A Survey of Data Cleansing Techniques for Cyber-Physical Critical Infrastructure Systems

Chapter *in* Advances in Computers · June 2016

DOI: 10.1016/bs.adcom.2016.05.002

3 authors, including:

Sahra Sedigh Sarvestani
Missouri University of Science and Technology
124 PUBLICATIONS **1,172** CITATIONS

A Survey of Data Cleansing Techniques for Cyber-Physical Critical Infrastructure Systems

Mark Woodard, Michael Wisely, Sahra Sedigh Sarvestani

Missouri University of Science and Technology, Rolla, Missouri, USA

Abstract

Critical Infrastructure Cyber-Physical Systems heavily depend on accurate data in order to facilitate intelligent control and improve performance. Corruption of data in these systems is unavoidable, resulting from both intentional and unintentional means. The consequence of making control decisions based on erroneous or corrupted data can be severe including financial loss, injury, or death. This makes employing a mechanism to detect and mitigate corrupted data crucial. Many techniques have been developed to detect and mitigate corrupted data. However, these techniques vary greatly in their capability to detect certain anomalies and required computing resources. This article presents a survey of data cleansing techniques and their applicability to various control levels in a Critical Infrastructure Cyber-Physical Systems.

Keywords: Data cleansing, critical infrastructure, cyber-physical systems

Contents

1 Introduction **5**

2 Running ITS Example **6**
- 2.1 Mobile Infrastructure . . . 9
- 2.2 Static Infrastructure . . . 10
  - 2.2.1 Road Side Units . . . 11
  - 2.2.2 Traffic Control . . . 12
  - 2.2.3 Central Traffic Management . . . 12

3 Data Cleansing Techniques **12**
- 3.1 Sources Data Errors . . . 14
- 3.2 Data Error Detection . . . 15
- 3.3 Statistical Detection . . . 16
  - 3.3.1 Estimation-based Detection . . . 16
  - 3.3.2 Rule-based Detection . . . 17
  - 3.3.3 Learning-based Detection . . . 18
  - 3.3.4 Hybrid Detection . . . 20
- 3.4 Behavioral Detection . . . 20
- 3.5 Data Error Mitigation . . . 22

4 Classification of Techniques Based on Type of Anomaly **23**
- 4.1 Types of Data Anomalies . . . 23
- 4.2 Classification of Techniques . . . 24
  - 4.2.1 Rule-based Detection . . . 24
  - 4.2.2 Estimation-based Detection . . . 25
  - 4.2.3 Learning-based Detection . . . 26
  - 4.2.4 Behavioral Detection . . . 26
- 4.3 Static Elements of ITS Infrastructure . . . 27
  - 4.3.1 Road Side Units . . . 27
  - 4.3.2 Traffic signals and Dynamic Lanes . . . 28
  - 4.3.3 Central Traffic Management . . . 28
- 4.4 Mobile Elements of ITS Vehicle . . . 28
  - 4.4.1 Vehicle Operation . . . 29
  - 4.4.2 Navigation . . . 29
  - 4.4.3 Driver . . . 30

5 Classification of Techniques Based on Requirements and Constraints 30
5.1 Types of System Constraints 30
5.1.1 Hardware Resource Constraints 31
5.1.2 Network Induced Constraints 32
5.2 Classification of Techniques 32
5.2.1 Rule-based Detection 32
5.2.2 Estimation-based Detection 34
5.2.3 Learning-based Detection 35
5.2.4 Behavior-based Detection 35
5.3 Static Elements of ITS Infrastructure 37
5.3.1 Road Side Units 37
5.3.2 Traffic signals and Dynamic Lanes 37
5.3.3 Central Traffic Management 37
5.4 Mobile Elements of ITS Vehicle 38
5.4.1 Vehicle Operation 38
5.4.2 Navigation 38
5.4.3 Driver 39

6 Classification of Techniques Based on Type of Data 39
6.1 Types of Data 40
6.2 Classification of Techniques 40
6.2.1 Rule and Estimation-based Detection 40
6.2.2 Learning-based Detection 41
6.2.3 Behavior-based Detection 41
6.3 Static Elements of ITS Infrastructure 41
6.3.1 Road Side Units 42
6.3.2 Traffic signals and Dynamic Lanes 42
6.3.3 Central Traffic Management 43
6.4 Mobile Elements of ITS Vehicle 43
6.4.1 Vehicle Operation 43
6.4.2 Navigation 43
6.4.3 Driver 43

7 Classification of Techniques Based on Hierarchical Location 44
7.1 Levels of System Hierarchy 44
7.2 Classification of Techniques 45
7.3 Static Elements of ITS Infrastructure 45

7.3.1 Road Side Units 46
7.3.2 Traffic signals and Dynamic Lanes 46
7.3.3 Central Traffic Management 46
7.4 Mobile Elements of ITS Vehicle 46
7.4.1 Vehicle Operation 46
7.4.2 Navigation 47
7.4.3 Driver 47

8 Conclusion **48**

1. Introduction

Modern society is becoming more and more dependent on access to accurate real-time and stored information. Critical infrastructure systems are not different as they transition from purely physical systems to Critical Infrastructure Cyber-Physical Systems (CPS) to meet performance requirements and growing demands. In a CPS, a layer of cyber infrastructure is added to the the physical infrastructure to improve functionality. This cyber infrastructure consists of intelligent embedded systems, communication capabilities, distributed computing, and intelligent control [1]. The cyber infrastructures facilitates intelligent control to better adapt to changes in demand and production. Smart power grids, intelligent water distribution networks, and smart transportation systems are all examples of modern CPS.

Intelligent control systems in CPS make decisions by processing real-time and previously stored data. The intelligent control systems calculate optimal control settings by processing data from the controller's immediate area as well as system-wide data. Processing data from system-wide sources prevents adverse consequences caused by localized control. Bakken et al. [2] discuss how the use of real-time measurements can address many of the power generation and distribution challenges in the smart grid. Hoverstad et al. [3] discusses the need for data cleansing on load prediction algorithms used in the smart grid. Specifically the added robustness achieved by removing sensor data errors prior to executing the prediction algorithms.

CPS reliance on real-time field data makes the systems susceptible to severe consequences caused by corrupted data. Buttyán et al. [4] presents the design and protection challenges of cyber infrastructure in CPS, discussing fault tolerance, security, and privacy of sensor nodes, networking protocols, and operating systems. One example of severe consequences resulting from cyber infrastructure failure in a stock market is discussed by Kirilenko et al. [5]. In August of 2012, a financial computing system failure consisting of a software error cost the mid-size financial firm Knight Capital $10 million per minute. While this failure had economic consequences, failures in other critical infrastructure and manufacturing systems could result in physical injury or loss of life. To prevent these failures it is essential to detect and mitigate data failures. This process is know as data cleansing.

The motivation for the survey presented in this chapter are the catastrophic critical infrastructure failures in recent history. Miller et al. [6] discusses the failure of a Bellingham, WA gas pipeline which ruptured and

within 1 1/2 hours leaked 237,000 gallons of gasoline into a creek flowing through Whatcom Falls Park in June 1999. The gasoline ignited burning approximately 1 1/2 miles of forest along the creek killing 3 people and injuring 8 others. Due to the company's practice of performing live database development work on critical components, real-time sensor data was unavailable to control systems. As a result, the control systems were unable to react to the failure. Another failure, discussed by Berizzi [7] and Buldyrev et al. [8], occurred in Italy on September 28, 2003. The failure was triggered by a single line failure near the Swiss-Italian border, which caused a cascading failure resulting in half of Italy being without power for multiple days. This local failure led to Internet communication network nodes failures, which in turn caused further breakdown of the power systems control. Although these examples are not the result of corrupted data, they demonstrate how CPS rely on accurate, real-time data and the potential for failures induced by data corruption.

This chapter serves as an extension to our previous work [9], and presents a survey of data cleansing techniques and classifies them based on their applicability in CPS. Figure 1 is a taxonomy of the topics presented in this chapter drawing from recent papers as shown in Figure 2.

The remainder of this chapter is structured as follows. In Section 2, we introduce an example Intelligent Transportation System, which is a type of CPS. This example application will be used for comparison of techniques. In Section 3, we discuss sources of corrupted data and introduce a number of data cleansing techniques. In Section 4, we classify techniques based on the type of anomalies which can be detected and mitigated. In Section 5, we classify techniques based on hardware, communication and time constraints. In Section 6, we classify techniques based on the type of data which can be cleansed by each technique. In Section 7, we classify techniques based on their hierarchical location within the system. Lastly, Section 8 addresses future directions for this research.

2. Running ITS Example

Intelligent Transportation Systems (ITS) are CPS aimed at improving performance and safety of transportation networks [10]. ITS refers to all modes of transport including road, rail and air. All of these transportation systems have similar challenges [11]. However, road transportation systems

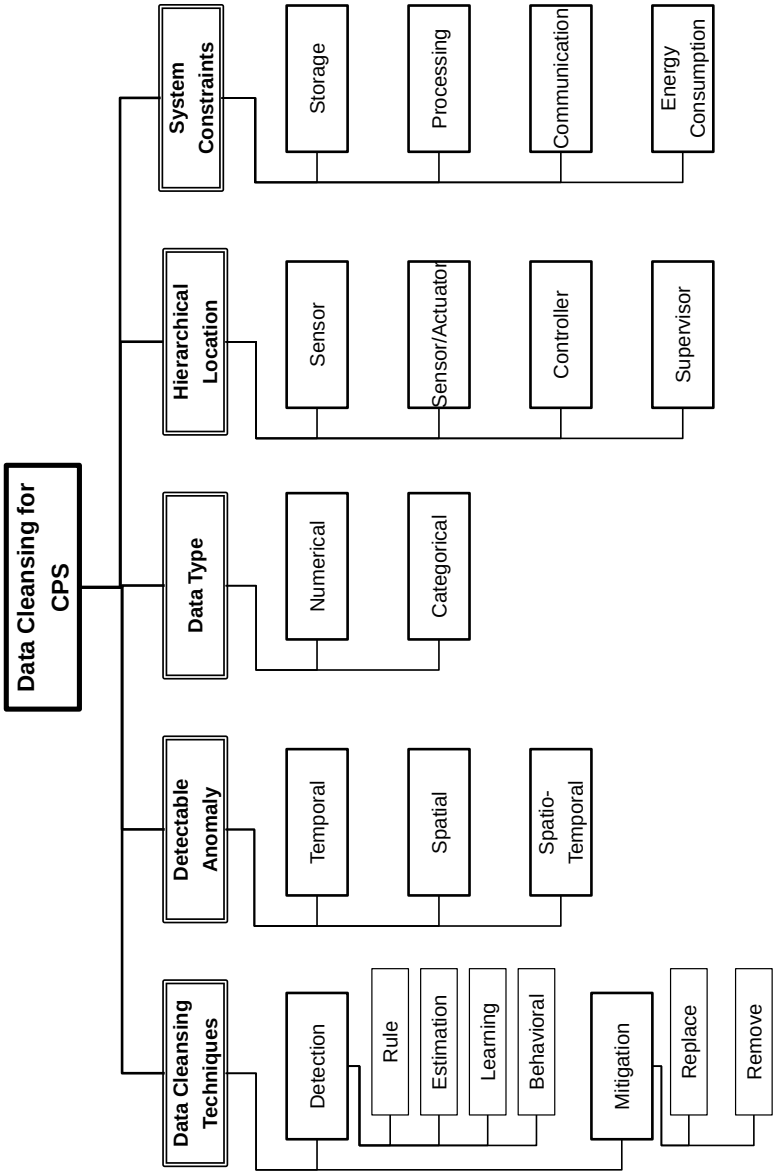

Figure 1: Taxonomy of Data Corruption Research.

