# SWATS: Wireless Sensor Networks for Steamflood and Waterflood Pipeline Monitoring

**SunHee Yoon, Stanford University; Information Sciences Institute, University of Southern California**
**Wei Ye, Information Sciences Institute, University of Southern California**
**John Heidemann, Information Sciences Institute, University of Southern California**
**Brian Littlefield, Chevron Corporation**
**Cyrus Shahabi, University of Southern California**

## Abstract

State-of-the-art anomaly detection systems deployed in oilfields are expensive, not scalable to a large number of sensors, require manual operation, and provide data with a long delay. To overcome these problems, we design a wireless sensor network system that detects, identifies, and localizes major anomalies such as blockage and leakage that arise in steamflood and waterflood pipelines in oilfields. A sensor network consists of small, inexpensive nodes equipped with embedded processors and wireless communication, which enables flexible deployment and close observation of phenomena without human intervention. Our sensor-network-based system, Steamflood and Waterflood Tracking System (SWATS), aims to allow continuous monitoring of the steamflood and waterflood systems with low cost, short delay, and fine-granularity coverage while providing high accuracy and reliability. The anomaly detection and identification is challenging because of the inherent inaccuracy and unreliability of sensors and the transient characteristics of the flows. Moreover, observation by a single node cannot capture the topological effects on the transient characteristics of steam and water fluid to disambiguate similar problems and false alarms. We address these hurdles by utilizing multimodal sensing and multisensor collaboration, and exploiting temporal and spatial patterns of the sensed phenomena.

State-of-the-art anomaly detection systems deployed to monitor pipeline (oil, steam, water, and sewer) networks have major shortcomings. Supervisory Control And Data Acquisition (SCADA) systems [1] for pipeline networks in oilfields, for example, are expensive (equipment and maintenance), not scalable (low density in time and space), inflexible (protocol change and software update), not interoperable (hardware and software), and provide the data or result with long delay. Moreover, field engineers need to control and maintain the equipment manually. Furthermore, because SCADA systems utilize long-range point-to-point communications between the control room and each well, they are less energy-efficient to operate for the long term, and do not support collaboration among wells for in situ automation for monitoring.

The wireless sensor network (WSN) is an attractive technology for applications in extreme environments such as oilfields, which have dangerous chemicals at high pressure and temperature. Furthermore, equipment is located down holes or sometimes under the sea. Production and injection wells can be distant from power, control, and operator. Once a system is deployed, it is difficult to physically access the sensors, so it is desirable to be able to maintain and monitor the systems remotely as much as possible. Such a network of sensors consists of battery powered nodes that collaborate to observe and conclude the health of an oilfield. If we use inexpensive sensors, it becomes economically feasible to deploy a large number of sensor nodes over a large area to cover the entire oilfield, providing much higher spatial and temporal resolution in sensor readings. Figure 1 depicts a conceptual diagram of steamflood monitoring using a WSN.

We have designed a system using WSNs, called the Steamflood and Waterflood Tracking System (SWATS), to detect,

*This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC), IIS-0238560 (PECASE) and the NSF Center for Embedded Networked Sensing (CCR-0120778), and partly funded by the Center of Excellence for Research and Academic Training on Interactive Smart Oilfield Technologies (CiSoft); CiSoft is a joint University of Southern California-Chevron initiative. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors) and do not necessarily reflect the views of any of the above organizations or any person connected with them.*

 0890-8044/11/$25.00 © 2011 IEEE 

| System | Architecture | Storage and control | Flexibility | Cost | Node density | Data rate | Network protocol |
|---|---|---|---|---|---|---|---|
| SCADA | Centralized | Central site | Inflexible | Expensive | Low | Low | Proprietary |
| WSN | Decentralized | Local sensor node | Versatile | Inexpensive | High | High | Non-proprietary |

Table 1. *Comparison between SCADA and WSN systems for oilfield monitoring.*

identify, and localize major problems that arise in steamflood and waterflood pipeline networks in oilfields. Our system aims to allow continuous monitoring of the steamflood and waterflood system with low cost, short delay, and fine-granularity coverage while providing high accuracy and reliability. Our system detects and identifies major anomalies in steamflood pipeline networks: blockage, leakage, outside force damage, generator, and Splitigator malfunction. These anomalies are disambiguated from many false alarms: generator outage, downhole pressure change, phase splitting at piping tees, change in two-phase steam quality, sensor noise and sensor fault, and environmental effects. SWATS also detects and identifies major anomalies in waterflood pipeline networks such as blockage and leakage with minor changes.

Detecting problems in steam and water pipeline networks is challenging because sensors inherently have inaccuracies. Erroneous sensor readings coupled with transient changes in flow rate, temperature, or pressure might trigger false alarms, which makes it challenging to confidently detect a problem in steam and water pipeline networks.

