Sudarshan S. Chawathe’s Publications

2019-12-09

Notes:
- A PDF version suitable for offline viewing or printing: pubs.pdf.
- Bibliographic files for your citation pleasure:
  - BibTeX: pubs.bib
  - RIS: pubs.ris
  - EndNote: pubs.end
- Validation checks:
- Older list with more details of earlier publications: http://chaw.eip10.org/opubs/.
- Missing abstracts, keywords, and other details to be added soon.


  This paper provides a method for automatically classifying diseases in rice plants by analyzing photographs of rice leaves. The method uses image processing algorithms to detect leaves and likely disease-induced lesions in the leaves. Next, several attributes are computed based on the dimensions of leaves and lesions, the numbers and shapes of lesions, as well the color characteristics of lesions and intact portions of leaves. These attributes are used to build classification models using well established algorithms. The method is evaluated using a publicly available database of rice leaf images.

  ⊳ Rice Disease; Rice Leaf; Image Processing; Classification; Machine Learning.


  This paper explores the application of computational methods to the analysis of the large and growing corpus of news articles and related data on climate change. Topics are analyzed using Latent Dirichlet Allocation and methods customized to specific news sources that take advantage of keywords and other metadata that may be present. Results of this method on news articles drawn over several months are presented.

  ⊳ Climate Change; News; Topic Modeling; Machine Learning


  Citizen science efforts often include data collection by volunteers. Computerizing such data collection provides several benefits, including improved data consistency, shorter time from collection to use, and immediate feedback to the data collectors. Implementing such a computerized data collection system is often challenging because it is difficult to accurately
estimate the level of participation and, therefore, the required load-handling capacity. Overestimating the capacity results in unnecessary infrastructure costs while underestimating it leads to sluggish or failed systems. The so-called serverless or cloud based systems are attractive in this context because they permit the apparent (paid) infrastructure to scale with load. Determining cost profiles of different designs in this environment and, therefore, selecting a suitable one are challenging tasks that are addressed by this work.


  Cloud-hosted NoSQL database services, such as AWS DynamoDB, offer significant advantages, including low start-up costs, high performance and availability, wide scalability, and ease of deployment and management. These advantages have led to their rapid adoption and growth. However, the data storage, querying, and modification features supported by such NoSQL services are very rudimentary in comparison with those of relational and object database systems. Further, data modeling decisions made to map application requirements to the supported NoSQL model have very significant impact on not only performance but also financial cost incurred in using the services. Unlike the well developed body of work for relational- and object- database design, there is a great dearth of systematic procedures for NoSQL database design. This paper addresses this design problem by providing methods that map standard data models to the typically idiosyncratic and rudimentary models supported by NoSQL database services, using AWS DynamoDB as a specific instance.

  ▶ Cloud Computing; NoSQL; Databases; Data Modeling.


  This paper describes methods that use historical data on the rates of occupancy of car parking facilities to predict future occupancy rates. The methods are evaluated using a publicly available dataset of car park occupancy rates. The results suggest that a usable level of prediction accuracy may be achieved using only a modest amount of data that is easy to gather using current technologies.

  ▶ Intelligent Transportation Systems; Smart Cities; Car Parking; Regression; Machine Learning.


