IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 24, 2018, accepted October 12, 2018, date of publication October 22, 2018,
date of current version November 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2877097

Multivariate Sensor Data Analysis for Oil Refineries and Multi-mode Identification of System Behavior in Real-time

ATHAR KHODABAKHSH[1], (Student Member, IEEE), ISMAİL ARİ [1], (Member, IEEE), MUSTAFA BAKİR[2], AND ALİ OZER ERCAN[3], (Senior Member, IEEE)

[1]Computer Science Department, Özyeğin University, 34794 Istanbul, Turkey
[2]Process Improvement and Software Department, Tüpraş, 41790 Kocaeli, Turkey
[3]Electrical and Electronics Engineering Department, Özyeğin University, 34794 Istanbul, Turkey

Corresponding author: Athar Khodabakhsh (athar.khodabakhsh@ozu.edu.tr)

This work was supported by a grant from Turkish Petroleum Refineries Inc. (TUPRAS) R&D Center.

ABSTRACT Large-scale oil refineries are equipped with mission-critical heavy machinery (boilers, engines, turbines, and so on) and are continuously monitored by thousands of sensors for process efficiency, environmental safety, and predictive maintenance purposes. However, sensors themselves are also prone to errors and failure. The quality of data received from these sensors should be verified before being used in system modeling. There is a need for reliable methods and systems that can provide data validation and reconciliation in real-time with high accuracy. In this paper, we develop a novel method for real-time data validation, gross error detection and classification over multivariate sensor data streams. The validated and high-quality data obtained from these processes is used for pattern analysis and modeling of industrial plants. We obtain sensor data from the power and petrochemical plants of an oil refinery and analyze them using various time-series modeling and data mining techniques that we integrate into a complex event processing engine. Next, we study the computational performance implications of the proposed methods and uncover regimes where they are sustainable over fast streams of sensor data. Finally, we detect shifts among steady-states of data, which represent systems' multiple operating modes and identify the time when a model reconstruction is required using DBSCAN clustering algorithm.

INDEX TERMS Complex event processing, gross error classification, gross error detection, oil refinery, sensor data, stream data, system behavior.

I. INTRODUCTION

In an oil refinery, everything happens in big proportions: liquids flow in tons/hour rate, temperatures are measured in hundreds to thousands °C, and electricity is produced in megawatts. Thousands of people and millions of dollars are at stake every moment as one tiny malfunction or mistake in the system can cause serious damage to the entire plant and the workers, or generate losses in revenue. Thus, achieving continuous safety, process efficiency, long-term durability and planned (vs. unplanned) downtimes are among the main goals for industrial plant management. Due to mission-criticality of processes, oil & gas businesses have already implanted thousands of sensors inside and around their physical systems [1]. Raw sensor data continuously streams via distributed control systems (DCS) and supervisory control and data acquisition (SCADA) systems measuring temperature, pressure, flow rate, vibration, level, etc. of drills, turbines, boilers, pumps, compressors, and injectors. Another aspect in real-time system identification is updating the models according to the currently received pattern from stream data [2]. Since the systems are dynamic and the quality of models are dependent on both quality of data and system model, it is crucial to assess the current context of the system as the relations among model variables can change over time.

Achieving all these goals, necessitate to continuously monitor and verify the accuracy of measurements streaming in from numerous and various types of sensors placed all around the refinery. However, sensors are also prone to failures and measurement errors. At normal operation, sensor measurements are assumed to be noisy (*i.e.* to have random errors). Electro-mechanical effects such as malfunctioning, un-calibrated or broken sensors, and human factors also introduce systematic errors (*a.k.a.* ''gross errors'') to these measurements. Ability to differentiate sensor errors

from real system abnormalities is crucial for the safety of the plants. However, there is a lack of real-time industrial Data Reconciliation (DR) and Gross Error Detection (GED) methods, or data cleaning services, that can be used by oil refineries. To motivate and demonstrate the use of real-time DR-GED, we obtained time-series data from the power and petrochemical plants of a real refinery with approximately 11.5 million tons/year processing capacity [3].