Challenges in identification arise from the complexity in pipeline topology (split, merge, etc.). A single sensor cannot capture the topological effects on the transient characteristics of steam and water fluid to disambiguate similar problems and false alarms. Low energy, processing, and storage availability in sensor nodes create further constraints in the design of our system. Designing intelligent collaboration algorithms is challenging with conflicting requirements such as low-end hardware, long lifetime, and accurate results.

We address these challenges by creating a multimodal sensing and multisensor collaboration algorithm that utilizes a decision tree for anomaly identification and localization. We build a decision tree to capture the salient pressure and flow characteristics of each problem and distinguish them from false alarms. Even though we use low-fidelity sensors, we increase accuracy by combining the sensor readings from multiple sensors and exploiting the underlying data correlations. We form clusters of WSNs with energy-efficient short-range multihop communication for wells physically close each other, and deploy an IEEE 802.11 mesh network with a long-range-high-speed communication network among the clusters and the control room.

We make three contributions in this study. First, we propose to use WSNs to monitor oilfields. Steamflood and waterflood pipeline monitoring is a *novel application* for WSNs. Second, SWATS introduces the *first domain-specific correlation-based decision tree* algorithm to automate detection, identification, and localization problems in the steamflood and waterflood pipeline. Third, SWATS improves on state-of-the-art steamflood and waterflood pipeline monitoring. By using a WSN, it enables dense and continuous steamflood and waterflood pipeline monitoring cost effectively.

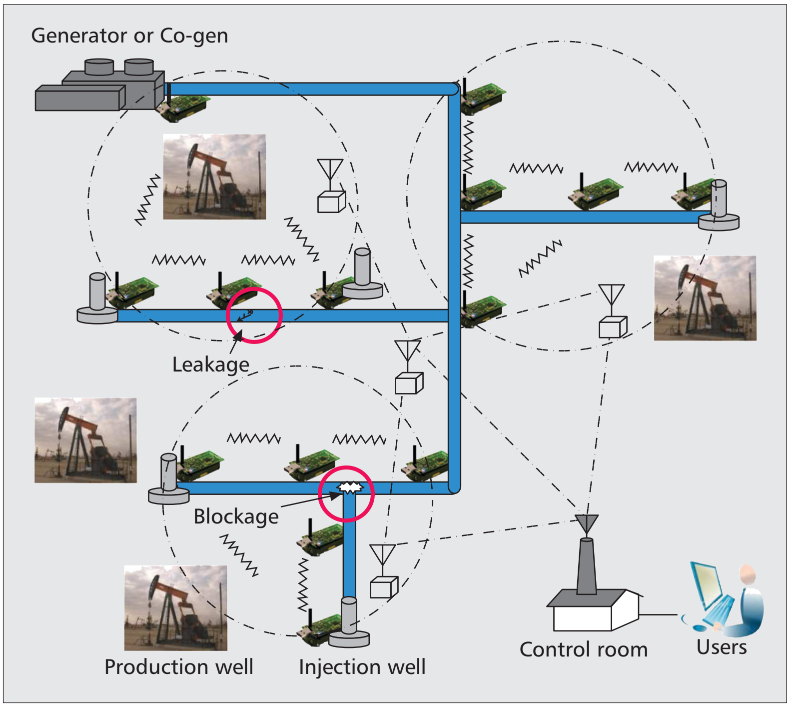


Figure 1. *A conceptual diagram of a steam distribution system and monitoring it using a WSN.*

## Related Work

We classify some studies related to SWATS in two dimensions: applications and techniques.

### Applications

There are two kinds of monitoring applications closely related to SWATS: pipeline monitoring and target tracking.

*Pipeline Monitoring* — Pipeline monitoring is widely used in industry applications to monitor pipelines conveying water [2], oil [3], multiphase gas [4], and two-phase steam [5]. Most of these pipeline monitoring systems, however, are expensive, manually maintained, only a little scalable, inflexible, and not interoperable, and provide data or results with long delays. Kim *et al*. [6] proposed a household water usage monitoring system by measuring the flow on each water outlet in a household using vibration sensors. Like Pipenet [2], SWATS is an emerging application using WSNs, which have complementary characteristics with the above systems. SWATS monitors steamflood and waterflood pipelines in oilfields cost effectively.

*Target Tracking* — SWATS, although similar, is fundamentally different from target tracking applications [7, 8] in sensor networks. SWATS attempts to localize the static location of a problem, while target tracking localizes the position of a moving object. The problems in an oilfield are confined within a pipeline, while an object being tracked in a tracking application might move in an undetermined and open path. In

SWATS cross-check only across the neighboring sensors along the trajectory of the fluid is required to validate a reading, while in target tracking such comparison and correlation is done across all the neighboring nodes. Many explicit rules are used to identify and classify problems and many types of false alarms in SWATS, while very few, if any, rules are used in target tracking. SWATS performs in situ sensing, while target monitoring performs remote sensing.

### Techniques

There are three main techniques related to SWATS in the literature: SCADA, collaborative fusion, and decision tree.