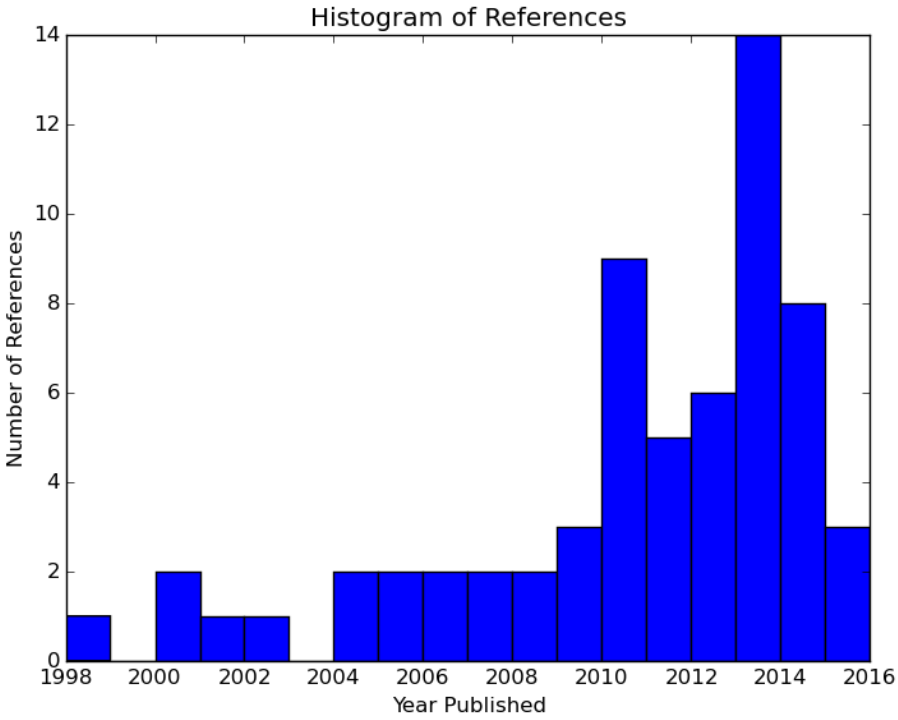

Figure 2: Histogram of Papers Cited.

are used as an example system in this article. All following references to ITS will refer to road transportation systems.

ITS technologies include everything from basic traffic management systems such as vehicle navigation and traffic signal control to more advanced systems that allow Vehicle to Everything (V2X) communication to improve control and information dissemination between vehicles, roadside units, infrastructure, pedestrians, and cyclists [12]. ITS technologies also include unmanned vehicle technologies including self-driving vehicles and automatic parking systems. The example ITS system used in this article focuses on Infrastructure to Vehicle (I2V), Vehicle to Vehicle (V2V), and Vehicle to Infrastructure (V2I) communication as well as information dissemination and the intelligent traffic control that communication facilitates [13].

These services can be classified based on where in the infrastructure the required computer processing occurs. Elements of ITS infrastructure can be classified as mobile infrastructure or static infrastructure.

2.1. Mobile Infrastructure

Mobile infrastructure consists of all ITS elements without a static network connection i.e. vehicles.

Vehicles on modern roadways range from classic cars with no digital systems to fully autonomous, unmanned vehicles. ITS systems must be designed to accommodate the full range of vehicles. We will categorize vehicles into three classes based on their functionality: traditional, intelligent and unmanned vehicles.

Traditional vehicles are all vehicles without V2V or V2I communication capability. They do not provide data directly to the ITS. Vehicles in this category may or may not have I2V capabilities which would provide the driver with additional information about congestion, such as 2-way GPS traffic updates. This category also includes vehicles with adaptive cruises control or advanced collision avoidance systems such as blind spot sensors and back up sensors. While these technologies improve vehicle safety and control, they do not provide data to other components in an ITS.

Intelligent vehicles are vehicles with V2V or V2I communication capability that are controlled by a human driver. These vehicle are equipped with an on-board sensor suite with the capability to monitor the locations and actions of surrounding vehicles as well as detect road obstacles and conditions. These vehicles utilize on-board processing and storage systems to analyze collected data. Collected information is communicated to surrounding vehicles or the

ITS infrastructure via road side unit. The wireless communications capability falls into two categories based on the indented recipient. Short-range communication is used to communicate with neighboring vehicles and road side units using the IEEE 802.11p protocol, which was specifically developed for ITS and mobile ad hoc or mesh networking. The second type of communication is longer range communications using IEEE 802.16, WiMAX, GSM, or 3G. This type of communication is used to communicate with a central traffic management center or to access other relevant data sources.

Unmanned vehicles are vehicles with the same capabilities as intelligent vehicles. However, the collected data is used to directly control the vehicle rather than to assist a human driver. In addition, collected data may be provided to other components of an ITS.

2.2. Static Infrastructure

Static infrastructure within ITS includes purely physical infrastructure including roads, highways, and bridges as well as the static, cyber-enhanced infrastructure. The static ITS infrastructure does not move during operation and includes devices such as traffic signals and road sensors.

The cyber layer of an ITS system is structured and functions similar to a sensor database architecture. Sensor database architectures are classified based on where the data is stored. These architectures range from traditional sensor databases, where data is stored in a centralized database, to distributed databases, where every sensor node has its own database.

Traditional sensor networks described by Akyildiz et al. [14] are not applicable to ITS due to large networking overhead and delay. Another sensor network architecture is the distributed sensor database system, which places databases closer to the controller and sensor nodes. This architecture can be thought of as a data logging network. In this type of sensor network, all sensors send all sensed data to secondary storage, which can be retrieved in bulk. This architecture permits duplication of stored data to improve performance. Distributed database architectures are not specific to sensor networks. Many approaches to distributed databases are summarized by Hurson et al. [15] including federated and multi-databases which address issues such as data distribution and transparency as well as query and transaction processing. A further distributed sensor network architecture is discussed by Bonnet et al. [16] is the sensor database model. In this architecture, each sensor node holds a database that can be dynamically queried. Tsiftes et al. [17] discuss this sensor network architecture and propose a database management

system.

A practical ITS would use a combination of distributed and sensor database architectures at various hierarchical levels of the system. Combining these architectures may improve performance by limiting the communication of raw data, energy, bandwidth, and scalability. Additionally, this architecture improves maintainability and fault recovery by storing performance data at the sensor nodes. Amadeo et al. [12] discuses the benefits of using a Named Data Networking model for ITS. Which would require this type of data base architecture. However, this architecture has challenges including the system updates and database management due to its distributed nature.

2.2.1. Road Side Units

A Road Side Unit (RSU) collects traffic data from a static sensing area along a road and transmits data to traffic control devices as well as a Central Traffic Management center. These devices also serve as an information source for intelligent vehicles to collect future traffic information [18].

RSU can sense traffic information using a number of methods. One method for collecting traffic information is the triangulation method. Triangulation uses mobile phones as anonymous traffic probes. The phones transmit presence announcement signals to the mobile phone network which can be observed by an RSU. This network data is collected and analyzed using triangulation and converted into traffic flow information. This method works for all types of vehicles, provided that a powered-on mobile phone is in the vehicle. Another method is vehicle re-identification. This method uses some unique identification from an in-vehicle devices, such as Bluetooth MAC addresses or an RFID toll tags. As a vehicle travels along a route, multiple RSU detect a specific vehicle and record a time stamp. This information is shared and analyzed to determine speed, travel times, and traffic flow for a road segment. This method requires technology within the vehicle to transmit a unique id. Conveniently, most modern vehicles use wireless communication between components, which can be used to identify a vehicle. Lastly, V2I communication provided by intelligent vehicles can be used to collect traffic flow data. Many other techniques can be used to collect traffic flow data such as two-way GPS or satellite navigation systems, inductive loop detection, traffic video cameras, and audio detection.

RSU use information from multiple sources to create an accurate picture of traffic flow on a specific road segment by using data fusion based approaches to intelligently combine data. These data fusion techniques create

a more accurate representation of the traffic then any single sensing method.

2.2.2. Traffic Control

ITS allows for traffic control systems that are more advanced than traditional timed traffic signals [19]. One type of control device is intelligent traffic lights, which use traffic data collected at the local intersection, as well as future traffic information provided by RSUs, to create a dynamic time schedule to maximize the flow of traffic through an intersection. Another control system is variable speed limits. These systems work to minimize traffic density in congested areas by dynamically changing the speed limit of roads based on weather conditions, road conditions, or the presence of congestion areas. Lastly, dynamic lanes can be used to provide more inbound or outbound lanes depending on the flow of traffic as traffic in many metropolitan areas is not symmetric.

2.2.3. Central Traffic Management

A Central Traffic Management (CTM) system could be centralized or distributed over a control area. In either case, a CTM collects and analyzes data from intelligent vehicles, unmanned vehicles, and RSU to facilitate control decisions [20]. Each central traffic management office would have a server for data storage and processing. The processed data could be used for high level coordination of the traffic control devices. The central office could then broadcast data back to vehicles to improve navigation and control.

3. Data Cleansing Techniques

In general, data cleansing involves exploring a data set, detecting possible problems, and attempting to correct detected errors [21]. Definitions of data cleansing vary depending on the field and application.

Traditionally, data cleansing was the detection and removal of duplicated records of differing formats. This type of data cleansing was conducted in data warehouses; executed when multiple databases are merged, also called merge and purge. [22] Duplicate identification in warehouse data cleansing is also called record linkage, semantic integration, instance identification, or object identity problem. Bertossi et al. [23] discusses matching dependencies for data cleaning of similar attributes for clean query answering. Modern cleansing also includes detection of erroneous data. This type of data cleansing is a part of data/information quality management referred to as Total

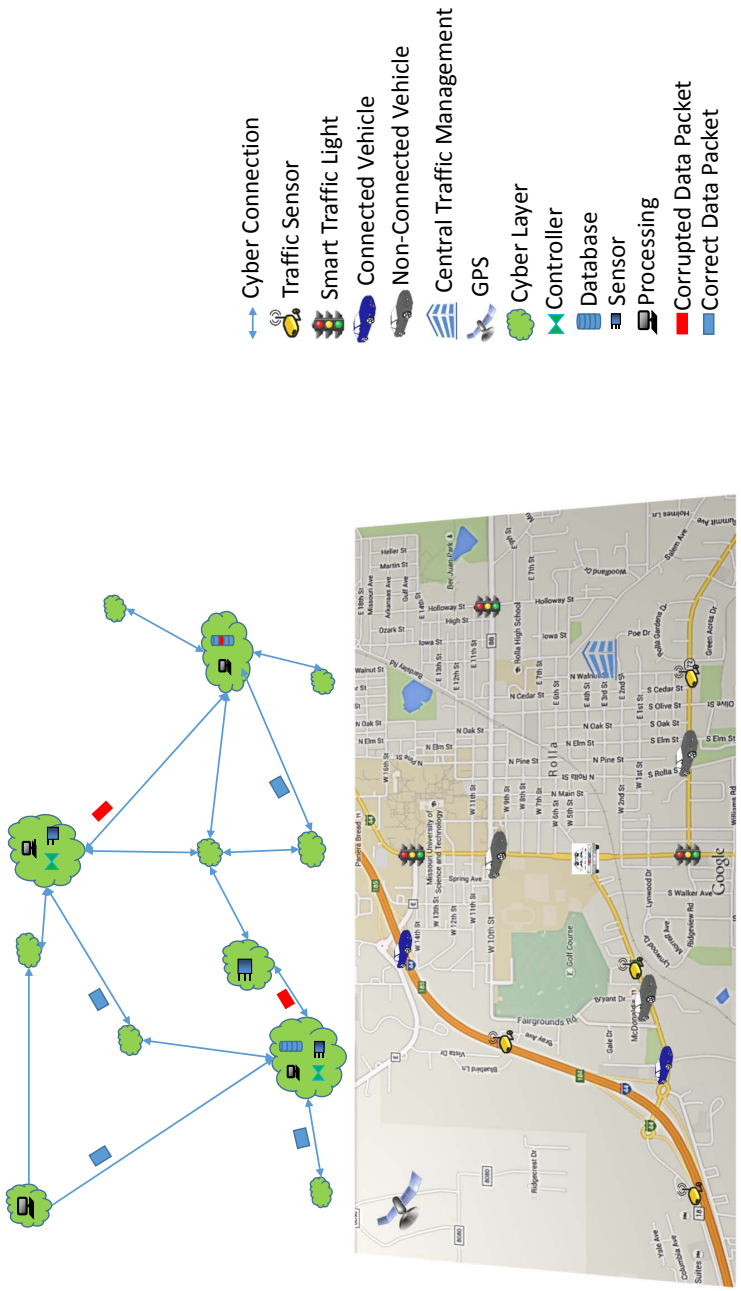

Figure 3: An intelligent water distribution network

Quality Data Management (TQDM). [24] In line with TQDM, Dallachiesa et al. [25] present NADEEF as an end-to-end off-the-shelf semiautomatic data cleansing solution. Budka et al. [26] presents the challenges of data preprocessing and cleansing for prediction in control systems using soft sensors identifying some if the key challenges. This type of data cleansing is directly applicable to critical infrastructure CPS applications, in which a huge amount of data is collected and used for control. TQDM includes clerical error and organizational behavior aspects which are beyond the scope of CPS applications.