Our contributions in this paper for DR-GED methods are as follows:

- First, we analyze multivariate data including flow, temperature, pressure using ARMA time-series modeling technique and generate synthetic datasets for ground truth tests.
- Second, we compare the accuracy of three GED models: an optimizer using Instantaneous Mass Balance constraint (IMB), Kalman Filter (KF), and Kalman Filter with Unity Gain constraint (KF-UG counterpart) over real and synthetic sensor data.
- Third, for the detected errors, we apply gross error classification (GEC) using a Complex Decision Tree (CDT), neural network (NN), and K-Nearest Neighbor (KNN) algorithms, and classify sensor errors into four types called Bias, Drift, Precision Degradation and Failure [4]. Applying GEC on top of GED is crucial for oil refineries, and we demonstrate that the accuracy of GED techniques can vary per error type. Each error type requires a relevant action on sensors: some error types are fixable whereas others aren't and need sensor replacement.
- Fourth, we integrate these trained DR-GED, GEC models into a complex event processing (CEP) engine and verify the accuracies and performances of the proposed techniques over real refinery datasets.
- Finally, we use validated data for analysis of changing system modes. The aim of operational mode analysis is to identify the time where operation shifts from one steady-state to another and a model reconstruction is required. The context shift is analyzed based on fluctuations in streaming sensor data using DBSCAN clustering.

Overall, we discuss selection and synergetic use of a set of analytical tools from different domains including data mining, statistics, and distributed systems to address challenges faced in petrochemical industry. The aim of this paper is not only detecting outliers and improving the accuracy of data, but also recognizing the state of physical sensors and industrial system and identifying the time of model reconstruction. We hope that our contributions for oil refineries will also contribute to Industry 4.0 and digital transformation efforts of other future factories [5], [6].

II. BACKGROUND AND RELATED WORK

Data validation and reconciliation (DVR) techniques were developed to improve data quality and satisfy plant models within the last few decades. The process model equations such as mass equilibrium and conservation laws were used to perform DR-GED. Data reconciliation is a mathematical model [7] that reduces inconsistency between measured data and physical model by reducing the effect of random errors on data. Reconciled estimates are expected to be more accurate and without (or at least smoothed) outliers. To accomplish accuracy improvement of data, outlier detection methods were developed and applied together as a companion technique to DR. Shcherbakov *et al.* [8] studied outlier detection and anomaly detection [9], [10] as data-driven approaches developed for identification of unexpected patterns in data and can be categorized into four types: distribution-based, distance-based, clustering-based and density-based [11] that are applicable to data directly. But, DR-GED techniques use underlying physical system model to satisfy constraints in addition to outlier detection. Approaches like statistical outlier detection are applicable where data follows a certain distribution. For model construction of a system fed with stream data, time-series analysis such as auto-regression, moving average and exponential smoothing are required to fit model, monitor and understand underlying forces of process model [12] in order to make future predictions and replace possible missing values [13]. Time-series models are applicable in a wide variety of sectors such as health-care [14], transportation [12], forecasting [15], and stock market. Error detection and classification such as ours provide a data quality improvement for better system modeling and isolation of faulty processes.

DR-GED applications have been used in chemical or petrochemical processes [16], [17] since their analysis quality has been known to directly improve the process performance and increase profits as well as safety [18]. Most prior studies focused on performing DR-GED offline using static samples of data collected from relatively old system logs. Over the past decades, the number of data resources such as sensors used in industrial facilities and Internet of Things (IoT) has risen dramatically. Yet, online data processing remains a challenge. Attempting to store these data first to analyze them later creates additional IT costs, unwanted delays to actionable information, and mishandling of threats or opportunities. Fortunately, there are now tools to process data on-the-fly as they move from DCS and SCADA systems to selected destinations. Neither relational databases nor distributed batch processing systems [19] alone are designed to cope with industrial data analytics. In sectors such as finance and mobile telecommunications, enterprises started employing CEP engines [20] for tracking Key Performance Indicators (KPI) in real-time or for carrying out rule-based alarm management. Nowadays, heavy industries also want to complete their ''digital transformation'' or Industry 4.0 journey [21] by extending their data architectures with real-time complex analytical [22], [23]) capabilities. However, data needs to be validated first in real-time, for the rest of the online analytical models to work accurately.

do Valle *et al.* [24] collected benchmarks for DR-GED issues introduced in the literature for mass and energy balance preservation. Zhang *et al.* [25] studied DR and parameter

estimation (DR-PE) for systems with multi-operating conditions and suggested a PCA-based steady-state detection technique extended with clustering to partition the data into different modes of operation, so that accurate reconciliation can be applied for each mode. They also addressed the DR-GED problem for dynamic systems using particle filters [26]. Guo *et al.* [27] proposed a systematic approach for DR of a thermal system using mass balance and improved data accuracy. Cai *et al.* [28] proposed a multi-source information fusion based fault diagnosis methodology for multiple-simultaneous faults using Bayesian networks that increases accuracy. Ruan *et al.* [29] used a symbolic representation of time-series data for reducing the volume while also extracting patterns.

