extra dimension
        disp('Using Unit Sphere Normalization')
        [normTrain,normPredict,normN]=sphere_norm(train,predict);
end

%K=max(train(:,N)); % set number of output classes (K)
K = 150;
[numPatt,numCol] = size(normTrain);

% convert target data to binary
target = zeros(numPatt,K);       % create empty target vector matrix
for i = 1:numPatt % loop through training patterns
    if(train(i,numCol)==0)
        target(i,1) = 1;  % if zero assign to class 1 to avoid error
    else
        target(i,train(i,N)) = 1; % enter 1 in row corresponding to class
    end
end
```

```
%% Network Training Phase
if trainMode ==0
    disp('Network Training has begun...');
    tic % start timer
    J =  numHidNodes; % set number of Kohonen nodes

    Wij = rand(numCol,J); % initialize Kohonen weights to random numbers

    if wijInitMode == 2
        Wij = normTrain'; % initialize Kohonen Weights
        J = numPatt; % Set number of Kohonen nodes
    end

    Wjk = rand(J,K); % initialize Grossberg weights to random numbers

    iteration = 1;                              % set starting iteration to 1
    RMSE = 1;                                   % set initial RMSE value to 1


   for epoch = 1:maxIterations
        errorCalc = 0; % initialize calculation variable for RMSE

        if trainSort == 1
            randomorder = randperm(numPatt);
            randNormTrain = normTrain(randomorder,:);
            randTarget = target(randomorder,:);
        else
            randNormTrain = normTrain;
            randTarget = target;
        end
            if iteration == maxIterations
                randNormTrain = normTrain;
                randTarget = target;
                disp('max iterations reached');
            end
        for n = 1:numPatt % loop through training patterns

            x = randNormTrain(n,:); % peel off vector of training pattern at
step n

            switch normMode
                case 1 % no normalization
                    for j=1:J % find distance to each Kohonen node
                        dist(j)=sqrt(sum((x'-Wij(:,j)).^2));
                    end

                    [minZ,ind] = min(dist); % find minimum distance
                case 2 % L2 normalization
                    for j=1:J % find distance to each Kohonen node
                        dist(j)=sqrt(sum((x'-Wij(:,j)).^2));
                    end
                    [minZ,ind] = min(dist); % find minimum distance

                case 3 % Unit sphere normalization
```

```
                    for j=1:J % find dot product for each Kohonen node
                        Zj(j)=x * Wij(:,j);
                    end
                    [maxZ,ind] = max(Zj); % find maximum
            end

            Zj = zeros(1,J); % all or nothing winner
            Zj(ind) = 1;

            if iteration ~= maxIterations
                Wij(:,ind) = Wij(:,ind) + alpha*(x'-Wij(:,ind)); % update
Kohonen weights
            end

            yEst=Zj*Wjk; % estimate output/target

            if iteration ~= maxIterations
                Wjk(ind,:)=Wjk(ind,:)+beta*(randTarget(n,:)-yEst); % update
Grosberg weights
            end

            errorCalc=errorCalc+sum((randTarget(n,:)-yEst).^2); % perform
calculation for RMSE

            trainOutput(n,:)=Zj*Wjk; % store output
        end

        % display iteration count every 5th iteration
        RMSE=sqrt(errorCalc/(numPatt*K)); %keep track of RMSE
        if mod(iteration,5) == 0
            disp(['Iteration ',num2str(iteration)])
            disp(num2str(RMSE));
        end

        saveRMSE(iteration) = RMSE;
        iteration = iteration + 1;
        if iteration > (maxIterations/4) && iteration < (maxIterations/4)*3
            alpha = alpha - (alpha - 0.1)*(iteration/((maxIterations/4)*3-
maxIterations/4));
        elseif iteration >= (maxIterations/4)*3
            alpha = 0.05;
        end
    end

    figure(3)
    plot(1:length(saveRMSE),saveRMSE,'b-')
    title('Plot of RMSE vs Iteration')
    xlabel('Iteration')
    ylabel('RMSE')
    % convert target/output data back to numerical value
    for i=1:numPatt
        [Max,ind]=max(trainOutput(i,:));
        trainOut(i,ind)=1;
        trainLoc(i,1)=ind;
```

```
    end

    display('Network Training Completed')
    trainingTime = toc
else
    load ('Output Data/rCPN Output/SHP_TSS1_weights.mat');
    J = size(Wij,2);
    trainLoc = [];
end