Data cleansing, as it pertains to CPS, occurs in two phases: detection and mitigation, as described in Woodard et al. [9]. Before we discuss detection and mitigation, an overview of data the sources of erroneous data is presented.

3.1. Sources Data Errors

In order to discuss sources of erroneous data, an overview of fault tolerance and dependability terminology is necessary. More detailed discussions are provided in Aviźienis et al. [27]. In fault tolerance and dependability, a system is describe as being in a specific state in the presence of a disruptive event based on the systems ability to provide its specified service. The terms used to describe the threats to system operation are failure, error, and fault. A system failure is the state in which a system does not comply with system specifications. An error is a system state that may induce a failure. More specifically, a failure occurs when an error causes an alteration of the system services. A fault is the cause of an error. Faults are classified based on their persistence, activity, and intent. Corrupted or erroneous data can be a failure, error, or fault depending on its location in the system. Producing corrupted data is a system failure; processing corrupted data is a system error; accepting corrupted data as input is a system fault. Erroneous data can be created within a system via intentional and unintentional means.

Intentionally erroneous data is the result of an attack. Attacks can be classified as purely cyber, purely physical, or combined cyber-physical depending on the source of the attack. Mo et al. [28] describe these types of attacks on critical infrastructure systems. An example of a cyber-physical attack on an intelligent water distribution system is provided by Amin et al. [29]

Unintentionally erroneous data is the result of corruption during communication, processing, or storage in addition to inaccurate sensor readings. Cebula et al. [30] provide a very detailed taxonomy of cyber and physical

risks to computer communication, processing, or storage information technology assets. The input data to a system may be erroneous as Aggarwal et al. [31] discusses the need for data cleaning of sensor readings. Sensor readings are often created by the inherently noisy process of converting a measured quantity such as voltage into another measured quantities such as temperature. Additionally, errors can be introduced by external conditions, sensor aging, miscalibration, or intermittent failures of sensors. However, in many large distributed systems, the cause of the erroneous data is difficult to determine and the same data cleansing techniques may be used regardless of the source of the data error.

3.2. Data Error Detection

As stated above, data errors can be produced by a number of sources including miscalibrated or faulty sensor hardware as well as errors in processing, storage, and communication. The detection phase of data cleansing is the identification of potential errors. In general, this is done by locating anomalies in the system. Rajasegarar et al. [32] discuss the importance and general challenges of anomaly detection in sensor networks as it pertains to fault diagnosis, intrusion detection, and monitoring applications. The primary challenge in the development of any anomaly detection algorithm is that sensor networks are highly application and domain dependent. Examples of domain specific techniques are proposed by Yin et al. [33], who model wind turbine data, and Freeman et al. [34], who model aircraft pilot-static probe data. Both of these anomaly detection techniques are able to detect anomalies in data with significant measurement noise and unknown distribution. However, these techniques are not suitable for other domains and do not scale to the size of CPS. Another major issue in CPS data corruption detection is determining when anomalous data is truly a data error. Tang et al. [35] investigate the trustworthiness of sensor data and propose a method called Tru-Alarm to eliminate false alarms by filtering out noise and false information. Tru-Alarm is able to estimate the source of an alarm by constructing an alarm graph and conducting trustworthiness inference based on the graph links.

Detection techniques can be classified by the method employed to detect potential data errors. Zhang et al. [36], Chandola et al. [37], and Fang et al. [38] provide comprehensive overviews of anomaly detection techniques. Table 1 below is a summary of these approaches and recent advances in

anomaly detection. These approaches include statistical detection and behavioral detection approaches. Statistical detection approaches can be further classified into estimation-based, rule-based detection, learning-based detection and hybrid approaches.

3.3. Statistical Detection

Data errors can be detected by locating data anomalies or statistical irregularities in the data. Faulty sensors typically report easily distinguishable extreme or unrealistic values, however, not all data anomalies are the result of erroneous data. Extreme environmental variations can produce data anomalies that should be distinguished from data errors. Statistical anomaly detection approaches detect anomalies by checking how well the data fits a statistical distribution model of the data. Anomalies are detected when data values fall outside an acceptable tolerance set by the user. Statistical anomaly detection uses statistical metrics such as mean, standard deviation, range, etc. to estimate normal data values and detects outliers if the data falls outside this expected range. The statistical model used to capture the distribution of the data can be assumed or estimated based on previously recorded data.

3.3.1. Estimation-based Detection

Estimation-based statistical approaches use probability distribution models of the data to detect anomalous values. Probability distribution models can be parametric or non-parametric based [36]. Parametric models assume knowledge of the data distribution, i.e. Gaussian-based model. Non-parametric models such as histograms and kernel density estimators, do not assume knowledge of the data distribution. Histogram models estimate the probability of data occurrences by counting the frequency of occurrence and detect anomalies by comparing the new data with each of the categories in the histogram. Kernel density estimators estimate the probability distribution function (pdf) for some normal data. An anomaly is detected if new data lies in the low probability region of the pdf. Fang et al. [39] propose an energy efficient detection method using an ARIMA model. The ARIMA model is a statistical model used time series analysis. It has three terms, auto-regression (AR), integration (I), and moving average (MA) to represent the data. The auto-regression term compares the new value to historical data using linear regression. The integration term differences the original data series to make the process stationary. The moving average term captures the influence of

extreme values. Each sensor node maintains a matrix of all maximum and minimum differences between itself and its neighbors. Then, using a voting mechanism, values are marked as valid or erroneous. Some estimation-based statistical approaches do, however, have the potential to bias the cleansed data. Bilir et al. [40] show that an incorrectly computed residual covariance matrix while using sparse inverse covariance estimation method can result in data errors. The miscalculation can lead to the sequential removal of good data eventually leading to a biased estimation. The authors also present a method to correct this error.

Estimation-based approaches are mathematically proven to detect anomalies if a correct probability distribution model is used. However, knowledge of the probability distribution is unavailable in many real-world applications, making non-parametric approaches more useful. Non-parametric approaches require additional hardware and storage, but they are able to detect anomalies very quickly.

3.3.2. Rule-based Detection

Rule-based statistical approaches are the simplest form of anomaly detection. An acceptable lower and upper limit for the data is set and any value outside of this range is an anomaly. This technique requires only the definition of an outlier to be set, making it inflexible and resulting in many false positives or undetected anomalies if the tolerance is set too low or high. The benefits of this technique are that it is fast, requires no additional storage capability, and can be implemented in few lines of code; making it ideal for sensor nodes.

Another simple rule based statistical approach to anomaly detection is statistical inference using the mean and variance of a data set. Ngai et al. [41] uses a chi-square test performed over a sliding window. In this example, the system determines that at least one value in the sliding window is anomalous if the chi-square value falls outside of a range specified by the user. The acceptable level must be configured prior to operation. This node-local approach can detect anomalies in the data stream of a single sensor while imposing no additional network overhead. Statistical inference techniques cannot adapt to changing ranges, which are very common in long-term wireless sensor network installations. Panda et al. [42] propose another very simple rule-based anomaly detection method which calculates the mean and variance of a set of neighboring sensors to determine if a sensor is faulty. Rule-based statistical methods can be implemented on minimal hardware

and detect anomalies very quickly provided the data is well behaved and the rules are set appropriately. As such, other approaches have been developed that do not rely on user-set parameters.

3.3.3. Learning-based Detection

Learning-based statistical approaches utilize data mining, clustering, and classification algorithms to group data based on data similarities [38]. An anomaly is detected when data does not belong to a group. These techniques have very high detection rates but require additional processing and storage hardware.

A decentralized clustering approach to anomaly detection is set forth by Rajasegarar et al. [43]. This approach was designed specifically for hierarchical (tree-based) networks. Leaf nodes take sensor readings and cluster them into fixed-width clusters. Each non-leaf node in the tree takes clusters from its children and merges them together. Anomaly detection is performed at the root node by finding clusters that are further away from other clusters by more than one standard deviation above the average cluster distance. Chang et al. [44] use an Echo State Network (ESN), a neural network in which all neurons are connected to each other, to perform anomaly detection. The ESNs are trained before the nodes are deployed, so they are not very flexible. They operate in a similar fashion to Bayesian networks where the sensor's value is compared to the value predicted by the ESN. The advantage of using a neural net in this case is that it has much lower CPU and RAM requirements than a Bayesian network. An improvement to this approach is put forth by Obst [45]. Instead of building recurrent neural networks beforehand, each node communicates with its immediate neighbors to build a model of the values observed by its sensors. This model is then used to estimate anomalies in the readings.

Classification approaches use a learned model to organize data into classes; in this case, normal or anomalous. One classification approach uses Bayesian networks to model sensor values and predict when values are anomalous [38]. Mayfield et al. [46] have developed a tool called ERACER that uses relational dependency networks to correct anomalous data and fill in missing data. The tool runs on a sensor network base station and develops linear models of sensor data, taking into account readings from other sensors at that node and readings from neighbor nodes. Another example using Bayesian networks is [47], where the concentration of various gases in a mine's atmosphere is monitored. The network models sensor values over time as well as physical

relationships between sensors. The system learns a baseline for the mine's gas concentrations that adapts to the natural fluctuations in gas concentration. It can detect both single-sensor anomalies and multi-node anomalies and events.

Ni et al. [48] propose using a hierarchical Bayesian space-time model to detect trustworthy sensors. The disadvantage of this technique is the amount of work required to set up the model. This technique results in excellent anomaly detection if model accurately represents the data. However, as with all models, if the model is poorly matched to the data, the system performance degrades. A more advanced classification approach is the nearest neighbor approach. This approach uses a distance metric, for example Euclidean distance, to determine how similar a value is to its neighbors. An anomaly is detected if the distance between neighbors is more than a user specified threshold. Expanding on this approach, Branch et al. [49] use a distributed algorithm to detect outliers as data propagates through a sensor network. In this approach, each node maintains a set of outlier data points from itself and its neighbors. A ranking function is used to map data values to non-negative real numbers which indicate the degree to which the data value can be regarded as an outlier with respect to the dataset. Nodes transmit data they suspect will cause the outlier set of their neighbors to change. This is similar to a distributed k-nearest-neighbors classification approach. This technique is flexible with respect to the outlier definition, allowing for dynamic updating and in-network detection, reducing bandwidth and energy consumption.