Rafiee and Behrouzshad [17] studied DR-GED using material and energy conservation laws in natural gas processing. Jiang *et al.* [30] studied GED for data obtained from a coal-fired power plant. They modeled the system at steady-state and reconciled the data using the mass and energy balance equations. They employed statistical global test to detect gross errors and serial elimination technique to identify the error sources. Their steady-state technique compares to the basic Instantaneous Mass Balance (IMB) method described in this paper. We find that IMB and similar ''stateless'' techniques are less effective compared to modified Kalman filters that track system behavior when the system is not at steady-state. They also did not discuss applying GED on top of streaming data.

Since the industrial systems are dynamic it is crucial to assess the correct time to update the cyber model. For this purpose, Zhang *et al.* [31] proposed an incremental model tracking framework for quality-directed adaptive analysis named AQuA. Lughofer *et al.* [32] proposed an incremental rule splitting concept to autonomously deal with gradual drifts for local distributions. Zhu and Geng [33] proposed a ''multi-scenario'' parameter estimation for dynamic systems. Zheng *et al.* [25] proposed parameter estimation with multi-operating conditions for data reconciliation in steady-state. For handling drifts in data streams Shaker and Lughofer [34] proposed an adaptive forgetting factor depending on current intensity of drift in stream data. The aim is to identify the time where the state is shifting from one steady-state to another and a model reconstruction is triggered. In this paper, we detect the context shift of the system according to the error fluctuations of the trained DBSCAN clusters.

Our approach combines steady-state modeling with real-time model updates, operational mode identification, and data cleaning via error detection all at once.

The rest of the paper is organized as follows. Section 3 describes the cyber-physical systems and proposed data architecture for oil refineries. Section 4 details different GED methods including IMB, KF, and DREDGE and different GEC methods including CDT, NN, and KNN and describes DBSCAN analysis for recognizing the behavior of industrial system and identifying the time for model reconstruction. Section 5 discusses the experimental results, compares the accuracy and computational performance of the described methods for different error types and state analysis. Finally, Section 6 concludes the paper and discusses future work.

III. DESCRIPTION OF CYBER-PHYSICAL SYSTEMS IN OIL REFINERIES

Cyber-Physical Systems (CPS) are described by Rajkumar *et al.* [35] as ''physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core''. Industry 4.0 will be realized [36] by connecting CPS with Cloud via Internet of Things (IoT) and providing distributed, secure, intelligent analytical data services at the Edge or the Cloud [37], [38]. Oil & gas businesses have already implanted thousands of sensors inside and around their physical systems. Sensor data continuously streams in via DCS and SCADA systems measuring temperature, pressure, flow rate, *etc.* of drills, turbines, boilers, pumps, compressors, and injectors.

Figure 1 illustrates the the CPS and software architecture of our oil refinery. The major ''physical'' components are the power systems depicted in Figure 2 and the crude oil processing columns depicted in Figure 3; their ''cyber'' counterparts are composed of the sensors, servers and the data-based services that store and process the digital models. The power plant provides electricity to the rest of the refinery and has about 80 megawatts of generation capacity. There are 8 boilers with maximum flow capacities of 100 tons/hour, which turn hot water into super-heated and highly-pressurized vapor. The vapor output from every boiler is directly fed into a corresponding steam turbine with an alternator, which turns the thermo-kinetic energy into electrical energy. The flow rates of the inputs (hot water) and the outputs (vapor) of the boilers are measured by flow rate sensors (S-In1, S-Out1, etc.) and these measurements are fed into the models. Placing redundant (*i.e.* extra) sensors around the physical systems increases reliability and allows replacement of missing values that help to locate and classify errors or detect sensor failures.