  The Trusted Remote Function Interface (TRFI) is a small library that exposes services via a REST API to allow function execution with scientific programming languages. Functional units are uploaded to a remote server using the provided REST API. The API stores registered functions for later execution. Maintaining code using this technique allows clients to repeatedly execute functions without having the native language, typically Octave or Python, installed on the client machine. A common problem in scientific applications is the requirement for a program to interface with scientific scripting languages. Typically, this is not a straightforward approach for accomplishing the data exchange and subsequent function execution on that data from popular languages such as Java or JavaScript. This task is extraordinarily cumbersome if the interpreter, used by the scientific programming language, is not installed locally. By separating the function signature from the underlying implementation, and providing a uniform REST API, the TRFI library allows function interfacing in two ways. First, direct interfacing by using the equipped Java library. Second, the more common scenario is interfacing remotely by deploying the library using a JAX-RS compatible web server. The result of the TRFI library’s design and the provided REST API is the facilitation of code interoperability and reuse for scientific applications.
We present an approach for scientific data management systems to apply certificates to scientific objects, which are typically unformatted datasets, to facilitate analysis by climate scientists. For a program to process data, the program requires cleaned data in a form that supports automatic manipulation. Most systems require that data must adhere to a specific format to achieve that goal. The technique described in this paper takes the opposite approach; instead, any dataset may be imported and manipulated in the system. But upon initial import, however, only a subset of system functions may work with any given dataset. As the data is refined and transformed by system functions, more functions may become compatible. Certificates are associated with objects that pass constraint validation within the system to ensure that they conform to function requirements. The attached object constraints represent invariant properties of the object, which may be used by functions in the system as function preconditions. Furthermore, the functions defined in the system may associate certificates with the newly generated results. Certificates related to function results are effectively function postconditions, which in turn are used to associate certificates with the objects generated in the system. Additionally, attached object certificates reflect the refinement of data into a more pristine version. This paper describes the technique for modeling and enforcing the constraints for data scientists that have similar requirements.

Indoor localization using radio-frequency signals in the 2.4 GHz band is attractive due to availability of low-cost commodity WiFi hardware. However, using such signals for localization is challenging due to signal-propagation complexities such as multipath, fading, and shadowing. This paper describes a method for classifying indoor locations using frequency-domain signatures of RF signals. The method is evaluated using a publicly available dataset of detailed signal measurements in a real environment.

DynamoDB is a popular NoSQL database service that permits queries in a restrictive but useful query language. The metered costs (which translate to financial costs) of executing such queries are measured in units of provisioned capacity or number of requests. Costs of equivalent queries may differ by orders of magnitude but the onus of choosing a low-cost equivalent query is on the service’s client and must be performed by query rewriting. This paper formulates this query-rewriting problem for DynamoDB and outlines methods for choosing low-cost equivalent queries.

Low-cost and commodity off-the-shelf surface electromyographs (sEMGs) may be used for unobtrusive detection of human hand gestures. Although these EMG signals are not as detailed as conventional ones, an experimental investigation of feature engineering and classification reveals that they can yield accurate hand gesture information.

- **Hand gesture detection; surface electromyograph; sEMG; EMG; COTS; sensors; classification; machine learning**


  Modern ultrasonic flowmeters provide routine diagnostic information that may be used to infer their health. This inference task is modeled as a classification problem and studied experimentally using a publicly available dataset. A few classifiers, such as Bayesian Networks, provide good accuracy and also suggest relationships among the diagnostic variables.

- **Ultrasonic flowmeter; diagnostics; classification; machine learning**


  The study of topics that frame the discourse of climate change in news and social media is useful for understanding media and public perceptions of the field and its recent developments. Computational methods for topic modeling, syntactic analysis, and guided data exploration may be applied to readily available big-data streams to extract topics and related information in near-real time.


  We present the software integration of ice core dating tools to the Climate Data Workbench (P301 system). The implementation allows researchers to use different annual indicators in ice core time series in order to develop and apply time scales. During the creation of the time scale, an interpolated, dated version of the actively investigated core is presented to the researcher in real-time.


  Condition-based maintenance (CBM) of hydraulic systems requires methods for condition monitoring: Sensors installed in a hydraulic system for this purpose generate streams of real-time data that must be analyzed to accurately characterize the health of the system. Prior work has developed an experimental hydraulic system with such an installation and yielded a public dataset of sensor readings with associated values of condition variables that quantify the system’s health. This paper presents classification-based methods for inferring these condition variables from the sensor data streams. These methods significantly improve on the classification accuracy reported in prior work on this data. Further, this accuracy is maintained even when the number of sensor-based attributes used as input is substantially reduced.