A method to improve the performance of learning-based approaches uses principal component analysis (PCA) to reduce the dimensionality of a data. PCA is a technique that uses spectral decomposition to find normal behavior in a data set. PCA is used to reduce dimensionality before detection by finding a subset of data which captures the behavior of the data. Chitradevi et al. [50] proposes a two-step algorithm. First, a PCA model is built that can be used for fault detection. Second, the mahalanobis distance is used to determine the similarity between the current sensor readings against the developed sensor data model. However, conventional PCA approaches are sensitive to data anomaly frequency in collected data and fail to detect slow and long-duration anomalies. Xie et al. [51] addresses this problem by using a multi-scale principal component analysis (MSPCA) to detect anomalies and extract and interpret information. MSPCA uses both wavelet analysis and principal component analysis. The time-frequency information of the data is

captured using wavelet analysis while principal component analysis is used to detect data anomalies. This technique allows for detecting gradual and persistent anomalies with different time-frequency features.

3.3.4. Hybrid Detection

Lastly, a hybrid approach is proposed by Warriachet al. [52] to detect data anomalies based on the three methods. Combining rule-based methods, estimation-based, and learning-based methods, they are able to leverage domain and expert knowledge, sensor spatial and temporal correlations and inferred models for the faulty sensor readings using training data. This approach has the benefits of the above approaches but also requires more processing capability and power at sensor nodes.

3.4. Behavioral Detection

Behavioral detection approaches have also been implemented to detect the erroneous data in the system by detecting anomalous behavior of a system rather than analysis of the data. Many of these approaches were developed as a part of intrusion detection systems (IDS). Liao et al. [53] provide a comprehensive overview and classification of general computing IDS approaches. These classifications are signature-based detection, anomaly-based detection, and stateful protocol analysis. Signature-based detection, also known as knowledge-based detection, detects a pattern or string that corresponds to a known attack. This technique is limited to detecting known attacks from previously analyzed events. Anomaly-based detection determines the normal behavior of the system and detects anomalies by comparing the current behavior with the normal behavior model. Anomaly-based detection can monitor any type of activity, including network connections, number and type of system calls, failed login attempts, processor usage, number of e-mails sent, etc. This approach can detect both known and unknown attacks. Lastly, stateful protocol analysis, also known as specification-based detection, compares a vendor-developed profile of specific protocols to current behavior. An example would be monitoring protocol states such as pairing requests and replies. Modi et al. [54] provide a survey of IDS techniques used for cloud computing. Many of the approaches use techniques similar to statistical anomaly detection, as well as neural networks and fuzzy logic.

CPS specific IDS approaches have also been developed. Buttán et al. [4] discuss the WSAN4CIP Project which investigated a number of attack detection methods to determine if a sensor node is compromised. The project

Detection Approaches	Description
Rule-Based	Rule-Based detection approaches set acceptable limits for data values. These limits can be determined from an outlier set or using statistical inference.
Estimation-Based	Estimation-Based detection approaches uses probability distribution models of the data to detect anomalous values. These approaches requires knowledge of the data distribution or the use of histograms or kernel density estimators to assume a distribution. They are mathematically proven to detect anomalies if the correct distribution model is used.
Learning-Based	Learning-Based detection approaches utilizes data mining, clustering, and classification algorithms to group data. Anomalies are detected when new data does not belong to a group.
Behavior-Based	Behavior-Based detection approaches utilizes signature, anomaly, and stateful protocol analysis of a system to detect anomalies is system behavior.

Table 1: Summary of Statistical Anomaly Detection Approaches

included intrusion detection and prevention techniques that were adapted to the wireless environment. A micro-kernel in the sensor node operating system supports multiple levels of security and determines if the code deployed on a sensor node is unchanged. Mitchell et al. [55] provide a detailed review of CPS related IDS research. In addition to IDS, for traditional networked computing systems, CPS IDS monitors both the embedded components and the physical environment, which under attack may exhibit abnormal properties and behavior. However, this is complicated by legacy technology still used in many CPS. Some legacy components are based on mechanical or hydraulic control with no cyber component, making them difficult to modify or access. Thus CPS IDS must define acceptable component behavior based on sensor readings of the physical environment.

3.5. Data Error Mitigation

Once anomalous data has been detected it must be mitigated in order to prevent operational disruptions in the CPS. Therefore, it is important that the detection and mitigation process does not hinder normal operation. It is essential that data errors are detected and mitigated while the data is still viable.

Detected data errors or missing data can be mitigated in a number of ways. In some higher level data cleansing activities, multiple cleansing alternatives are available on a system. In this case, automatic data cleansing requires a set of policies to determine the appropriate option. Mezzanzanica et al. [56] present a model-based approach for developing a policy for the data cleansing of a data set. In some cases, data cleansing requires a domain expert to be involved in the data cleansing effort. Gschwandtner et al. [57] presents an interactive visual analysis tool called TimeCleanser. This system is designed for data cleansing of time-oriented data. TimeCleanser combines semi-automatic data quality checks and data visualizations to assist the user.

In CPS, the mitigation technique is additionally dependent on the criticality and valid time interval of the data. Mitigation can be accomplished by correcting, replacing, or ignoring the data error. In many CPS applications, the useful life of a single piece of data is very short making some correction or replacement techniques inappropriate. Additionally, many correction and mitigation techniques require a great deal of computation making the energy consumption prohibitive. However, in other applications, missing and corrupted data minimizes the quality of information and ignoring these errors may cause a serious effect in data analysis.

Gantayat et al. [58] provide a review of research on missing or incomplete data. A variety of techniques are used to generate predicted values. Many of these approaches are very similar to the anomaly detection techniques discussed. The following are approaches for mitigating missing and corrupted data:

- Imputation: This technique replaces missing data values with an estimation based off the data stream's probabilistic model.
- Predicted Value Imputation: This technique replaces missing data with estimated values based on the data set. The estimation methods vary in complexity from mean or mode values to more complex estimates from training data.

- Distribution Based Imputation: This technique replaces missing data using a classification algorithm. A set of pseudo-instances is created when a missing value is encountered. Each pseudo-instance is tested. The replacement value is selected using a weighted comparison.
- Unique Value Imputation: This technique replaces the missing value using simple substitution from historic information.
- Replacing Missing Data: This technique replaces the missing data with a value from a test case that resembles the current data set.
- Rough Sets: This technique uses lower and upper approximations to determine a replacement value. The benefit of this technique is that there is no need for preliminary or additional information about the data. A number of extensions to rough set have been proposed including tolerance relation, non-symmetric relation, and valued tolerance relation.
- Similarity Relation: This technique replaces the missing data after making a generalized decisions based on the entire data set.

These techniques can be employed to replace corrupted or missing data allowing for correct execution.

4. Classification of Techniques Based on Type of Anomaly

In this section we will classify the data cleansing techniques presented in Section 3 based on the type of anomalies each technique is able to detect. Using this classification, we will then discuss where in an ITS each cleansing technique would be most appropriate. In order to classify these techniques, an understanding of various types of data anomalies must be presented.

4.1. Types of Data Anomalies

Jurdak et al. [59] classify data anomalies into three broad categories: temporal, spatial, and spatio-temporal. Table **??** contains a summary of these types of anomalies.

Temporal data anomalies are local to one sensor node and can be detected by observing sensor values over time and observing a number of attributes which indicate an error. These attributes include high variability or lack of

variability in subsequent sensor readings, gradual reading skews, or out-of-bound and extreme readings. Examples of failures that result in this type of anomaly are as follows. A sensor may fail into a locked state or fail to obtain new samples making the sensor reading remain the same over long periods of time. Another example is as a sensor loses calibration, its data values drift away from the true value resulting in a gradual skew of sensor readings. A major malfunction of the sensor could produce out-of-bound readings that are physically impossible. And lastly, high variability in sensor readings could arise from sensor voltage fluctuations. However, high variability can also result from major changes in the sensed environment. The detection of temporal data anomalies requires the data stream from a single sensor node as well as stored historical data.

Spatial data anomalies occur when one sensor's data readings are significantly different from the readings of surrounding nodes at a single time period. Detecting this type of anomaly requires a network-aware algorithm. Data redundancy between sensors is exploited to determine which sensors may have faulty readings. This type of detection is only possible for certain types of data with low spatial variation, such as air temperature or humidity. In this type of data, a change in one area will affect the readings of surrounding sensors. Networks with high spatial variation, especially video and audio data, are usually incapable of detecting such anomalies.

Spatio-temporal anomalies combine attributes of both temporal and spatial anomalies. These anomalies are somewhat rare but also more difficult to detect. For example, a storm progressively moving through an area causing sensor nodes to fail would be a spatio-temporal anomaly. Spatio-temporal anomaly detection requires both a network-wide detection algorithm as well as data streams from multiple sensors.

4.2. Classification of Techniques

A variety of the techniques presented in Section 3 can be employed to detect each of these types of data anomalies. Table 3 contains a summary of the cleansing techniques classified by detectable anomaly.

4.2.1. Rule-based Detection

The detection capability of rule-based statistical approaches depends a great deal on the implementation. The simplest rule-based statistical approach which rejects data if it falls out side of a specified range of acceptable

Type of Data Anomaly	Description
Temporal	Temporal anomalies in sensor readings exhibit high variability in subsequent sensor readings, lack of change in sensor readings, gradual reading skews, or out-of-bound readings.
Spatial	Spatial anomalies in sensor readings are significantly different from surrounding nodes' readings.
Spatio-temporal	Spatio-temporal anomalies exhibit a combination of temporal and spatial anomaly attributes. These are rare but difficult to detect

Table 2: Summary of Data Anomaly Types

values, determined by the mean and variance of the expected data set, can not detect temporal, spatial or spatio-temporal. Simple rule-based anomaly detection can be very effective if the data is well behaved and the rules are set appropriately. However, if the scope of analyzed data is expended even this simple approach can detect Temporal and spatial anomalies.

Temporal anomalies can be detected using rule-based statistical approaches if the analysis is performed using a sliding window over a single data stream. The chi-square test performed over a sliding window, outlined in Ngai et al. [41], is able to detect temporal anomalies provided the length of the window and the user specified acceptable chi-square test level is configured appropriately for the application and the volatility of the data.

Spatial anomalies can be detected using rule-based statistical approaches if the analysis is performed over multiple time synchronized data streams. The statistical inference approach outlined by Panda et al. [42] uses a voting system based on the calculated mean and variance of a set of neighboring data streams to detect a spatial anomaly.

Spatio-temporal anomalies cannot be detected using rule-based statistical approaches. This is do to the high dimensionality of data which prevents detection. Rule-based detection approaches are statistical inference techniques which cannot adapt to changing ranges, which are very common in long-term sensor network applications.

4.2.2. Estimation-based Detection

Estimation-based statistical approaches, which utilize a probability distribution models of the data to detect anomalous values can detect temporal

and spatial anomalies. An example is the detection technique utilizing the ARIMA model proposed by Fang et al. [39]. Temporal anomalies can be detected using the auto-regression, integration, and the moving average terms from a single data stream. The new data from that stream are compared with the ARIMA model to determine if the new data is anomalous. Spatial anomalies can be detected by maintaining multiple ARIMA model for all neighboring data streams. The new data is compared with the ARIMA models of neighboring nodes to detect a spatial anomalies. Similarly, other detection techniques compare new data with a known pdf or kernel density estimate of a single or neighboring data streams. The limitation of these techniques is how well the data fits the probability distribution. Estimation-based detection, like the rule-based detection, cannot detect spatio-temporal anomalies due to the high dimensionality of data. Unlike rule-based detection approaches, estimation-based approaches can adapt to changing ranges of long-term sensor network applications.