Crude oil columns take the oil as input and deliver several by-products such as liquid propane gas, fuel oil, kerosene, diesel, and asphalt. A preflash unit reduces the pressure and provides the first vaporization, where the vapor goes to a debutanizer for distillation and the liquid mix goes to an atmospheric column for separation. On the digital side, we have servers and software for online and offline processing of received data. For offline data processing, we used the Hadoop Distributed framework [19] for providing the Extract, Transform and Load (ETL) services. This service gathers raw data from all sensor streams and presents them in a unified format. Using offline data, we can extract the steady-state models for physical systems and use this prior information to instantiate cyber models. Using online data, we can tune the system models dynamically and detect gross errors in real-time. We deployed DR-GED models and GEC

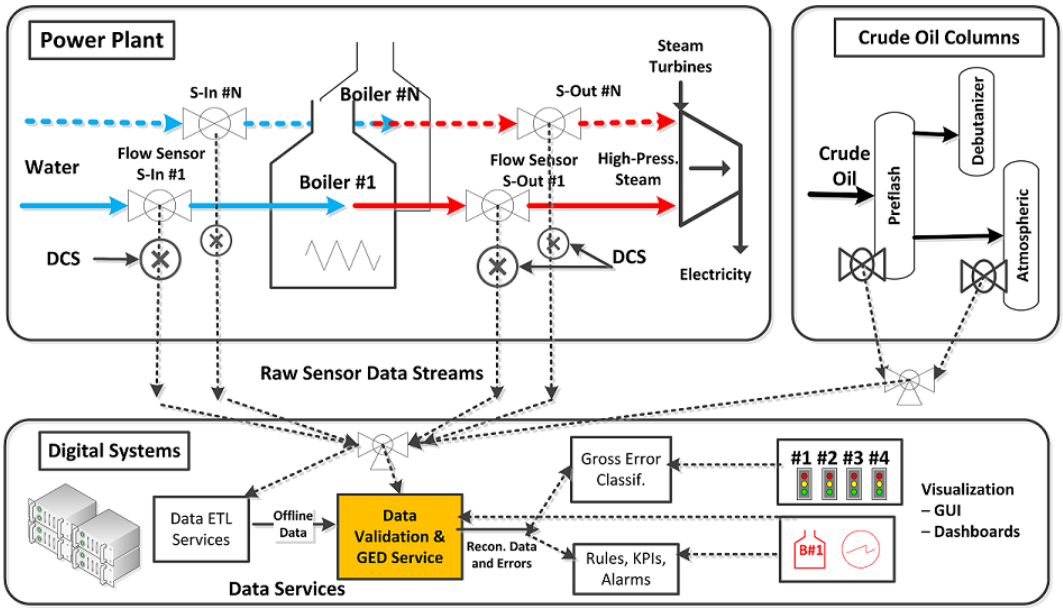

FIGURE 1. Illustration of the cyber-physical systems inside our oil refinery; high-level operation of the power plant, petrochemical plant for crude oil processing, and data stream processing services.

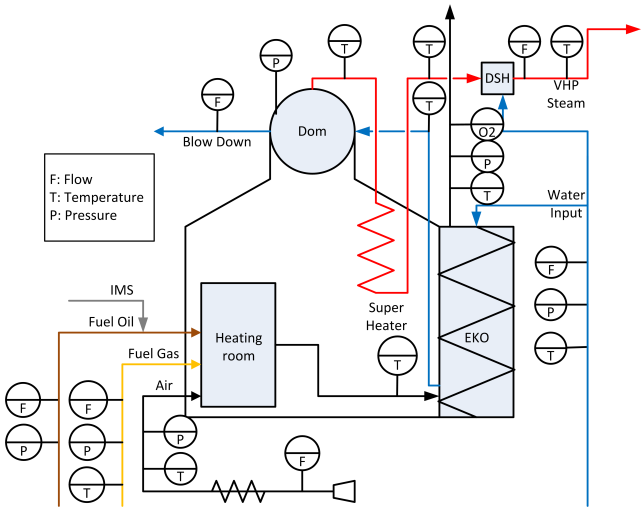

FIGURE 2. Illustration of a boiler power plant. Various types of sensors are implanted for real-time data collection. The boiler can change states among the desired stream pressure levels (low, high, very-high) or go automatically into heating, cooling, recycling, and condensing modes. VHP: Very-high power steam output; DSH: De-Super Heater.

algorithms inside a CEP engine. The benefits of using CEP engines for data stream analytics are at least three-fold:

- They can turn raw data into actionable information quickly, thus helping oil refineries catch critical issues to avoid losses in real-time.
- They can eliminate unwanted data early in the data pipeline, saving further CPU, memory, storage and energy costs.
- They can catch transient or emerging patterns, which never show up in an offline data mining analyses.