- **Condition monitoring, condition-based maintenance, hydraulic systems, sensors, classification**

Detecting falls and other mishaps using data from sensors worn by individuals is an important task with applications in healthcare. A related task is using such sensor data to detect routine activities of daily living. This paper models such detection of falls and routine activities as a classification problem. Using a publicly available dataset of real accelerometer traces generated by participants performing intentional falls and other activities, the efficacy and performance of several classifiers are studied experimentally.

- fall detection, activities of daily living, accelerometers, sensors, classification


  Blockchain datasets, such as those generated by popular cryptocurrencies Bitcoin, Ethereum, and others, are intriguing examples of big data. Analysis of these datasets has diverse applications, such as detecting fraud and illegal transactions, characterizing major services, identifying financial hotspots, and characterizing usage and performance characteristics of large peer-to-peer consensus-based systems. Unsupervised learning methods in general, and clustering methods in particular, hold the potential to discover unanticipated patterns leading to valuable insights. However, the volume, velocity, and variety of blockchain data, as well as the difficulties in evaluating results, pose significant challenges to the efficient and effective application of clustering methods to blockchain data. Nevertheless, recent and ongoing work has adapted classic methods, as well as developed new methods tailored to the characteristics of such data. This chapter motivates the study of clustering methods for blockchain data, and introduces the key blockchain concepts from a data-centric perspective. It presents different models and methods used for clustering blockchain data, and describes the challenges and some solutions to the problem of evaluating such methods.


  We present a small library for maintaining the provenance of objects in a software model called The Tiny Java Library for Maintaining Model Provenance (TJLP). A unique characteristic of the library is that it may be applied to existing software models with minimal modification. The library allows the software developer to introduce the ability to move back (undo) and forward (redo) through an object’s instance history with minimal code modification. The requirement is that the model implements the Model interface. Finally, methods that are considered critical in the object’s provenance are adorned with an Undoable annotation. The code necessary to maintain the object’s history is automatically inserted into the critical, undoable-method bytecode when the class definition is loaded by an extended class loader. The states of the model objects are preserved both in memory and on disk to accommodate various computer system configurations. The library performs well for small to medium size models using the default settings, but it may be customized in order to perform better with larger models, especially if the model size approaches the RAM of the underlying computer system.


  The study of past climate enables a better understanding of present and future climate conditions. However, directly measured data for temperature and other climate variables is available for only the recent past (a few hundred years). Study of climate in the more distant
past, from centuries to millennia before present, requires the use of indirect methods which use other variables as proxies. Chief among such methods is the use of data derived from ice cores. Analyzing such ice-core data in order to gain insights into past climate is a complex task that requires data from diverse sources to be combined, transformed, and visualized in multiple and often novel ways. In the past, such analysis was often performed using an ad hoc collection of software tools, such as spreadsheets and plotting programs. There are two primary reasons why this past approach to analyzing data is no longer effective: First, recent technological advances in the physical and chemical processing of ice cores to extract measurements have resulted in orders-of-magnitude increase in the volume of data. Not only does this volume of data render some software tools inoperable but also it makes it difficult for a human to interpret data visually. Second, and more important, ad hoc application of multiple tools to analyze data, even when it produces usable results, typically leaves no systematic record of the precise sequence of transformations that yield a data product, such as a chart of temperature over time, from the original data sources. The P301 project addresses these shortcomings of prior data analysis methods by providing an interactive, graphical software workbench with a few notable features in this context: First, it can analyze even the largest ice-core datasets available today, and more, in interactive times (a few seconds at most). Second, it permits a scientist to interactively use, define, and compose software tools for analyzing data in diverse and powerful ways. Third, all transformations of both tools and data are automatically recorded by the system in a manner that permits examination, study, transformation, and workflow management.


  The Internet of Things (IoT) has rapidly transitioned from a novelty to a common, and often critical, part of residential, business, and industrial environments. Security vulnerabilities and exploits in the IoT realm have been well documented. In many cases, improving the security of an IoT device by hardening its software is not a realistic option, especially in the cost-sensitive consumer market or in legacy-bound industrial settings. As part of a multifaceted defense against botnet activity on the IoT, this paper explores a method based on monitoring the network activity of IoT devices. A notable benefit of this approach is that it does not require any special access to the devices and adapts well to the addition of new devices. The method is evaluated on a publicly available dataset drawn from a real IoT network.