4.2.3. Learning-based Detection

Learning-based statistical approaches which utilize various data mining clustering and classification algorithms to group data for comparison have very high detection rates of all three types of anomalies [38]. For example, temporal and spatial anomalies can be detected using the neural network based ESN technique described by Chang et al. [44] and the Bayesian space-time model proposed by Ni et al. [48]. In both cases the network model predicts a data value which is used for comparison. The network model captures the time varying and spatial variation of the data. Another example of more detailed temporal anomaly detection approach is the MSPCA presented by Xie et al. [51]. This technique can detect gradual temporal anomalies such as calibration drift and persistent anomalies with different time-frequency features. Spatio-temporal anomalies can be detected using the PCA detection approach proposed by Chitradevi et al. [50]. This technique uses PCA to reduce the dimensionality of a data through spectral decomposition. The decomposed data is used to determine the normal behavior in a data set.

4.2.4. Behavioral Detection

Many behavioral approaches utilize the statistical anomaly detection approaches classified above to analyze system behavior rather then sensor or data input. Modi et al. [54] and Buttán et al. [4] describe techniques which use rule-based, estimation-based, and learning-based to compare system states

Cleansing Approaches	Detectable Anomaly
Rule-Based	Temporal anomalies, if analysis is preformed over a sliding window of recent values. Spatial anomalies, if analysis is conducted using data from neighboring nodes. Spatio-temporal anomalies can not be detected due to dimensionality of data.
Estimation-Based	Temporal and Spatial anomalies, if an appropriate probability distribution is used. Spatio-temporal anomalies can not be detected due to dimensionality of data.
Learning-Based	Temporal, Spatial and Spatio-temporal anomalies, depending on learning or clustering model used.
Behavior-Based	Not applicable, behavioral detection uses the statistical techniques to analyze system logs or system models of the behavior

Table 3: Classification of Cleansing Approach Based on Type of Anomaly

and detect normal and anomalous behavior. For example, the processor usage of various nodes over time can be compared with mean and variance rules or probability distributions of normal behavior to detect anomalies.

4.3. Static Elements of ITS Infrastructure

In this section the static elements of the example ITS system described in Section 2 will be used to demonstrate where in a CPS each types of data anomalies may be encountered dictating the cleansing technique.

4.3.1. Road Side Units

Road side units function as sensor and database nodes, which are queried by control elements in the transportation system and passing vehicles. This makes it essential for clean data to be in the database. The data anomalies likely to be encountered in an RSU are temporal and spatial. A temporal anomaly would be a dramatic change in traffic speed or congestion. This anomaly could be the result of an accident or a faulty sensor. Spatial anomalies would result from multiple RSU communicating and storing duplicated data. This stored information would be compared to detect spatial anomalies collaboratively among neighboring RSU and traffic control systems. How-

ever, RSU do not have a wide enough perspective to detect spatial-temporal. This would require collecting data from many RSU in a large physical region.

4.3.2. Traffic signals and Dynamic Lanes

Traffic signals and dynamic lanes have the same functionality of an RSU, including multiple sensors and storage databases, with the addition of a control system. This makes it essential for clean data to be available to the intelligent control system. The data anomalies likely to be encountered are similar to those of the RSU: temporal and spatial. The safety focus of traffic control require more rigorous detection of these anomalies to ensure safe operation. Spatial-temporal anomalies would not be detectable because of the limited perspective of the environment.

4.3.3. Central Traffic Management

Central traffic management can be divided into the autonomous control elements and semi-autonomous control elements. Autonomous control elements would control multiple traffic signals and dynamic lanes for a stretch of road or subdivision. This perspective would necessitate detection of basic spatial-temporal anomalies in addition to temporal and spatial anomalies. Detection and cleansing of spatial-temporal anomalies would be required for detecting trends in failure to better adapt control decisions. An example of this would be a storm or flooding during rush hour causing congestion as well as RSU, traffic light failures from a power outage. As the storm progressed across the area, spatial-temporal anomaly detection would be required to detect failures. The autonomous nature and immediacy of control decisions would limit the complexity of the spatio-temporal anomaly detection.

Semi-autonomous control elements would be at a higher level with a human in the loop. At this level, a wide perspective of the road networks and high level control would necessitate more advanced spatial-temporal anomaly detection. The responsibility of this type of control element would be to detect trends in order to inform specialists. This type of detection is essential for human intervention in the autonomous control and for longer term control decisions and modifications as a part of the design cycle.

4.4. Mobile Elements of ITS Vehicle

In this section, the mobile elements of the example ITS system described in Section 2 will be used to demonstrate where different types of data anomalies may be encountered, as well as the appropriate cleansing techniques for those anomalies.

ITS Element	Type of Data Anomaly
Road Side Unit	Requires detection of temporal and spatial anomalies to ensure that clean data is in database to be used by control elements.
Traffic Control	Requires detection of temporal and spatial anomalies to make accurate control decisions.
Central Traffic Management	Autonomous control elements require basic spatio-temporal anomaly detection in addition to temporal and spatial anomaly detection. Semi-autonomous control elements require rigorous spatio-temporal anomaly to detect trends for human in the loop control.

Table 4: Data Anomalies for ITS Static Infrastructure

4.4.1. Vehicle Operation

Nodes supporting vehicle operation may experience both spatial and temporal anomalies. From the temporal side, sensors can record values from sensors and identify when values suddenly change. For example, a temperature sensor could track the last three temperature readings, so that it can compare new readings with the average so far. Values that lie outside of the expected range could be considered outliers.

As for spatial anomalies, a vehicle may rely on several sensors to ascertain the actual state of a sensed environment. For example, on a cold morning, a vehicle may use information from a thermal sensor and four tire pressure sensors to decide whether low pressure in four tires is a result of the weather change or a result of a series of leaks.

4.4.2. Navigation

Anomalies at the navigation level impact the ability of navigation equipment to offer useful advice. Errors in GPS data are extremely common. Navigation equipment should be able to detect temporal errors, where a GPS sensors reports sudden, distant changes in position. In V2V environments, vehicles may collaborate to improve their location accuracy. If surrounding vehicles indicate their location differs, spatial anomalies can be detected and mitigated. Although spatio-temporal anomalies may be possible for navigation, it is less likely to occur and not worth spending resources to identify or mitigate.

ITS Element	Type of Data Anomaly
Vehicle Operation	Temporal anomalies are possible and detectable in data sensed by various on-board sensors. Spatial anomalies may also be possible in cases where data from several of the same type of sensor may be available.
Navigation	Temporal anomalies in GPS data can be detected and handled. Spatial anomalies are possible depending on the availability of location data from neighboring vehicles.
Driver	Temporal anomalies may arise from sensors assisting automated driving mechanisms, e.g. collision avoidance systems. Spatial anomalies may occur when platoons of autonomous vehicles collaborate to make driving decisions.

Table 5: Data Anomalies for ITS Mobile Infrastructure

4.4.3. Driver

Anomalies at the driver level impact the act of driving. Temporal anomalies may be experienced by systems that have direct control over a vehicle. For example, anomalies from proximity sensors should be detected to allow collision avoidance systems to make timely decisions about steering or braking. Spatial anomalies may also appear as fully autonomous vehicles sense their surroundings and collaborate on driving decisions. Platoons of autonomous vehicles may communicate to make various decisions including braking or lane changing.

5. Classification of Techniques Based on Requirements and Constraints

In this section we will classify the data cleansing techniques presented in Section 3 based on constraints of various elements in the system. Using this classification, we will then discuss where in an ITS each cleansing technique would be most appropriate.

5.1. Types of System Constraints

The hardware constraints of a networked system can be evaluated from two perspectives. The first perspective is from the hardware resources of

the nodes. Nodes with limited computational resources, including mobile devices, may not capable of executing the same algorithms as more powerful nodes. These nodes may lack the memory or disk space required to execute the algorithm, or they may lack the CPU speeds required to obtain a result in a reasonable amount of time. Additionally, some nodes may have constrained energy resources, which further limits computational ability. Computationally intensive algorithms increase energy consumption and may deplete energy resources at a faster rate.

The second perspective is from the constraints induced by the network. Low-bandwidth or error-prone networks increase the time and energy spent transmitting data from one node to another. In addition, data corruption may occur as a result of communication over a network. The cleansing technique chosen to inspect data received over a network may vary depending on the nature of the network.

5.1.1. Hardware Resource Constraints

Vyas et al. [60] discuss the constraints induced by various hardware architectures of control systems and sensor processing. Evaluating hardware architectures as it pertains to reading sensors, precision in executing a control algorithm, and the time to execute a control algorithm and perform an action with an actuator.

The authors propose an architecture where Sensor Processing Units (SPU) are able to perform some computations autonomously and communicate sensor data to a primary processor for more complex tasks. The SPU are configured to continuously execute user-defined functions, which can read data from sensors and perform small computations. As required, the primary CPU on the sensor node can retrieve output from the SPU for use in other tasks. The abilities of SPU are limited, with respect to the primary CPU on a sensor node. Simple tasks, including reading sensor data and performing basic data fusion, can be accomplished by an SPU. However, more complicated tasks require the intervention of the primary CPU.

For the benefit of devices with limited energy resources, many CPUs support a low-power mode. When the primary CPU is not required, a sensor node could enter a low-power mode to conserve energy. When necessary, SPU can issue an interrupt to the primary CPU, so that the CPU can perform computations that the sensor device cannot do alone.

Low-power mode may reduce the energy consumption of a sensor node, leading to a longer battery life, but complex data cleansing algorithms run-

ning on the primary CPU of a node will still impact the energy resources of that node. With or without the availability of a low-power mode, one may prefer less demanding data cleansing techniques to improve the battery life of sensor nodes. If the data cleansing technique is simple enough, it may be performed by the SPU; further allowing the primary CPU to remain in a low power state.

5.1.2. Network Induced Constraints

Zhang et al. [61] discuss the constraints induced by the network in a networked control system. A networked control system can be small scale such as a modern vehicle or large scale such as a metropolitan power grid. In both cases, sensors, actuators, and control decision nodes are networked together to create the control system. The network constraints discussed are time delay, packet losses, time-varying transmission intervals, multiple node access issues, and data quantization error. Though these constraints may factor into one's choice of data cleansing technique for a specific application, they are more detailed than required for our discussion on resource constraints for data cleansing techniques. The constraints of the system used for classification of cleansing techniques will be are storage, processing, communication, energy consumption.

5.2. Classification of Techniques

The techniques presented in Section 3 vary in terms of hardware and communication requirements. Table 7 contains a summary of the cleansing techniques classified by detectable data type.

5.2.1. Rule-based Detection

Given their simplicity, rule-based cleansing techniques are appropriate to execute on resource constrained devices. Anomaly detection rules are pre-defined. It is simply a matter of applying those rules to data items to determine whether items fit expectations of data or if they are anomalous. Additionally, the detection of anomalous data is based solely on the set of rules and does not rely on building models of data. With respect to other techniques, rule-based approaches should require fewer CPU and storage resources. As a result, energy consumption may be lower than techniques requiring more complex algorithms. However, as the number and complexity

Type of Resource Constraints	Description
Processing Constraints	Constraints on the execution time and processing power required to achieve an appropriate execution time for the cleansing technique.
Storage Constraints	Constraints on the storage resources and the size of database needed to store the relevant historical data required to execute the cleansing technique.
Communication Constraints	Constraints on the networking requirement. Specifically, these constrains refer to the availability of data from other system components as required by the cleansing technique.
Energy Constraints	Constraints on the energy that may be consumed by data cleansing techniques. Both processing and communication tasks may contribute to the overall energy consumption of a node.

Table 6: Summary of Resource Constraints

of anomaly detection rules grows, storage, processing, or energy constraints may become problematic.

Communication and energy overhead may increase drastically if detection rules depend on values from neighboring nodes. Raw data collected from neighboring nodes may be used as part of the anomaly detection process. However, rule-based techniques do not maintain models of data that is considered normal. As a result, nodes attempting to identify anomalies based on values sensed by neighbors may incur high communication costs. Matching data aggregated from neighbors is computationally inexpensive, but gathering that data may require significant communication overhead as well as increased energy consumption.