*SCADA Systems* — SCADA [1] is a computer system for gathering and analyzing real-time data, which usually consists of remote telemetry and sensors, controllers, networks, a data server, and a user interface. SCADA systems are used to monitor and control a plant or equipment in industries such as water and waste control, energy, oil and gas refining, and transportation. This system is neither flexible nor interoperable, and is expensive to deploy and maintain. Table 1 compares the differences between SCADA and WSN systems.

Liou [3] proposed a software-based pipeline leakage detection system for crude oil and refined petroleum using the SCADA system. Unlike SWATS, this system only detected pipeline leakage. Erickson and Twaite [4] developed a Pipeline Integrity Monitoring System (PIMS) that helps detect pipeline leakage and track the gas composition of the wet gas pipelines. However, PIMS only considers the single problem of pipeline leakage.

There are new approaches to replace expensive SCADA systems. Stoianov *et al*. proposed Pipenet [2], a WSN-based prototype pipeline monitoring system deployed by the Boston Water and Sewer Commission (BWSC). Three online monitoring applications (hydraulic and water quality monitoring, remote acoustic leak detection, and monitoring combined sewer outflows) feature high sampling rate, fine-time synchronization, and complicated signal processing. Although Pipenet and SWATS detect anomalies based on the correlations in sensor readings, Pipenet did not provide an in-network processing algorithm such as SWATS that reasons about the sensor data and makes decision at *local* sensor nodes through collaboration. Thus, existing systems are limited to data collection or are specifically designed to address a single class of problem.

*Collaborative Fusion* — Collaborative fusion is the process of combining (fusion) and evaluating information obtained from multiple heterogeneous sensors into a single composite picture of the environment.

Gu *et al*. [7] built a distributed surveillance application satisfying requirements of low-end hardware, long lifetime, and processing sophisticated functions such as signal processing and classification functions. Liu *et al*. [8] proposed a distributed and dynamic group management method for multiple target tracking. Both approaches utilize collaborative sensor fusion and conserve energy by trying to maximize local computation and minimize communication. They have simple problem sets (a few different objects or multiples of the same object) to be classified. Unlike SWATS, these approaches did not use the decision tree algorithm to detect, classify, and localize the tracked objects.

*Decision Tree* — The decision tree takes as input a set of properties describing an object or a situation, and outputs a yes/no decision (Boolean outcome) or classification tree (discrete outcome) or regression tree (continuous outcome) [9]. Implementation of a decision tree is simple and computationally efficient, which makes it appropriate for the complicated online diagnosis applications in WSNs. However, understanding the domain knowledge is crucial to build a decision tree.

Ramanathan *et al*. [10] designed a debugging tool called Sympathy that detects, classifies, and localizes the sensor network failures. Sympathy uses the empirical decision tree to determine the most likely cause of packet loss in the network, while SWATS uses a theoretical decision tree based on fluid dynamics to determine the anomaly in the steam and water pipelines. Sympathy uses a simple binary decision tree (yes/no decision), while SWATS uses a complicated multidimensional decision tree (classification tree). Zhao *et al*. [11] proposed a prototype diagnostic system that integrates both approaches of model-driven signature analysis and utility-driven sensor queries. However, this system detects and classifies problems occurring in a single node, but does not diagnose problems over WSNs. Thus, they do not explore the localization problem because all problems occur at the designated node in their case studies.

## Motivation

Heat delivery to an oil well is a major cost in the operation of thermally heated oil wells. This cost can be significantly reduced by finer and faster control of heat delivery to the malfunctioning equipment or pipelines. *Steamflooding* is one of the thermally enhanced oil recovery (TEOR) techniques, and utilizes the heat contained in the steam to make heavy oil (< 20° API) more fluid for easier oil recovery [12]. This is an economics-driven problem because the steam generation and distribution uses about half of the total budget for the entire oilfield operation. The goal of steamflooding is to optimize the quantity of steam injected to each injection well so that the amount of heat delivered by the stream pipeline networks is fair and constant.

*Waterflooding* is the dominant method of enhanced oil recovery (EOR) [12]. Water is an effective fluid for maintaining reservoir pressure and driving oil toward a producer for non-heavy oil. The goal of waterflooding is similar to steamflooding; maintaining fair and constant delivery of water to each injection well at the maximum efficient rate for cost reduction.

*Critical flow rate* refers to the flow rate when it reaches a sonic velocity in the throat of an orifice or a choke [5, 13]. A property of critical flow is that the flow rate is dependent only on the upstream conditions and the physical description of the orifice or a choke. Maintaining the critical flow rate is important in steamflood systems because it ensures the delivery of a constant amount of heat to each well. Providing constant delivery of water to each target well is important because the amount and angle of individual injection are designed for maximal production by understanding the correlation among various factors: the level of permeability, geological formation and heterogeneity, angular unconformity, and degree of subsidence and uplift. Oilfield engineers want to *detect* the situation where the actual flow rate is out of target injection in steamfloods and waterflood, *identify* its causes, *localize* its origin, trigger an alarm immediately, and provide feedback to the machines that control steam or water injection to halt steam or water injection until further diagnosis. Figure 2 shows the equipment currently used in the SCADA steamflood monitoring system.