  Optical Burst Switching (OBS) networks provide a practical alternative to optical packet switching and optical circuit switching by separating control information from the primary data, sending the former on a separate control channel. However, this separation also renders OBS networks susceptible to a denial- or degradation-of-service attack (intentional or otherwise) when the data provisioned by a header packet on the control channel does not materialize. This paper addresses the problem of detecting and characterizing such problems and describes a method based on monitoring network traffic on the control and data channels. The method is evaluated on a publicly available dataset.

Persons and devices in indoor environments, such as office buildings, may determine their location using Bluetooth LE beacons, such as iBeacons. Some number of these beacons are distributed over the environment of interest and their identifiers and locations are broadcast widely. The vector of received signal strengths from all these beacons may be intuitively expected to correlate well with location in the physical environment. However, the complexities of Bluetooth signal propagation in environments with obstructions and channels (walls, furniture, ducts, etc.) make it difficult to compute locations in this manner from only the signal values and known locations of beacons. Instead, a data-driven approach that uses a training set composed of observed signal strength vectors at known locations is more effective. This paper studies such methods using a publicly available dataset obtained by collecting training data in an academic building.


The classification of functioning and disabilities in children and youths is an important task that informs healthcare. The ICF-CY (International Classification of Functioning, Disability, and Health in Children and Youth) provides a standard framework for such classification. Occupational therapists use the ICF-CY in conjunction with observations of the routine activities performed by a child (such as eating, toileting, washing) to determine a suitable diagnostic group. This paper presents a method for assisting occupational therapists and others in this task using machine learning. The method is studied experimentally using a publicly available dataset of self-care activities.


Scientific Java programs often need to interact with specialized programming environments, such as Octave and Matlab, that focus on numerical computations. This paper presents the HDFJavaIO library that allows Java programs to interact with Octave using Hierarchical Data Format 5 (HDF5) files, which are commonly used in the scientific community for working with large data sets. Because features of HDF5 files include almost all of the features of NetCDF, this library and method may also be used to create data files that can be used with NCL scripts and other applications that use these large-data formats without the need for further modifications by Java application developers. This paper presents the relevant details of the Octave HDF5 file format and the Java techniques used to build the data interchange library. It also presents the results of an experimental analysis of the library’s performance and its comparison to existing approaches.


Activities of Daily Living (ADLs), or a person’s routine activities of self-care, are important factors influencing the feasibility of home health care or aging in place for many individuals. Automated, sensor-based recognition of such activities affords home stay, greater independence and privacy, and improved quality of life to individuals who would require stay in a
supervised or medical facility. This paper describes a data-driven framework for the design and deployment of such an automated system for activity recognition using simple, unobtrusive, and privacy-friendly binary sensors. It presents the results of an experimental study, with both numerical and qualitative observations, of this framework on a publicly available real dataset.


  Recent technological advances enable the gathering of extensive data on vehicular trajectories of large numbers of travelers at an unprecedented level of detail. Such trajectory datasets provide a wealth of information for purposes such as urban planning, carpool formation, and public-transportation design. This paper describes methods for analyzing and visualizing such data with an emphasis on sparse-traffic environments. It outlines the needs of applications in this domain and presents methods for clustering trajectories and for visualizing the results. The methods are evaluated by an experimental study on a publicly available dataset from real travelers.


  Blockchains such as those used by the Bitcoin and Ethereum cryptocurrencies provide a global, observable record of all transactions and associated data. Analyzing blockchain data is useful for tasks such as detecting fraudulent activities, studying the use and growth of the system, and understanding its levels of anonymity and traceability. Such analysis is challenging due to the high volume and rapidly changing characteristics of popular blockchains. In particular, online (soft real-time) analysis of blockchains requires methods that adapt organically to changes in the data. This paper describes such a method based on self-organizing maps and reports on experiments using the Bitcoin blockchain data.