The history maintained by a node may also impact the resource requirements for rule-based cleansing techniques. Nodes that store a long history of raw data for rule-based matching require more storage than those that work with raw data live. This issue is exacerbated by rules that require storing raw data from neighboring nodes. In addition to storage requirements, a longer history of raw data requires increased processing time to arrive at a result and may increase energy consumption as well.

5.2.2. Estimation-based Detection

Estimation-based techniques require different levels of processing and storage resources to use and store the statistical models that represent normal data. Storage requirements vary depending on the model. Some models can be represented with a few floating-point parameters (e.g., Gaussian distribution), while others increase in size as they increase in detail (e.g. a histogram). Depending on the cleansing method, extra storage resources may be required to keep several models for data retrieved by local sensors. Some methods may also require storage of models for values retrieved from neighboring nodes as well.

Depending on the complexity of models, more processing resources may be required to determine whether or not data items fit within bounds that models define as normal.

Unlike rule-based techniques, estimation-based techniques do not necessarily need to collect raw data from neighbors to decide whether data items are anomalous. Instead of relying on raw data from neighbors, nodes can store a model for each neighbor that represents the normal values for that neighbor. If the model changes, a node can update its neighbors, so that they may update their stored models accordingly. Periodic model updates

should incur less communication overhead than raw data aggregation as well as potentially lower energy consumption.

5.2.3. Learning-based Detection

Like rule-based techniques, learning-based techniques detect anomalies in raw data. However, they require additional overhead that is not required of rule-based or estimation-based techniques. Learning-based techniques do not require pre-defined rules nor do they assume the raw data will fit particular models. Instead, the model of normal data is learned, which leads to highly accurate identification of data anomalies.

This accuracy comes at a cost. Algorithms for learning models can be computationally expensive, increasing processing time and storage while also potentially increasing power consumption. In addition to the resources required to execute learning algorithms, it takes time for the learned model to become accurate enough to be usable. A node's model can be adjusted as it receives new information from its sensors or its neighbors. By including data from neighboring nodes in the learning process, a node may learn its model more quickly. It may also be possible to offload the learning process on more powerful hardware, but this comes at the cost of communication.

Another way to reduce processing overhead is to build a partial model of normal data ahead of time using historically recorded data. However, building a model ahead of time requires making assumptions about data that will be collected. There may be a trade-off between building a flexible, highly accurate model from scratch and reducing the time required to build a useful model.

Once a model is learned, using it to identify anomalies is fast. However, continuing to update the model with the learning algorithms requires additional processing time.

5.2.4. Behavior-based Detection

Behavior-based detection relies on inspecting previous system states to identify anomalous states. Maintaining historical data incurs larger storage costs as the size of the history grows. However, behavior-based techniques are most applicable at a higher level in a CPS, where nodes are more likely to have reasonable computational and storage resources. These nodes aggregate information about system state from other components in the system. The information retrieved from other system components helps build a system-level view at the cost of processing and communication.

Cleansing Approach	Resource Constraints
Rule-Based	Lower processing and storage requirements, when rules are simple. Higher communication costs if the rules rely on data gathered from neighboring nodes. Nodes do not develop models and depend on raw data for anomaly detection.
Estimation-Based	Requires moderate processing and storage resources to detect anomalies and store statistical models. Communication requirements are lower, however, as statistical models of values seen by neighbors can be stored and reused, instead of relying on raw data.
Learning-Based	More complex; require more processing and storage resources. Though models can be stored compactly, large quantities of data and extra processing time are required to develop highly effective models.
Behavior-Based	Requires data from other nodes to capture the state of the system at the expense of additional communication. Historical data can be stored for future reference.

Table 7: Classification of Cleansing Approach Based on System Constraints

5.3. Static Elements of ITS Infrastructure

In this section the static elements of the example ITS system described in Section 2 will be used to demonstrate where in a CPS the hardware constraints would dictate the cleansing technique.

5.3.1. Road Side Units

Road Side Units are deployed throughout the network making cost the limiting factor. The sensor database nature of an RSU would require a moderate sized storage to facilitate raw data storage of local and neighboring sensor data. However, the data retention policy of the system would dictate the size of storage. The communication requirement of an RSU is very high to facility vehicle and control element access to the data. The processing requirement of an RSU is very low as its primary function is sensor, storage and communication. Energy efficiency is a concern as installations may be solar/batteries powered. The high communication requirement would require limiting the processing power of the node to meet energy limitations.

5.3.2. Traffic signals and Dynamic Lanes

Traffic signals and dynamic lanes have less of a cost limitation the RSU because of the safety critical aspects of their operation. The storage requirements of traffic signals and dynamic lanes would be similar to RSU. The processing and communication requirements of traffic signals and dynamic lanes, however, would be higher then an RSU. The control algorithms used by traffic signals and dynamic lanes require processing and communication to collaboratively make decisions. The short control loop necessitates faster processing. Energy efficiency is not a limitation of these control elements because traffic signals utilize infrastructure power with a battery backup in case of power failure.

5.3.3. Central Traffic Management

Central traffic management elements both autonomous and semi-autonomous control elements have similar system constraints. The high level nature of these nodes means cost is less of a limiting factor. These control elements will require the most processing power and storage. The control algorithms in a central traffic management system would utilize the most data both current and historical. Communication requirement will be dependent on the amount of raw data collected by the node. The communication requirement would be limited if only aggregated data was collected and database queries

ITS Element	Hardware Constraints
Road Side Unit	Highly constrained due to deployment and cost. Requires moderate storage, low processing, and high communication and energy efficiency.
Traffic Control	Less limited by cost due to safety critical nature of control. Storage and communication similar to RSU. Requires higher processing to execute control algorithm. Energy Efficiency not an issue.
Central Traffic Management	Less limited by cost as these are large investments. Very high storage storage and processing to execute control algorithm. Communication is dependent on granularity of data collected. Energy Efficiency not an issue.

Table 8: Resource Constraints for ITS Static Infrastructure

were execute in the sensor database nodes. Energy efficiency is not really an issue for central traffic management aside from the cost associated with a higher energy bill.

5.4. Mobile Elements of ITS Vehicle

5.4.1. Vehicle Operation

Sensors for vehicle operation are typically embedded systems and are not as computationally sophisticated as other components of the vehicle. Although their processing power may be lower, these components should still be able to clean data items very quickly. Simpler data cleansing techniques, like rule-based techniques, can cleanse data to some degree with short execution times without requiring more advanced CPUs. Communication for these elements is short distance, due to the proximity of the devices.

5.4.2. Navigation

Navigation equipment tends to be more capable than the embedded devices responsible for sensing vehicle state. These devices must be capable of interacting with a human user, sometimes approaching a PC-level user

ITS Element	Resource Constraints
Vehicle Operation	Constrained processing and storage resources. Limited, short rage communication. Depending on location, limited energy resources.
Navigation	Less constrained, though still limited, processing and storage. More capable communication for V2V or V2I.
Driver	Less constrained, though still limited, processing. Requires rapid response for safety-critical decisions. May include heavy communication load for fully autonomous vehicle coordination.

Table 9: Resource Constraints for ITS Mobile Infrastructure

experience. They are designed with a fair amount of disk space and reasonably fast processors. In applications where V2V communication is required, communication may be heavy.

Because of their access to more capable computational resources, navigation equipment may perform more sophisticated data cleansing techniques. Learning-based techniques can be used to detect anomalies in GPS data [62], which can be performed by navigation equipment.

5.4.3. Driver

Control decisions that affect driving should be based on accurate information and action should happen quickly. To meet these goals, data anomalies should be identified quickly to permit a fast control response. Considering the safety implications of systems like collision avoidance, it makes sense that manufacturers would invest in more powerful equipment to process decisions. Higher performance processors and more storage enable more advanced cleansing techniques, like behavior-based and learning-based.

For fully autonomous vehicles, communication may be heavy as well. The vehicle may consider information gathered from neighboring vehicles when preparing to make a control decision.

6. Classification of Techniques Based on Type of Data

In this section we will classify the data cleansing techniques presented in Section 3 based on the type of data that can be cleansed by each approach. Using this classification, we will then discuss where in an ITS each cleansing technique would be most appropriate.

6.1. Types of Data

The types of data that will used for classification are numerical and categorical data [63]. Statistical analysis, which data mining is rooted in, also recognizes additional data types including binary, ordinal, binomial, count, additive, and multiplicative. These statistical data types can be mapped to numerical or categorical data types because their distinguishing characteristics are not relevant to a control system in a CPS.

Numerical data types are any kind of quantitative data including any data that can have a number associated with it suitable for ranking. Numerical data can be discrete or continuous values describing absolute or relative measurements. Discrete data represents items that can be counted such as number of cars waiting at a stop light or the number of lanes that are open in a tunnel. A computer system would store this type of data as integer values. Continuous data represents measurement data such as the average speed of vehicles on a certain road. A computer system would store this type of data as floating point values. Numerical data is the most common type of data in a control system.

Categorical data types are qualitative in nature. Categorical data is data that represents characteristics such as the status of a road segment (open, closed) or an identification tag such as a license plate number. This type of data also includes binary or true/false data. System states are considered categorical data, for example, the status of a traffic light (green, red, yellow). In a computer system this type of data can be stored as strings or as enumerations.

6.2. Classification of Techniques

A variety of the techniques presented in Section 3 can be employed to analyze and detect anomalies in each of these types of data. Table 10 contains a summary of the cleansing techniques classified by detectable data type.

6.2.1. Rule and Estimation-based Detection

Rule-based and estimation-based detection approaches are similar enough to consider them as a single class from the data type point of view. Only the simplest rule-base detection approach is relevant for categorical data. This detection may include checking incoming data describing the state of a neighboring system is in fact an acceptable state, e.g. "blue" is not an state of a traffic light. Beyond this simple detection, anomalies in categorical data cannot be detected by rule- or estimation-based approaches. This is

because both require relative distances for comparison of data values. The mean and variance of a system system state is irrelevant. However, strictly categorical data without any order/ranking is not a realistic scenario for a CPS control system. Numerical data, however, is specifically what these detection techniques are designed for.

6.2.2. Learning-based Detection

Learning-based detection approach suffer from the same limitation as rule- and estimation-based approaches in terms of detecting anomalies in purely categorical data. The statistical nature of learning-based detection makes them very powerful for numerical data anomalies with capabilities beyond those of rule- and estimation-base approaches.

Learning-base detection has the added capability of detecting anomalies using a combination of categorical and numerical data. An example of this would be a data packet containing the state of a stop light, categorical data, and sensor measurements from traffic flow in the various directions around the light. Clustering approaches can be used to determine outlines based on this state and numerical data.

6.2.3. Behavior-based Detection

Behavior-based detection approaches are ideally suited for numerical, categorical, and the combination of categorical and numerical data. Numerical data extracted from system events can be analyzed using the statistical detection approaches above to detect anomalies. System events may include any type of logged activity such as network number and type of system calls, processor usage, and number of packets sent or received by a node.

However, behavioral detection techniques are ideal for categorical and combined numerical and categorical data. For example, stateful protocol analysis described by Liao et al. [53]. In this approach, the present system state and the sequence of historical system states are compared to an expected system operation profile. Numerical data can be used in addition to the sequence of system states to compare against a system model. An anomaly is detected when the system deviates from the behavior of the model.

6.3. Static Elements of ITS Infrastructure

In this section the static elements of the example ITS system described in Section 2 will be used to demonstrate where in a CPS each data type is encountered dictating the cleansing technique.