Problems resulting in the out of critical flow rate in steamflooding can be due to blockage, leakage, equipment malfunction in both generator and Splitigator, and outside force or third-party damage (Table 2). Problems resulting in the out of target injection can be leakage, plugging, and equipment malfunction.

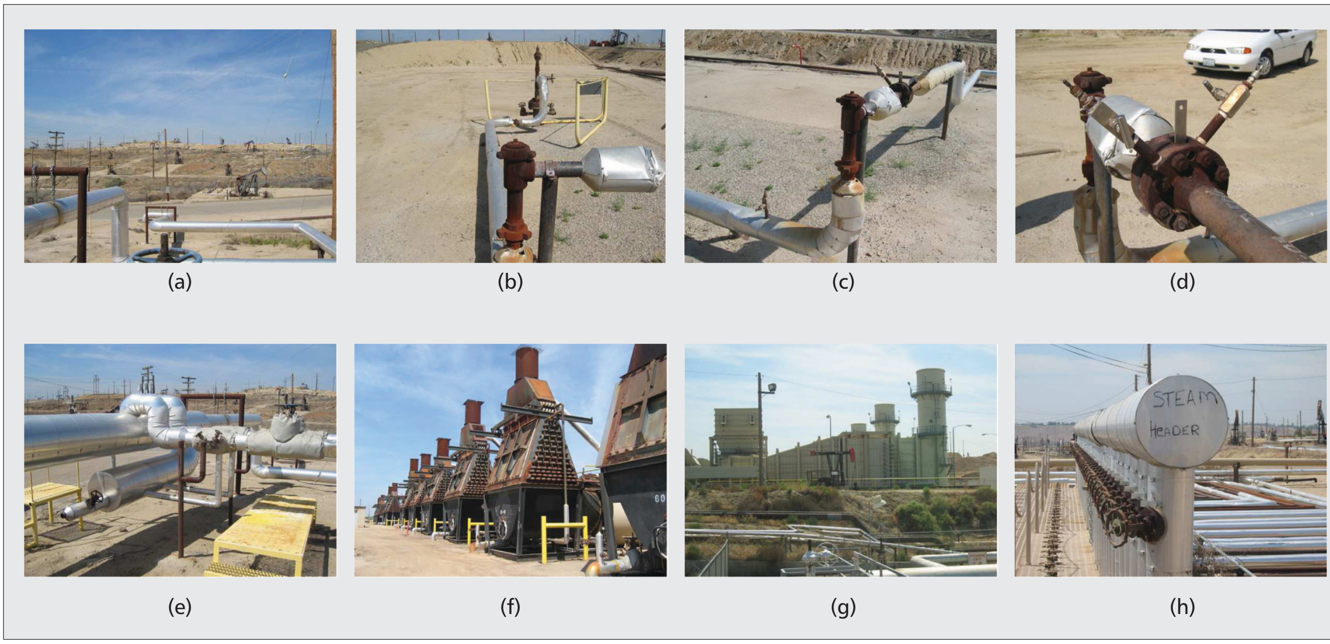


Figure 2. *Equipment currently being used in the SCADA system for steamflood pipelines monitoring in an oilfield: a) an overview of the oilfield in Kern River field, Bakersfield; b) injection well (top part); c) downstream pressure meter (left) and upstream Orifice flow meter (right) of Choke (middle); d) Orifice flowmeter; e) splitigator, which controls steam quality constant between upstream and downstream; f) steam generator; g) co-gen(erator) of steam and electricity; h) steam stimulator.*

Blockage and leakage are the major concerns, which can result in out of critical flow rate of steam. Blockage, which is often observed at the Splitigator and choke, is often caused by scale deposition from saturated steam or leftover debris and foreign objects after construction. Leakage, often observed near pipeline flange and joint, is caused by pipeline corrosion and loose junction. Blockage and leakage are also the primary concerns in waterflood monitoring. Incipient detection of these problems is challenging because at the early stage of problem the pressure and flow rate change is difficult to distinguish from those of normal or transient fluctuations and false alarms. Moreover, in s real environment multiple problems can happen simultaneously in addition to various false alarms, which makes anomaly detection and identification even more challenging. Generator malfunction and Splitigator malfunction are equipment-specific problems for which we strategically deploy multiple sensors near the equipment. Outside force or third party damage happens less frequently than blockage (plugging) and leakage, and they are easy to detect because they show a sudden change in pressure and flow rate.