  Phishing and other malicious email messages are increasingly serious security threats. An important tool for countering such email threats is the automated or semiautomated detection of malicious email. This paper reports work on using fuzzy rules to classify email for such purposes. The effectiveness of a fuzzy rule-based classifier is studied experimentally on a real dataset and compared with results for other classifiers, including those based on crisp rules and decision trees. The human-readability and editability of the classifiers produced by these methods is also studied.


  We study the large-scale soft-realtime distributed collection, analysis, and reporting of data, emphasizing low-cost, low-overhead solutions that scale gracefully as usage varies over several orders of magnitude.


  We present a lightweight Java library that simplifies maintenance of the provenance of software object models. The implementation is based on annotations that are interpreted by an extended class loader to inject the Java bytecode to enable model maintenance.

Ice core archives provide the most direct and detailed evidence of past climate and atmospheric conditions. However, the resolution of traditional ice core sampling methods limits the scope of information that can be extracted from the ice regarding meteorological events (e.g., dust storms, volcanic eruptions, anthropogenic emissions) that are captured at interannual to sub-annual scales. Using laser ablation inductively coupled mass spectrometry (LA-ICP-MS), a novel ultra-high-resolution multi-element sampling method for ice cores, we recovered the highest-resolution continuous glacio-chemical record yet from an ice core, measuring close to 5 million samples from 40 meters of core. This unique record was compiled using samples from the 2013 Colle Gnifetti ice core, located in the Swiss-Italian Alps. Here we present the first results from a new approach to high-resolution ice core data analysis through a new array of statistical tools, data processing algorithms and statistical machine learning tools adapted for ice core data sets. Our new data processing framework is designed to detect, extract and synthesize environmental signals from ultra-high-resolution glacio-chemical time series in concert with more traditional ice core sampling data to further refine paleoenvironmental signals. The authors gratefully acknowledge the Climate Change Institute at the University of Maine, funding from grant AC3862 of the Arcadia Fund and NSF grant PLR-1443306.


This project is a Java library for representing measurement units that provides easier avoidance and detection of a significant source of errors in scientific code. The technique uses the Java virtual-machine's class-loading extensions and annotations with run-time retention policies to enforce units conformance and conversion at run time. Analysis of the Java bytecode is performed at run time (or possibly compile time) to check conformance and conversion of unit-annotated types.


Text segmentation refers to the task of partitioning text into disjoint segments based on some matching and optimization criteria. Examples include partitioning text into words, graphemes, and phonemes. The problem is especially challenging when the language does not require spaces, punctuation, or other simple separators; when segments may be combined in nontrivial ways; and in the presence of errors in transcription or recognition. This paper focuses on a purely lexical method of segmentation: Text is segmented using only a dictionary of known words along with a compatible cost function. No grammatical or other higher-level knowledge is used. The method uses efficient algorithms for multiple-string matching, such as the classic Aho-Corasick algorithm, to yield significant improvements in running time when compared with a simpler dynamic programming algorithm. An experimental study compares the running times of the dictionary-based and dynamic programming algorithms.


Text segmentation refers to the task of partitioning text into disjoint segments based on some matching and optimization criteria. Examples include partitioning text into words,
graphemes, and phonemes. The problem is especially challenging when the language does not require spaces, punctuation, or other simple separators; when segments may be combined in nontrivial ways; and in the presence of errors in transcription or recognition. This paper focuses on a purely lexical method of segmentation: Text is segmented using only a dictionary of known words along with a compatible cost function. No grammatical or other higher-level knowledge is used. The method uses efficient algorithms for multiple-string matching, such as the classic Aho-Corasick algorithm, to yield significant improvements in running time when compared with a simpler dynamic programming algorithm. An experimental study compares the running times of the dictionary-based and dynamic programming algorithms.