%% Prediction Mode

% Normalize seed values based on normalized target
[numTest,numCol] = size(normPredict);
predictOutput = zeros(numTest,1);
switch normMode
    case 1 % no normalization
        ntarget = target;
        nseed = seed;
    case 2 % L2 normalization
        % get min and max of target/output vector then normalize
        % target/output
        %min_target = min(train(:,numCol));
        %max_target = max(train(:,numCol));
        %min_target = 0.7080;
        %max_target = 158.4058;
        ntarget = (train(:,numCol)-min_target)/(max_target-min_target);
        % if seed exists, then normalize that also
        if exist('initSeed')==1
            nseed = (initSeed - min_target)/(max_target-min_target);
        end

    case 3 % unit sphere normalization
        ntarget = train(:,numCol)/normN; % normalize taget using normN from
unit sphere normalization
        place = 0; % initialize extra dimension for unit sphere norm.
        % if a seed exists, then normalize that also
        if exist('initSeed')==1
            nseed = initSeed/normN;
        end
end

for n=1:numTest % loop through training patterns
    % lag output variable in time
    if numRecNode >=1;
        Loutput = []; % initialize empty lagged output vector
        for t = 1:numRecNode % step through recurrent nodes/lagged estimates
            ind = n-t; % compute lag from current time step
            if ind>0 % if lagged estimates exist add them to vector
                Loutput = [Loutput normPredictOutput(ind)];
            else % if lagged estimates doen't exist yet, then use seed value
                if exist('initSeed') == 1;
                    Loutput = [Loutput nseed(abs(ind) + 1)];
                end
```

```
            end
        end
    end
    if numRecNode==0;
        Loutput = [];
    end
    switch normMode
        case 1 % no normalization
            % replace "dummy" mean lagged values with estimates from
            % previous time step calculated above
            normPredict2(n,:) = [normPredict(n,1:size(predictRaw,2)) Loutput];

        case 2 % L2 normalizaton
            % replace "dummy" mean lagged values with estimates from
            % previous time step calculated above
            normPredict2(n,:) = [normPredict(n,1:size(predictRaw,2)) Loutput];

        case 3 % unit sphere normalization
            %             normPredict2(n,:) =
[normPredict(n,1:size(predictRaw,2)) Loutput place]; % add extra dimension to
prediction data set
            %             nuvect(n,:) = [normPredict2(n,[1:numInVar-1])*normN
normPredict2(n,[numInVar:size(normPredict2,2)])*normN];
            %             L(n,1) = sqrt(sum(nuvect(n,:).^2));
            %             d(n,1) = sqrt(normN^2-L(n,1)^2)/normN;
            %             normPredict2(n,size(normPredict2,2)) = d(n,1);
    end

    x=normPredict2(n,:); % strip off input pattern for current iteration

    switch normMode
        case 2 % L2 normalization
            for j=1:J % loop through each Kohonen node
                dist(j)=sqrt(sum((x'-Wij(:,j)).^2)); % calculate distance for
each Kohonen node
            end

            [minZ,ind] = min(dist); % find minimum
            ind = 1:J;
            distTemp = [dist' ind'];
            distTemp = sortrows(distTemp,1); % sort by distance
            kWinners = distTemp(1:numWin,2);
            kDist = distTemp(1:numWin,1);


        case 3 % unit sphere normalization
            for j=1:J % loop through each Kohonen node

                Zj(j)=x * Wij(:,j); % calculate dot product for each Kohonen
node
            end
            [maxZ,ind] = max(Zj); % find maximum
    end
```

```
    Zj = zeros(numWin,J); % all or nothing activation of k-winners
    yEst = zeros(numWin,K);
    % Zj(ind) = 1;

    %Zj(kWinners) = 1;
    for i = 1:numWin
        Zj(i,kWinners(i)) = 1;
        yEst(i,:)=Zj(i,:)*Wjk; % calculate estimated value of output/target
    end
    Zk = zeros(numWin,1);
    for ii = 1:numWin
        [Max,ind]=max(yEst(ii,:));
        Zk(ii) = ind;
    end
    wi = (kDist)./nansum(kDist);

    %[maxZ ind] = max(yEst);
    predictOutput(n,:)= round(nansum(wi.*Zk)); % store prediction
    %predictOutput(n,:) = ind;

    switch normMode % Normalize prediction for feedback into recurrent nodes
        case 2 % L2 normalization
            normPredictOutput(n,:) = (predictOutput(n,:)-
min_target)/(max_target-min_target);
    end

end
save(fname,'Wij','Wjk','trainLoc');
interp = predictOutput;
end % function end



function [] = evalANN
(predicted,observed,numDate,rainData,varName,titleName,numRecNodes)

predicted = [observed(1:numRecNodes); predicted];
figure()
subplot(3,1,2:3)
h = plot(numDate,predicted,numDate,observed);
ylabel(varName);
legend('Estimated','Observed','Location','Northwest')
title(titleName);
set(h(1),'LineWidth',1.5,'LineStyle','--','Color','m')
set(h(2),'LineWidth',1.0,'Color','k')
datetick(gca,'x','mm/dd','keepticks')
subplot(3,1,1)
bar(numDate,rainData);
set(gca,'Ydir','reverse');
ylabel('15-Minute Rainfall (in)');
datetick(gca,'x','mm/dd','keepticks')

figure()
loglog(observed,predicted,'.');
```

```
xlabel('Observed');
ylabel('Predicted');


RMSE = rmse(predicted,observed)
Corr= corr(observed,predicted)
Rsquare = rsquare(observed,predicted)
NSE = nashsutcliffe([numDate observed],[numDate predicted])

end
```