Cleansing Approaches	Applicable Data Type
Rule-Based & Estimation-Based	Ideal for purely numerical data. Unable to detect anomalies in categorical beyond the simplest validity check of a rule-based approach.
Learning-Based	Ideally suited for purely numerical data and combinations of categorical and numeric data. Unable to detect anomalies in purely categorical data.
Behavior-Based	Able to detect anomalies in numerical data extracted from system logs. Ideal for system state categorical data using stateful analysis.

Table 10: Classification of Cleansing Approach Based on Data Type

6.3.1. Road Side Units

The primary data type processed by road side units is numerical. Examples of sensors used by RSU are loop detector counts, traffic video, acoustical and in road magnetic sensors. In all cases, numerical data is extracted to capture the state of traffic in the sensed area.

Categorical data is also collected by RSU such as vehicle identification number or tag. These tags can be either transmitted to the RSU by the vehicle or gathered from the air such as bluetooth packets transmitted by a cellular phone in passing vehicles. Numerical data is extracted from this categorical data though collaboration between RSU, e.g. detecting speed by comparing time stamps of sequential RSU on a road segment. Numerical data is the primary data type collected and stored by an RSU.

6.3.2. Traffic signals and Dynamic Lanes

Traffic signals and dynamic lanes process the same data types as RSU with the addition of categorical data. The categorical data is the sequential control states of the controller and the controllers neighbors. Access to both numerical and categorical data facilitates more rigorous behavioral detection of the control system and its neighbors.

ITS Element	Type of Data
Road Side Unit	Primarily numerical data and numerical data extracted from categorical data.
Traffic Control	Primarily numerical data with the addition of controller state categorical data of neighboring controllers.
Central Traffic Management	Both numerical and categorical data collected from RSU and traffic signals and dynamic lanes.

Table 11: Data Types for ITS Static Infrastructure

6.3.3. Central Traffic Management

Central Traffic management process both controller state based categorical data and numerical data gathered from sensors. Autonomous and semi-autonomous central traffic management will process this type of data to detect anomalies in the traffic signals and dynamic lanes system states utilizing the numerical data provided by RSU.

6.4. Mobile Elements of ITS Vehicle

6.4.1. Vehicle Operation

Most sensor readings that facilitate vehicle operation are numerical. Various temperature and pressure sensors retrieve numerical values that can be scaled or otherwise modified. Categorical measurements do not apply to this area.

6.4.2. Navigation

Navigation equipment relies on numerical values as well. GPS data, as time, longitude, and latitude, can be retrieved and cleaned. Congestion data may be retrieved from I2V or V2V communication and can be represented as a measurement of the amount of traffic flowing through a road segment. Categorical measurements do not apply to this area.

6.4.3. Driver

Numerical sensor readings from proximity sensors may facilitate decisions made by autonomous and semi-autonomous vehicles. Additionally, categorical data may be available by inspecting messages received by other vehicles in the ITS. Consider a situation where a vehicle informs its neighbors of

ITS Element	Type of Data Anomaly
Vehicle Operation	Numerical
Navigation	Numerical
Driver	Numerical and categorical

Table 12: Data Types for ITS Mobile Infrastructure

its intention to change lanes. However, the vehicle never executes the lane change. This ITS equivalent of leaving ones blinker on could be detected as an anomaly, so that other vehicles can resume normal traffic behavior.

7. Classification of Techniques Based on Hierarchical Location

In this section we will classify the data cleansing techniques presented in Section 3 based on the hierarchical nature the system. Using this classification, we will then discuss where in an ITS each cleansing technique would be most appropriate.

7.1. Levels of System Hierarchy

Elements of an ITS can be categorized based on their location within the overall ITS hierarchy. Sensors are at the base of the hierarchy. They are capable of acquiring data and may have limited access to resources. The sensor actuator level of the hierarchy is above the sensor level. Sensor actuators are able to make some control decisions based on data acquired from local sensors. Next, the control level uses data collected by sensors and sensor actuators to make control decisions at a wider scope. Finally, the supervisor level encompasses human operator or supervisors who make decisions based on data retrieved from lower levels of the hierarchy. The retrieved data may be processed prior to use as the basis of decisions.

The sensor level of the hierarchy is the base, as it provides a foundation for the autonomous control decisions made by other levels in the hierarchy. Members of this level, such as road side units, are capable of collecting data from local sensors and disseminating values to others. Sensors are unable to make control decisions; they simply acquire and allow access to data items.

Because their role is simple, sensor level devices can use more simplistic data cleaning techniques. Rule-based techniques, for example, are simple to apply and can remove obviously erroneous data items. Estimation-based techniques may be applicable, as well, depending on the data being sensed.

At the sensor actuator level, local sensors are used to facilitate control decisions. For example, a traffic light might use in-road sensors or traffic cameras to control the flow of traffic at an intersection. Because these devices have some control over the infrastructure, they should be more capable machines with more resources than a sensor level device. With access to better resources than a sensor level device, sensor actuators can make use of more sophisticated data cleansing techniques.

For example, estimation-based techniques are more sophisticated than rule-based techniques, but they are still fast for applications that rely on short execution times. Learning-based techniques may also be feasible on sensor actuators. More capable hardware allows for more advanced algorithms (e.g., machine learning), which may be worth the additional costs to achieve more accurate anomaly detection. Sensor actuators may also make use of behavioral-based methods to verify its own behavior and check that its control loop leads to acceptable states.

Control level devices use data from sensors and sensor actuators to make control decisions for a complete road section that may have multiple lights or dynamic lanes. These devices have a larger scope of responsibility. As a result, more sophisticated methods are appropriate. Anomaly detection and data cleansing should be accurate to ensure decisions are made based on accurate data. Behavior-based and learning-based methods seem the most appropriate. Estimation-based methods may also fit, provided that there is a well developed, high level model for identifying anomalous data items. Control level devices have a wider scope of influence. As a result, they have more time to spend analyzing data and arrive at decisions.

The top of the hierarchy, the supervisor level, includes a central office with human operator/supervisor. Decisions made at this level affect the state of the system, as well as future design cycle decisions. Data may be interpreted by a human to aid in the design process. The most sophisticated cleansing techniques, i.e. learning-based and behavior-based, are appropriate for this level. Although data may be interpreted by a human, it is important for results to be accurate considering the broad impact that decisions have at this level.

7.2. Classification of Techniques

7.3. Static Elements of ITS Infrastructure

In this section the static elements of the example ITS system described in Section 2 will be used to demonstrate where in a CPS the hierarchical

location would dictate the cleansing technique.

7.3.1. Road Side Units

Road Side Units function at the sensor level without direct influence over the sensed area. The data transmitted by RSU can influence the decisions of intelligent vehicles utilizing I2V communication. However, this is not considered actuation in the environment because the vehicles are not directly controlled by the RSU.

7.3.2. Traffic signals and Dynamic Lanes

In addition to sensor capabilities for sensing traffic in a local area, traffic signals and dynamic lanes have direct control over a lane or a traffic interchange. This means that the traffic signals and dynamic lanes operate at the sensor actuator level.

7.3.3. Central Traffic Management

Central traffic management units operate at the control level and the supervisor level. The intermediate autonomous traffic management units operate at the control level. The autonomous traffic management units have control over traffic signals and dynamic lanes without directly controlling the state. An example of this would be alleviating a traffic congestion in an area by directing dynamic lanes and traffic signals that feed the congested area in order to slow incoming traffic. This level of traffic management does not directly control the changing of a traffic light. It has a higher level of control.

The highest level of control is the human-in-the-loop, semi-autonomous traffic management units that operate at the supervisor level. This level of operation is more focused on long term control and design cycle control rather than immediate control decisions. However, in an emergency the supervisory control can directly take over lower level control systems to mitigate a failure.

7.4. Mobile Elements of ITS Vehicle

7.4.1. Vehicle Operation

Sensors that support vehicle operations are at the sensor level. Tire pressure sensors, for example, can report the status of tires, but are unable to perform any action based on the state. Rain sensors and light sensors, on the other hand, are able to sense conditions and perform an action as a result, namely starting windshield wipers or turning on head lamps.

ITS Element	Hierarchical Location
Road Side Unit	Operate at the sensor level without direct influence over the state of the system.
Traffic Control	Operate at the sensor actuator level with the ability the sense the environment and influence the state of traffic on a road or at a traffic light.
Central Traffic Management	The autonomous traffic management units operate at the control level. The high level, human in the loop, semi-autonomous traffic management units operate at the supervisor level.

Table 13: Hierarchical Location for ITS Static Infrastructure

7.4.2. Navigation

In an autonomous vehicle, the navigation system plays an important role in driving decisions made by the vehicle. For this type of vehicle, the navigation system plays the role of a sensor, in that it determines its current location. However, the navigation recommendations are used to control the routes driven. In a way, the navigation system also acts as a sensor actuator. Its recommendations can be viewed as commands for driving to a particular destination.

Semi-autonomous vehicles may have collision avoidance or other driving control systems. In this case, the navigation system acts at the supervisor level. The driver makes the decision about changes to navigation based on the recommendations of the navigation system.

7.4.3. Driver

For autonomous vehicles, the human passenger has limited abilities to intervene with driving. Their ability to interact with a vehicle is strictly at a supervisor level. A vehicle may inform the user of diagnostic concerns, but control is ultimately left to the vehicle itself.

Semi-autonomous vehicles, on the other hand, permit humans to drive. Although some automated systems, such as collision avoidance, can momentarily intervene and take control, the driver is in control of the vehicle. In this case, the collision avoidance system acts at the controller level. It autonomously acts to avoid dangerous situations. Otherwise, the driver acts as

ITS Element	Location of Component
Vehicle Operation	Takes place at the sensor or sensor actuator level. Some devices simply gather data, while others are able to take action based on locally sensed information.
Navigation	Takes place at the sensor or supervisor level. While autonomous vehicles can use the navigation information to directly control driving behavior, other vehicles relay navigation data to a human driver, who ultimately decides on the chosen route.
Driver	Takes place at the controller or supervisor level. Autonomous vehicles use sensor data to directly control the path of the vehicle, while other vehicles rely on a driver to interpret and act on sensed information.

Table 14: Hierarchical Location for ITS Mobile Infrastructure

a supervisor. They are able to interpret results and take appropriate action behind the wheel.

8. Conclusion

Modern Critical Infrastructure Cyber-Physical Systems are designed to meet very high performance standards. This is only achievable using sophisticated intelligent autonomous control systems which are extremely data dependent. Therefore, it is essential to provide accurate real-time data to the control system. When these systems are deployed in extremely unpredictable environments it is essential that data cleansing is used to minimize the potential failures that would result from control systems processing erroneous data.

In this survey we discussed the sources of corrupted data and various general purpose data cleansing techniques including detection and mitigation methods. We classify these techniques based on their applicability to CPS specifically discussing the anomalies which can be detected, hardware constraints of various nodes within a CPS, type of data being processed, and hierarchical location within a CPS. Understanding the limitations of various data cleansing technique is essential to designing fault tolerant and survivable CPS. We are working to model the extent to which data errors can propagate with in a CPS. Specifically looking at the effect of data cleansing within a

system. This information will be essential to designing robust CPS.

Glossary

CPS	Critical Infrastructure Cyber-Physical Systems. 3, 47
ESN	Echo State Network. 16, 47
I2V	Infrastructure to Vehicle Communication. 7, 47
IDS	Intrusion Detection Systems. 18, 47
ITS	Intelligent Transportation Systems. 4, 47
MSPCA	Multi-Scale Principal Component Analysis. 17, 47
PCA	Principal Component Analysis. 17, 47
pdf	Probability Distribution Function. 14, 47
RSU	Road Side Unit. 9, 47
TQDM	Total Quality Data Management. 12, 47
V2I	Vehicle to Infrastructure Communication. 7, 47
V2V	Vehicle to Vehicle Communication. 7, 47
V2X	Vehicle to Everything Communication. 7, 47

References

[1] P. Derler, E. Lee, A. Vincentelli, Modeling cyber physical systems, Proceedings of the IEEE 100 (1) (2012) 13–28.