  We describe a method for representing measurement units for the Climate Data Workbench, providing easier avoidance and detection of a significant source of errors in scientific code. Our method uses the Java virtual-machine’s class-loading extensions, and annotations with runtime retention policies, to enforce units conformance and conversion at runtime.


  We describe fGMT, a functional interface to the very popular GMT collection of mapping and plotting tools. Our implementation uses scsh Scheme and is designed to permit incremental building of higher-level interfaces that incorporate domain-specific knowledge.


  The P301 Web API is a RESTful interface that allows P301 users to share data that have been uploaded to the P301 system. The system supports accessing data in JavaScript Object Notation (JSON) and Extensible Markup Language (XML) formats, which helps to facilitate the development of Web-based applications. A variety of queries for accessing the data in the system allows for flexibility in client system designs.

- **Toward a Domain-Specific Language for Patterns in Ice-Core Data.** Sudarshan S. Chawathe. In Proceedings of the Borns Symposium, April 2016.

  We describe a language for expressing simple patterns in time series data derived from ice-cores and similar sources. Such patterns use simpler features mapped to tokens by an earlier phase of analysis. In turn, they allow more complex features to be expressed and analyzed.

- **Interactive Exploration of Time Lines from Ice Core Data Sets.** Sudarshan S. Chawathe. In Proceedings of the Borns Symposium, April 2015.

  Time lines are derived from ice core data typically by counting layers or peaks in sequences of measured values. This work (in progress) explores the extent to which automation and interactive exploration may assist this task.


  This paper describes the design, implementation, and deployment of an application server whose primary infrastructure is an elastic cloud of servers. The design is based on the Representational State Transfer (REST) style, which provides significant benefits in a cloud environment. The paper also addresses implementation issues within a specific cloud service and highlights key decisions and their effect on scalability and cost. Finally, it describes our
experiences in deploying a widely used platform with both Web and mobile client interfaces and its ability to cope with load spikes while maintaining a low quiescent cost.


  An important method used to speed up forensic file-system analysis is white-listing of files: Well-known files are detected using signatures (message digests) or similar methods, and omitted from further analysis initially, in order to better focus the initial analysis on files likely to be more important. Typical examples of such well-known files include files used by operating systems, popular applications, and software libraries. This paper presents methods for improving the effectiveness and efficiency of such signature-based white-listing during file-system forensics. One concern for effectiveness is the resilience of the white-listing method to an adversary who has complete knowledge of the method and who may make small, inconsequential changes to a large number of well-known files on a target file-system in order to overload the analysis and thereby practically defeat it. Another concern is the ability to detect near-matches in addition to exact matches. Efficiency refers to primarily the rate at which a target file system may be processed during analysis; preparation-time, or indexing, efficiency is a lesser concern as that computation may be performed during non-critical times. Our work builds on techniques such as locality-sensitive hashing to yield an effective filter for further analysis tools.


  The integration and analysis of data sets from diverse sources provides scientists with an opportunity to gain insights that are not apparent from the individual data sets or sources. For many sources, improving technology and other factors have resulted in a very rapid growth in both the volume and the diversity of data. This wealth of data has the potential for significant scientific breakthroughs. However, this potential is difficult to realize unless there is a systematic and effective method for managing this data. The methods used by researchers in the past typically do not scale up to current and anticipated levels of data volume and diversity. The P301 project addresses this problem with the goal of accelerating the data flow from data sources to research results. Below, we outline one aspect of this work: Managing the syntactic and semantic consistency of data using an interactive framework that eases the task of importing, cleaning, analyzing, and visualizing data, and of recording such data transformations and results using histories and certificates.


  The 10Green Web application integrates air quality data from diverse sources and provides an intuitive interface that summarizes this information in a manner accessible to scientists and non-scientists alike. From a Computer Science perspective, this application presents interesting challenges in both the back end (e.g., data integration and analysis, maintainability) and the front end (e.g., Web-based visualization, interactive response times, and portability across very diverse client architectures). Here, we focus on scalability and outline the implementation aspects that allow the application to scale from a few hundred users to hundreds of thousands of concurrent users at low cost.