The anomaly detection system in use in the steam and water pipelines has prohibitive cost, long delays in measurement, and coarse measurements, and requires periodic manual inspection. Field engineers are interested in automating this manual and slow detection and correction process with a system that can detect problems fast, make decisions rapidly, and take actions to fix the problems quickly. Economic considerations dictate that such a system has to cost less than the current manual system and eventually save cost in oilfield operation by detecting and fixing problems in a timely manner. Our goal is to design a system that detects, identifies, and localizes problems reliably, quickly, and accurately while reducing cost.

## Research Challenges

In this section we describe the main research challenges in monitoring of steam and water pipeline networks in oilfields.

### Reliable Detection

Detecting problems in steam pipeline network is challenging because sensors inherently have inaccuracies. Erroneous sensor readings might trigger false alarms, which makes it challenging to confidently detect a problem in a steam pipeline network. Moreover, our measurement of the pressure and flow rate of steam in a pipeline changes even under normal operation because of pipeline friction and differences in pipeline diameters at different places. The transient characteristics of two-phase steam changes the steam quality even without any anomaly.

### Correct Identification

There are several causes of false alarms that make the correct identification of problems challenging (Table 2). The transient physical characteristics of steam and water fluid also make distinguishing problems from false alarms difficult. Due to multiple problems and false alarms, and the complicated steam properties and pipeline topology, the physical phenomena of each anomaly and false alarm can only be distinguished by:

- Comparison over nodes at a certain distance upstream and downstream in the pipeline
- Multimodal sensing and validation at each node

Since we cannot identify problems nor distinguish problems from false alarms using single-node processing, we need to design accurate and efficient collaboration algorithms, which is challenging.

### Timely Localization

Localization is less challenging than detection and identification once the system detects and identifies the problem correctly. One of the benefits of using a decision tree algorithm such as SWATS is simplifying the localization; the origin of the problem is at the best matching node for the rule identifying the problem.

### Efficient Network Protocols

Because anomalies can occur anywhere in the pipeline network in an oilfield, in situ in-network processing capability is needed on each node to support multiwell collaboration for

| Events to detect | | Physical phenomena | Where to deploy sensors | Reasons |
|---|---|---|---|---|
| Problems | Blockage | Blockage | Splitigator (water leg orifice, valve, steam leg orifice), choke | Scale, something left over after construction |
| | Leakage | Leakage | Flange, joint | Pipeline corrosion, pipeline junction loose |
| | Generator malfunction | Leakage or blockage | Generator | Generator breakdown |
| | Splitigator malfunction | Phase splitting | Before and after Splitigator | Splitigator malfunction |
| | Outside-force or third-party damage | Leakage | Flange, junction, near obstacles | Earthquake, stone |
| False alarms | Change in steam supply | Leakage or blockage | Generator | Generator outage, shortage in steam supply |
| | Downhole pressure change | Downhole pressure change | Injector | Vertical permeability, geological formation, heterogeneity |
| | Change in steam quality | Change in steam quality | Inlet of pipeline, before and after Splitigator, obstacles, and pipeline elevation | Due to the elevation or pressure change, or 2-phase transient property of steam |
| | Phase splitting at piping tees | Phase splitting | All branches including Splitigators | Difference in steam quality between upstream and downstream of branches |
| | Sensor noise and sensor fault | Inaccuracy in sensor readings | A pair of nearby sensors in the middle of pipeline for sanity check | Inaccurate and unreliable pressure meter, thermometer, and flow meter |
| | Environmental effects | Inaccuracy in sensor readings | A pair of nearby sensors in the middle of pipeline for sanity check | Environmental noise unique to this system such as pipeline friction, ambient temperature, etc. |

Table 2. *Classification of anomalies in steamflood monitoring in oilfield. All the events need multiple sensors for detection.*

automated monitoring. However, de facto SCADA systems commonly used in oil fields utilize expensive, inefficient, long-range point-to-point communications between the control room and each well, and do not support communication among wells. In order to address these problems, we form clusters of wireless sensors located near wells physically close to each other. All the nodes in a cluster conduct peer-to-peer communications because the problem is ubiquitous, and all the nodes are expected to sense and process data with approximately equal levels of intelligence. With an energy-efficient short-range multihop communication protocol rather than less efficient long-range communication, the system will be able to run on batteries and solar power for years. More important, sensor network enables multi-well interaction and collaboration, so that intelligent sensing and control algorithms such as SWATS can be implemented over multiple wells in an area. A long-range, high-speed communication network such as the IEEE 802.11 mesh network can be used to relay communication from clusters to the control room. Figure 1 illustrates our proposed wireless network architecture.

Designing an energy-efficient medium access control (MAC) protocol with low duty-cycle accompanied by a semantic communication protocol coupled with intelligent processing of SCADA data is critical for optimizing energy efficiency of resource constrained wireless nodes. We can achieve energy-efficient communication by utilizing domain knowledge such as the frequency of each anomaly. A customized sleep schedule accordingly will save the energy spent due to idle listening.