[2] D. E. Bakken, A. Bose, C. H. Hauser, D. E. Whitehead, G. C. Zweigle, Smart generation and transmission with coherent, real-time data, Proceedings of the IEEE 99 (6) (2011) 928–951.

[3] B. A. Hoverstad, A. Tidemann, H. Langseth, Effects of data cleansing on load prediction algorithms, in: Computational Intelligence Applications In Smart Grid (CIASG), 2013 IEEE Symposium on, IEEE, 2013, pp. 93–100.

[4] L. Buttyán, D. Gessner, A. Hessler, P. Langendoerfer, Application of wireless sensor networks in critical infrastructure protection: Challenges and design options [security and privacy in emerging wireless networks], IEEE Wireless Communications 17 (5) (2010) 44–49.

[5] A. A. Kirilenko, A. W. Lo, Moore's law versus murphy's law: Algorithmic trading and its discontents, The Journal of Economic Perspectives (2013) 51–72.

[6] B. Miller, D. Rowe, A survey of SCADA and critical infrastructure incidents, in: Proceedings of the 1st Annual Conference on Research in Information Technology, RIIT '12, ACM, New York, NY, USA, 2012, pp. 51–56.

[7] A. Berizzi, The Italian 2003 blackout, in: IEEE Power Engineering Society General Meeting, 2004., IEEE, 2004, pp. 1673–1679.

[8] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, S. Havlin, Catastrophic cascade of failures in interdependent networks, Nature 464 (7291) (2010) 1025–1028.

[9] M. Woodard, S. S. Sarvestani, A. R. Hurson, A survey of research on data corruption in cyber–physical critical infrastructure systems, Advances in Computers (2015).

[10] I. Solutions, Delivering intelligent transport systems driving integration and innovation, Tech. rep., IBM Corperation (2007).

[11] R. R. Rajkumar, I. Lee, L. Sha, J. Stankovic, Cyber-physical systems: the next computing revolution, in: Proceedings of the 47th Design Automation Conference, ACM, 2010, pp. 731–736.

[12] M. Amadeo, C. Campolo, A. Molinaro, Information-centric networking for connected vehicles: a survey and future perspectives, Communications Magazine, IEEE 54 (2) (2016) 98–104.

[13] K. N. Qureshi, A. H. Abdullah, A survey on intelligent transportation systems, Middle-East Journal of Scientific Research 15 (5) (2013) 629–642.

[14] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Communications Magazine 40 (8) (2002) 102–114.

[15] A. R. Hurson, Y. Jiao, Database system architecture–A walk through time: From centralized platform to mobile computing–keynote address, in: Advanced Distributed Systems, Springer, 2005, pp. 1–9.

[16] P. Bonnet, J. Gehrke, P. Seshadri, Towards sensor database systems, in: Mobile Data Management, Springer, 2001, pp. 3–14.

[17] N. Tsiftes, A. Dunkels, A database in every sensor, in: Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems, SenSys '11, ACM, New York, NY, USA, 2011, pp. 316–332.

[18] G. Dimitrakopoulos, P. Demestichas, Intelligent transportation systems, Vehicular Technology Magazine, IEEE 5 (1) (2010) 77–84.

[19] L. Li, D. Wen, D. Yao, A survey of traffic control with vehicular communications, Intelligent Transportation Systems, IEEE Transactions on 15 (1) (2014) 425–432.

[20] V. Milanes, J. Villagra, J. Godoy, J. Simo, J. Perez, E. Onieva, An intelligent v2i-based traffic management system, Intelligent Transportation Systems, IEEE Transactions on 13 (1) (2012) 49–58.

[21] E. Rahm, H. H. Do, Data cleaning: Problems and current approaches, IEEE Data Eng. Bull. 23 (4) (2000) 3–13.

[22] J. I. Maletic, A. Marcus, Data cleansing: Beyond integrity analysis., in: IQ, Citeseer, 2000, pp. 200–209.

[23] L. Bertossi, S. Kolahi, L. V. Lakshmanan, Data cleaning and query answering with matching dependencies and matching functions, Theory of Computing Systems 52 (3) (2013) 441–482.

[24] R. Y. Wang, A product perspective on total data quality management, Communications of the ACM 41 (2) (1998) 58–65.

[25] M. Dallachiesa, A. Ebaid, A. Eldawy, A. Elmagarmid, I. F. Ilyas, M. Ouzzani, N. Tang, Nadeef: a commodity data cleaning system, in: Proceedings of the 2013 ACM SIGMOD international conference on management of data, ACM, 2013, pp. 541–552.

[26] M. Budka, M. Eastwood, B. Gabrys, P. Kadlec, M. M. Salvador, S. Schwan, A. Tsakonas, I. Žliobaitė, From sensor readings to predictions: on the process of developing practical soft sensors, in: Advances in Intelligent Data Analysis XIII, Springer, 2014, pp. 49–60.

[27] A. Aviźienis, J. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, IEEE Transactions on-Dependable and Secure Computing 1 (1) (2004) 11–33.

[28] Y. Mo, T.-H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, B. Sinopoli, Cyber–physical security of a smart grid infrastructure, Proceedings of the IEEE 100 (1) (2012) 195–209.

[29] S. Amin, X. Litrico, S. S. Sastry, A. M. Bayen, Stealthy deception attacks on water SCADA systems, in: Proceedings of the 13th ACM international conference on Hybrid systems: computation and control, ACM, 2010, pp. 161–170.

[30] J. L. Cebula, L. R. Young, A taxonomy of operational cyber security risks, Tech. rep., DTIC Document (2010).

[31] C. C. Aggarwal, N. Ashish, A. Sheth, The Internet of things: A survey from the data-centric perspective, in: Managing and Mining Sensor Data, Springer, 2013, pp. 383–428.

[32] S. Rajasegarar, C. Leckie, M. Palaniswami, Anomaly detection in wireless sensor networks, IEEE Wireless Communications 15 (4) (2008) 34–40.

[33] S. Yin, G. Wang, H. R. Karimi, Data-driven design of robust fault detection system for wind turbines, Mechatronics 24 (4) (2014) 298–306.

[34] P. Freeman, P. Seiler, G. J. Balas, Air data system fault modeling and detection, Control Engineering Practice 21 (10) (2013) 1290–1301.

[35] L.-A. Tang, X. Yu, S. Kim, Q. Gu, J. Han, A. Leung, T. La Porta, Trustworthiness analysis of sensor data in cyber-physical systems, Journal of Computer and System Sciences 79 (3) (2013) 383–401.

[36] Y. Zhang, N. Meratnia, P. Havinga, Outlier detection techniques for wireless sensor networks: A survey, IEEE Communications Surveys & Tutorials 12 (2) (2010) 159–170.

[37] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Computing Surveys (CSUR) 41 (3) (2009) 15.

[38] L. Fang, S. Dobson, In-network sensor data modelling methods for fault detection, in: Evolving Ambient Intelligence, Springer, 2013, pp. 176–189.

[39] L. Fang, S. Dobson, Unifying sensor fault detection with energy conservation, in: Self-Organizing Systems, Springer, 2014, pp. 176–181.

[40] B. Bilir, A. Abur, Bad data processing when using the coupled measurement model and takahashi's sparse inverse method, in: Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2014 IEEE PES, IEEE, 2014, pp. 1–5.

[41] E.-H. Ngai, J. Liu, M. Lyu, On the intruder detection for sinkhole attack in wireless sensor networks, in: IEEE International Conference on Communications, 2006. ICC '06., Vol. 8, 2006, pp. 3383–3389.

[42] M. Panda, P. M. Khilar, An efficient fault detection algorithm in wireless sensor network, in: Contemporary Computing, Springer, 2011, pp. 279–288.

[43] S. Rajasegarar, C. Leckie, M. Palaniswami, J. C. Bezdek, Distributed anomaly detection in wireless sensor networks, in: 10th IEEE Singapore International Conference on Communication systems, 2006. ICCS 2006., IEEE, 2006, pp. 1–5.

[44] M. Chang, A. Terzis, P. Bonnet, Mote-based online anomaly detection using echo state networks, in: Distributed Computing in Sensor Systems, Springer, 2009, pp. 72–86.

[45] O. Obst, Distributed backpropagation-decorrelation learning, in: NIPS Workshop: Large-Scale Machine Learning: Parallelism and Massive Datasets, 2009.

[46] C. Mayfield, J. Neville, S. Prabhakar, Eracer: a database approach for statistical inference and data cleaning, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, 2010, pp. 75–86.

[47] X. R. Wang, J. T. Lizier, O. Obst, M. Prokopenko, P. Wang, Spatiotemporal anomaly detection in gas monitoring sensor networks (2008) 90–105.

[48] K. Ni, G. Pottie, Sensor network data fault detection with maximum a posteriori selection and bayesian modeling, ACM Transactions on Sensor Networks (TOSN) 8 (3) (2012) 23.

[49] J. W. Branch, C. Giannella, B. Szymanski, R. Wolff, H. Kargupta, In-network outlier detection in wireless sensor networks, Knowledge and information systems 34 (1) (2013) 23–54.

[50] N. Chitradevi, V. Palanisamy, K. Baskaran, U. B. Nisha, Outlier aware data aggregation in distributed wireless sensor network using robust principal component analysis, in: 2010 International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, 2010, pp. 1–9.

[51] X. Ying-xin, C. Xiang-guang, Z. Jun, Data fault detection for wireless sensor networks using multi-scale PCA method, in: 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011, IEEE, 2011, pp. 7035–7038.

[52] E. U. Warriach, T. A. Nguyen, M. Aiello, K. Tei, A hybrid fault detection approach for context-aware wireless sensor networks, in: IEEE 9th International Conference on Mobile Adhoc and Sensor Systems (MASS), 2012, IEEE, 2012, pp. 281–289.

[53] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: A comprehensive review, Journal of Network and Computer Applications 36 (1) (2013) 16–24.

[54] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, A survey of intrusion detection techniques in cloud, Journal of Network and Computer Applications 36 (1) (2013) 42–57.

[55] R. Mitchell, I.-R. Chen, A survey of intrusion detection techniques for cyber-physical systems, ACM Computing Surveys (CSUR) 46 (4) (2014) 55.

[56] M. Mezzanzanica, R. Boselli, M. Cesarini, F. Mercorio, A model-based approach for developing data cleansing solutions, Journal of Data and Information Quality (JDIQ) 5 (4) (2015) 13.

[57] T. Gschwandtner, W. Aigner, S. Miksch, J. Gärtner, S. Kriglstein, M. Pohl, N. Suchy, Timecleanser: A visual analytics approach for data cleansing of time-oriented data, in: Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business, ACM, 2014, p. 18.

[58] S. Gantayat, A. Misra, B. Panda, A study of incomplete data–a review, in: Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013, Springer, 2014, pp. 401–408.

[59] R. Jurdak, X. R. Wang, O. Obst, P. Valencia, Wireless sensor network anomalies: Diagnosis and detection strategies, in: Intelligence-Based Systems Engineering, Springer, 2011, pp. 309–325.

[60] S. Vyas, A. Gupte, C. D. Gill, R. K. Cytron, J. Zambreno, P. H. Jones, Hardware architectural support for control systems and sensor processing, ACM Trans. Embed. Comput. Syst. 13 (2) (2013) 16:1–16:25.

[61] L. Zhang, H. Gao, O. Kaynak, Network-induced constraints in networked control systems;a survey, IEEE Transactions on Industrial Informatics 9 (1) (2013) 403–416.

[62] J.-A. Ting, E. Theodorou, S. Schaal, Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, Ch. Learning an Outlier-Robust Kalman Filter, pp. 748–756.

[63] I. H. Witten, E. Frank, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2005.