The REST Framework for Dynamic Client Environments (RFDE) is a method for building RESTful Web applications that fully exploit the diverse and rich feature-sets of modern client environments while retaining functionality in the absence of these features. For instance, we describe how an application may use a modern JavaScript library to enhance interactivity and end-user experience while also maintaining usability when the library is unavailable to the client (perhaps due to incompatible software). These methods form a framework that we have developed as part of our work on a Web application for presenting large volumes of scientific datasets to nonspecialists.


  We describe the design and implementation of a method to serve hundreds of terabytes of image data (tiles) for a Web-based mapping service. The method allows the service to scale gracefully from a few dozen to thousands of concurrent connections. Map tiles are stored in implicit form in a database and the corresponding bit-mapped images are computed as needed using an efficient stored-procedure implementation. The implementation is also particularly well suited to deployment in the cloud computing environment.


  The Project 301 Data Explorer, P301dx, is a software workbench for climate-change data. It aids scientists with the tasks of storing, integrating, sharing, analyzing, and visualizing such data. The primary goal of Project 301 is improving the efficiency and effectiveness of the process of transforming raw data into easily interpretable scientific results.

- **Information Systems for Passenger Guidance in Transit Systems,** Sudarshan S. Chawathe. Invited presentation at the Symposium on Engineering and Technologies for the Metro Bogota (Metrosimposio), May 2010


- **Effective Whitelisting for Filesystem Forensics.** Sudarshan S. Chawathe. In Proceedings of the 7th IEEE Intelligence and Security Informatics Conference (ISI), Richardson, Texas, June 2009


- **Protecting Transportation Infrastructure.** Daniel Zeng, Sudarshan S. Chawathe, Hua Huang, and Fei-Yue Wang. *IEEE Intelligent Systems*, 22(5):8–11, September/October 2007


• **Segment-Based Map Matching.** Sudarshan S. Chawathe. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), pages 1190–1197, Istanbul, Turkey, June 2007


• **Distributing the Cost of Securing a Transportation Infrastructure.** Sudarshan S. Chawathe. *IEEE Intelligent Transportation Systems Society Newsletter*, 8(2):17–21, June 2006. Republished as one of two selected papers from the ISI-2006 conference


• **Differencing Data Streams.** Sudarshan S. Chawathe. In Proceedings of the 9th International Database Engineering and Applications Symposium (IDEAS), pages 273–284, Montreal, Canada, July 2005


• **Data Management in Interimistic Environments.** Abheek Anand and Sudarshan S. Chawathe. In Proceedings of the Third Workshop on E-Business (WeB), Washington, D.C., December 2004


• **Control of Personal Location Data.** Sudarshan S. Chawathe. In Proceedings of the Location Privacy Workshop, Schoodic Peninsula, Acadia National Park, Maine, August 2004

• **Managing RFID Data.** Sudarshan S. Chawathe, Venkat Krishnamurthy, Sridhar Ramachandran, and Sanjay Sarma. In Proceedings of the 30th International Conference on Very Large Data Bases (VLDB), pages 1189–1195, Toronto, Canada, August 2004
• **Privacy-Preserving Inter-Database Operations.** Gang Liang and Sudarshan S. Chawathe. In Proceedings of the Symposium on Intelligence and Security Informatics (ISI), volume 3073 of *Lecture Notes in Computer Science (LNCS)*, pages 66–82, Tucson, Arizona, June 2004

• **Skipping Streams with XHints.** Akhil Gupta and Sudarshan S. Chawathe. Technical Report CS-TR-4566, Computer Science Department, University of Maryland, College Park, Maryland, February 2004

• **Privacy-Preserving Inter-Database Operations.** Gang Liang and Sudarshan S. Chawathe. Technical Report CS-TR-4564 (UMIACS-TR-2004-09), University of Maryland, College Park, February 2004