## Reliable Delivery

The importance of a single data from in-network processing is incomparable to raw data. Moreover, an alarm notifying of an anomaly from the processing of SCADA data is time critical. Sometimes, a system designer might have to decide to trade precious energy for guaranteed delivery. To make an inherently unreliable low-power network reliable, the system must be robust against interference from the physical structure in an oilfield such as pipelines and generators, interference from existing point-to-point SCADA communication radios, and interference between low-power wireless networks and 802.11 networks. We need to design a network protocol that delivers data reliably in low-power wireless networks even with concurrent 802.11 mesh networks.

In short, the reasons the above challenges are difficult are:

- Low-cost sensors can be unreliable, inaccurate, and inefficient in its use of limited energy supply.
- False alarms can be mistaken for real anomalies.
- Topological effects of a pipeline must be taken into consideration.
- Transients in steamflood and waterflood must be taken into consideration.

There are other constraints, such as limited energy and processing power on each node, that may complicate system and algorithm design. Evaluating different design trade-offs and parameter selection are also challenging issues. We plan to investigate the trade-off between the centralized and distributed algorithms. Various parameters, such as sampling rate, the duration of the sample window, and the size of a neighboring group should be tuned for correctness, timeliness, and efficiency of the algorithm.

## A Pipeline Monitoring System in an Oilfield Using Wireless Sensor Networks

### Overview of Our Approach

The key technique of our algorithm is the identification of both real problems and false alarms with a decision tree by collaboratively exploiting spatial and temporal correlations in the sensor readings. We define the decision tree by capturing the salient characteristics of the pressure (or temperature) and flow rate in space and time as a consequence of each problem and false alarm.

The intuition behind our approach is that the neighboring sensor nodes in a pipeline should observe a coherent impact for each anomaly on pressure and flow rate in steam and water flow. We assume that inexpensive temperature, pressure, and acoustic flow meters are strategically placed in the pipeline network.

Because of possible inaccuracy in sensor readings, we use multimodal multinode collaboration to improve the correctness of problem diagnosis. Although we may detect the problems and false alarms correctly at a single node, single-node processing is not enough for correct identification of problems and false alarms. Most of the problems and false alarms present the same phenomena in pressure and flow rate in a node such as gradual drop, sudden drop, or ephemeral change. Several problems and false alarms are *only* distinguished by analyzing the physical signature over upstream and downstream nodes, and by comparison with multiple modalities such as pressure and flow rate simultaneously. We create spatial and temporal patterns in our decision tree algorithm by understanding the unique indications of each problem in fluid dynamics.

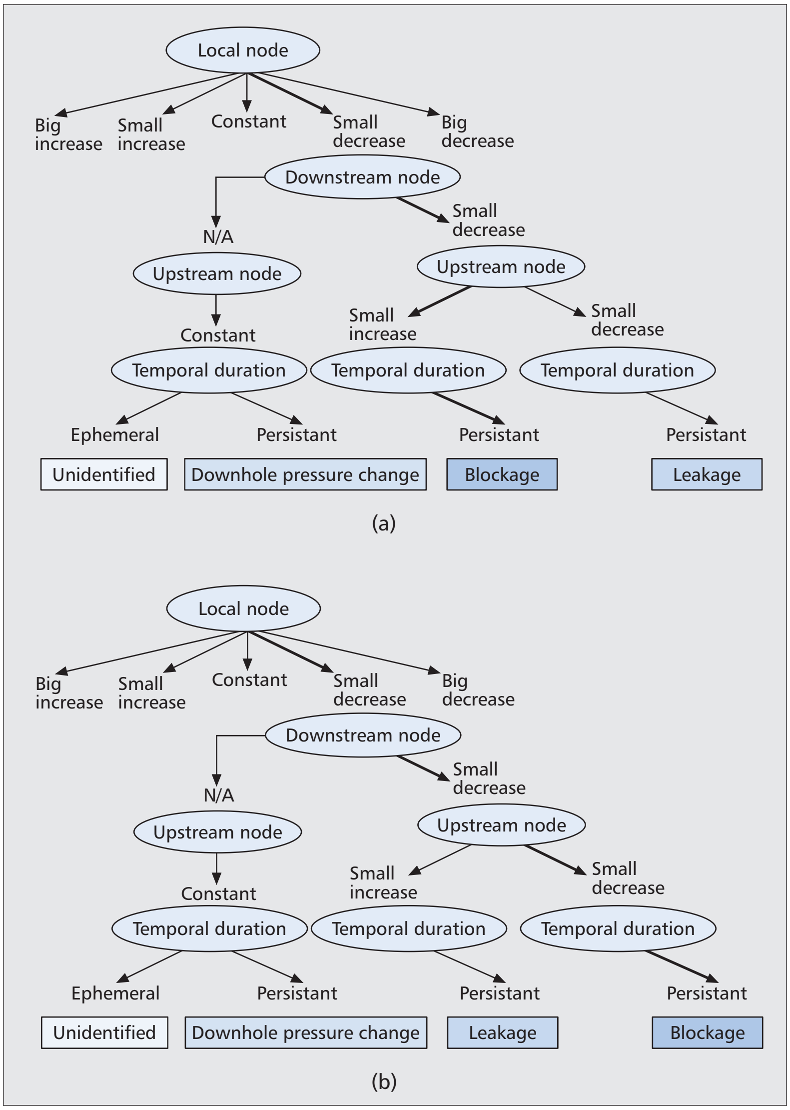


Figure 3. *Some branches of the decision trees on a) pressure; b) flow rate to identify the blockage.*