• **Efficient Peer-to-Peer Namespace Searches.** Vijay Gopalakrishnan, Bobby Bhattacharjee, Sudarshan S. Chawathe, and Pete Keleher. Technical Report CS-TR-4568, University of Maryland, College Park, Maryland, February 2004

• **Cooperative Data Dissemination in a Serverless Environment.** Abheek Anand and Sudarshan S. Chawathe. Technical Report CS-TR-4562, Computer Science Department, University of Maryland, College Park, Maryland, January 2004

• **XPaSS: A Multiple-Query Streaming XPath Query Engine.** Feng Peng and Sudarshan S. Chawathe. Technical Report CS-TR-4565, Computer Science Department, University of Maryland, College Park, Maryland, January 2004

• **Streaming XPath Subquery Evaluation.** Feng Peng and Sudarshan S. Chawathe. Technical Report CS-TR-4560, Computer Science Department, University of Maryland, College Park, Maryland, January 2004

• **Optimal Buffering for Streaming XPath Evaluation.** Feng Peng and Sudarshan S. Chawathe. Technical Report CS-TR-4561, Computer Science Department, University of Maryland, College Park, Maryland, January 2004

• **Semistructured Data in Relational Databases.** Sudarshan S. Chawathe, chapter 25, pages 1–19. Practical Handbook of Internet Computing. CRC Press, 2004

• **XSQ: A Streaming XPath Engine.** Feng Peng and Sudarshan S. Chawathe. Technical Report CS-TR-4493, Department of Computer Science, University of Maryland, May 2003

• **XPath Queries on Streaming Data.** Feng Peng and Sudarshan S. Chawathe. In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), pages 431–442, San Diego, California, June 2003


• **SEuS: Structure Extraction using Summaries.** Shayan Ghazizadeh and Sudarshan S. Chawathe. In Steffen Lange, Ken Satoh, and Carl H. Smith, editors, Proceedings of the 5th International Conference on Discovery Science, volume 2534 of *Lecture Notes in Computer Science (LNCS)*, pages 71–85, Lubeck, Germany, November 2002
• **Tracking Moving Clutches in Streaming Graphs.** Sudarshan S. Chawathe. Technical Report CS-TR-4376, Computer Science Department, University of Maryland, College Park, Maryland, October 2002

• **XSQ: Streaming XPath Queries.** Feng Peng and Sudarshan S. Chawathe. Technical Report CS-TR-4401 (UMIACS-TR-2002-81), Computer Science Department, University of Maryland, College Park, Maryland, September 2002

• **Discovering Frequent Structures using Summaries.** Shayan Ghazizadeh and Sudarshan Chawathe. Technical report, University of Maryland, Computer Science Department, 2002

• **Discovering Frequent Structures using Summaries.** Shayan Ghazizadeh and Sudarshan S. Chawathe. Technical Report CS-TR-4364, Computer Science Department, University of Maryland, College Park, Maryland, November 2001


• **VQBD: Visualizing, Querying, and Browsing Semistructured Data, Sudarshan S. Chawathe, Thomas Baby, and Jihwang Yeo, November 2000. Extended version of demonstration description.** http://cs.umaine.edu/~chaw/

• **Comparing Hierarchical Data in External Memory.** Sudarshan S. Chawathe. In Proceedings of the International Conference on Very Large Data Bases (VLDB), pages 90–101, Edinburgh, Scotland, September 1999

• **Describing and Manipulating XML Data.** Sudarshan S. Chawathe. *Bulletin of the IEEE Technical Committee on Data Engineering, 22*(3):3–9, September 1999


• **Managing Change in Heterogeneous Autonomous Databases.** Sudarshan S. Chawathe. PhD thesis, Stanford University, 1999


• **An Expressive Model for Comparing Tree-Structured Data.** Sudarshan S. Chawathe and Hector Garcia-Molina. Technical report, Stanford University Database Group, November 1997


• **Representing and Querying Changes in Semistructured Data (Extended Version), S. Chawathe, S. Abiteboul, and J. Widom. Available at http://www-db.stanford.edu, 1997**