### Steamflood Monitoring Algorithm in SWATS

Our steamflood monitoring algorithm tries to determine the potential causes resulting in out of critical flow rate at the critical flow choke, which can be blockage, leakage, equipment malfunction, or outside force damage. Because a decision tree algorithm can be sensitive to the choice of thresholds, thresholds used in this steamflood monitoring algorithm are tuned with the domain knowledge such as the parameters of pipelines, equipment, and the out of critical flow rate. We plan to optimize these threshold values offline using reinforcement learning techniques such as the Markov decision problem (MDP).

Our proposed algorithm consists of two stages (single-node processing and multinode collaboration) with six components.

*Single-Node Processing* — At each node, our algorithem performs in-node sensor readings validation (using multimodal sensing) and noise reduction. Then it analyzes the temporal trend locally to detect the onset of events.

Step 1: In-Node Sensor Readings Validation — In order to check the validity of sensor readings, we cross-check data in a node from *multimodal sensing* (pressure and temperature) at a given sampling frequency, *f*.

Step 2: Noise Reduction — In order to *clean* the raw data samples, we compute the average pressure and average flow rate over *s* raw samples. A single averaged value consistes of a single valid sensor reading.

Step 3: Event Detection — SWATS uses *temporal trending* to detect events. For the temporal trending at a local node, we capture the temporal pattern of pressure and flow rate by performing a linear regression of sensor readings over the window *W*, which results in the numeric slope and intercept for both pressure and flow rate. We determine the five classes of temporal trend (big increase, small increase, constant, small decrease, big decrease) using the slope for pressure and flow rate.

*Multinode Collaboration* — Our decision tree algorithm utilizes collaboration of neighboring nodes to reach consensus in their detection and identification results for the same phenomena:

Step 4: In-Network Event Detection Validation — In order to verify the local detection result, SWATS cross-checks the classified local temporal trend with both upstream and downstream neighbors for the same *W*. For this cross-check and other communications with neighboring nodes, a reliable point-to-point routing protocol can be used; we address routing in our future work. SWATS uses *voting* over *Numneighbor* number of upstream and downstream nodes to further validate the determination made at a node. The local event detection is validated if the result from voting is larger than *V*, where *V* is the threshold for agreement in voting for pressure and flow rate.

Step 5: Problem Identification — Once the events are validated, SWATS identifies the causes of problems by using the decision trees (discussed below) with the inputs of the classified temporal trends for pressure and flow rate across a given number of neighbors (*Numneighbor*). To provide the stable detection result, SWATS reports the identification result after *k-consecutive* identical classification. In addition, each node utilizes metadata such as pipeline elevation, logical location, equipment maintenance schedule, and physical proximity to equipment such as generator, branch, Splitigator, and choke to identify problems correctly.

Step 6: Problem Localization — To localize the problem, SWATS finds the best matching, most upstream node with the rule for the identified problem in the decision trees. The node satisfying the specific condition in the decision trees is the origin of the problem.

### Decision Tree Algorithm

SWATS classifies the anomalies into five types of problems and six false alarms (Table 2). The decision tree checks from critical to trivial causes: problems to false alarms. The algorithm first compares the problem set using the rules in the decision tree. Then it tries to distinguish the candidate problems from the related false alarms using:
- In-depth comparison of phenomena using a decision tree that is programmed on all the nodes
- The prior information such as scheduled outage or pipeline elevation disseminated from the central database
- The reported event from other nodes
- The information about proximity to equipment

We now present an example of a decision tree used to identify blockage in a pipeline. Blockage causes a gradual drop over a long time (small decrease) in both pressure and flow rate at the local and downstream nodes, while the pressure at upstream nodes increases due to the constant injection with a valve, and the flow rate drops. Alternatively, if the pressure at upstream nodes drops and the flow rate increases, while all other conditions are the same as with blockage, the algorithm considers the problem to be a leakage. On the other hand, if both the pressure and flow rate for the upstream node do not change and those readings for a local node do change (either fluctuate, increase, or decrease), the algorithm identifies the event as a downhole pressure change, a false alarm. Figure 3 shows a part of the decision trees used in this example.

## Conclusion

We describe a new problem and designed an in-network processing system that successfully monitors a steamflood and waterflood pipeline to detect, identify, and localize anomalies such as blockage and leakage. In SWATS we created a decision tree algorithm for problem and false alarm identification by collaboratively exploiting spatial and temporal correlations in sensor readings. SWATS represents a new approach to oilfield monitoring that has the benefits of low cost, flexible deployment, continuous monitoring, and accurate problem detection, identification, and localization quickly, reliably, and accurately, thereby improving the current SCADA system. Because SWATS utilizes the changing pattern of flows over time and space, it works better in a scenario in which anomalies introduce non-negligible changes in flow rate.

## References


[1] S. Boyer, *SCADA: Supervisory Control and Data Acquisition*, Instrument Society of America, 2004.

[2] I. Stoianov, L. Nachman, and S. Madden, "PIPENET: A Wireless Sensor Network for Pipeline Monitoring," *Int'l. Conf. Info. Process. Sensor Net.*, Apr. 2007.

[3] J. C. P. Liou, *Pipeline Variable Uncertainties And Their Effects on Leak Detectability*, American Petroleum Institute, Nov. 1993.

[4] D. Erickson and D. Twaite, "Pipeline Integrity Monitoring System for Leak Detection, Control, and Optimization of Wet Gas Pipelines," *SPE Annual Tech. Conf. & Exhibition*, Oct. 1996.

[5] S.-F. Chien and J. L. G. Schrodt, "Determination of Steam Quality and Flow Rate Using Pressure Data From an Orifice Meter and a Critical Flowmeter," *SPE Production & Facilities*, vol. 10, no. 2, May 1995.

[6] Y. Kim *et al.*, "NAWMS: Non-Intrusive Autonomous Water Monitoring System," *ACM SenSys*, Nov. 2008.

[7] L. Gu *et al.*, "Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments," *ACM SenSys*, Nov. 2005.

[8] J. Liu *et al.*, "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications," *Int'l. Conf. Info. Process. Sensor Net.*, Apr. 2003.

[9] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.

[10] N. Ramanathan *et al.*, "Sympathy for the Sensor Network Debugger," *ACM SenSys*, Nov. 2005.

[11] F. Zhao *et al.*, "Monitoring and Fault Diagnosis of Hybrid Systems," *Trans. Sys., Man, Cybernetics: Part B*, vol. 35, no. 6, Dec. 2005.

[12] K. C. Hong, *Steamflood Reservoir Management: Thermal Enhanced Oil Recovery*, PennWell Books, Jan. 1994.

[13] S.-F. Chien, "Critical Flow of Wet Steam Through Chokes," *J. Petroleum Tech.*, Mar. 1990.


## Biographies


SUNHEE YOON [M] (sunheeyo@usc.edu) is a member of research staff at Stanford University. Her research interests are in using mobile phones for collaborative applications and wireless sensor networks. She was previously a member of the TinyOS developer group, and has done research on sensor network applications for oilfield and environment monitoring, geospatial decision making, and fault-tolerant overlay networks. She received her B.S. and M.S. from Sookmyung Women's University, Seoul, Korea, and her Ph.D. from the University of Southern California.

WEI YE (weiye@isi.edu) is a senior staff scientist at Broadcom Corporation. He contributed to this work when he was at USC, where he was a research assistant professor in the Computer Science Department and the Information Sciences Institute (ISI). He worked and published extensively in the area of wireless sensor networks and underwater acoustic networks. He received his B.S. and Ph.D. degrees in electrical engineering from Xidian University, China, and his M.S. degree in computer ccience from USC in 1991, 1996, and 2001, respectively. He has served as Associate Editor of *IEEE Transactions on Mobile Computing*.

JOHN HEIDEMANN [SM] (johnh@isi.edu) is a senior project leader at USC/ISI and a research associate professor at USC in the Computer Science Department. At ISI he leads I-LENSE, the ISI Laboratory for Embedded Networked Sensor Experimentation, and investigates network protocols and traffic analysis as part of the Analysis of Network Traffic (ANT) group. He received his B.S. from the University of Nebraska — Lincoln and his M.S. and Ph.D. from the University of California at Los Angeles. He is a senior member of ACM and a member of Usenix.

BRIAN A. LITTLEFIELD (BrianLittlefield@chevron.com) is a Facilities Advisor Engineer with the Heavy Oil Unit of the Chevron's Energy Technology Company (ETC). He received his B.S. in chemical engineering from Oregon State University, and his M.B.A. in business from California State University Bakersfield in 1981 and 1992, respectively.

CYRUS SHAHABI [SM] (shahabi@usc.edu) is currently a professor and the director of the Information Laboratory (InfoLAB) at the Computer Science Department and also a research area director at the NSF's Integrated Media Systems Center (IMSC) at USC. He has two books and more than 100 research papers in the areas of databases, GIS, and multimedia. He is a recipient of a U.S. Presidential Early Career Award for Scientists and Engineers (PECASE).