Considering all input-output lines and the different types of measurements (water and vapor flow rates, temperature, pressure, fuel oil and fuel gas flow rates), there are about 1,000 sensors in the power plant and 60,000 sensors in the entire oil refinery, where a sample of one month real data measured every minute is made available for academic use at OpenML datasets web site [39]. Our goal is to process all of this raw, streaming sensor data and create valuable data services for generating clean and reconciled data, detecting and classifying gross errors (*i.e.* avoid false positives - FP), and raising alerts when the system is malfunctioning (*i.e.* true positives - TP). The system can also be used to track a set of KPI for the entire plant and report results in dashboards if the performances are below or above pre-defined thresholds.

IV. METHODOLOGY FOR DREDGE

DR-GED requires a mathematical model to reduce inconsistency between measured data and industrial process model. In this study, we use two main models for data reconciliation: mass balance constraints for systems in steady-state and Kalman filters [40] for time-varying processes. We also propose DREDGE, which is a special Kalman Filter implementation extended with Unity Gain that incorporates system dynamics into mass balance constraints. After modeling system, we apply a distribution-based outlier detection approach on the estimated system model as GED method. All methods are implemented into a CEP engine.

On the other hand, industrial systems can have multiple operational modes and there can be shifts among them during daily operation. Accordingly, the models are required to be reconstructed and time-varying parameters updated, which explains the emergence of local and window-based stream online analysis to be more reliable than offline (or batch-data) analysis.

TABLE 1. Sample data for Y_t, input-output flow values (ton/h) of one of the boilers. The complete data spans 5 months worth of measurements from 12/2014 to 4/2015.

Water	DSH	Vapor
54.15951	0.737258	52.311
53.48025	0.74118	51.476
54.11722	0.740656	53.231
51.46956	0.742319	53.128
54.07272	0.738685	51.057

TABLE 2. Sample flow measurements for 17 sensors Y_t (ton/h) of the petrochemical system of Fig 3 and corresponding matrix A. The complete data spans 2 months worth of measurements from 08/2014 to 10/2014.

1	2	3	4	...	17
14136.54	34.91064	1530.896	12549.15	...	910.1204
14139.98	35.59276	1537.567	12557.17	...	909.0833
14122.35	35.73056	1533.281	12553.83	...	906.283
14113.85	36.53142	1529.905	12545.45	...	902.0441
14126.97	35.47277	1527.698	12555.25	...	909.6918

A. DATA VALIDATION AND GROSS ERROR DETECTION

1) STEADY-STATE AND INSTANTANEOUS MASS BALANCE (IMB)

In theory, a process is in steady-state if the parameters that define the system's behavior are not changing over time. In practice, a process is never truly at steady-state. However, a plant can normally operate around steady-state for hours or days. For applications that have low frequency of change, steady-state reconciliation can be performed. In transient periods of changing states a dynamic model can be applied [4]. The general formulation of a linear steady-state system model is described in Equation 1 where Y_t is vector of n measurements, x_t is corresponding true values and ϵ is vector of unknown random errors.

$$Y_t = x_t + \epsilon \tag{1}$$

Related constraints such as mass balance is represented by $Ax = 0$ and the objective function for error minimization can be represented Equation 2:

$$min_x (Y_t - x_t)^T W (Y_t - x_t) \tag{2}$$

where W is a diagonal matrix that represents weights that are statistical properties of errors. The analytical solution to above problem can be obtained using Lagrange multiplier optimization method in Equation 4. Data reconciliation is built on the assumption of a linear system model with a normal distribution of random errors [4]. Distribution-based outlier detection method is constructed to detect non-random (gross) errors by applying the Global Test (GT) shown in Equation 3, where σ is standard deviation of data points used in calculation of normal distribution function and cumulative probability for the desired confidence interval.

$$GT = \sum_{t=1}^{n} \left(\frac{Y_t - x_t}{\sigma} \right)^2 \tag{3}$$

The first method used for GED in this paper is called IMB, which is a standard DR-GED method as explained above and the solution for reconciliation is shown in Equation 4. IMB enforces a mass balance constraint at every sensors' measurement instant and does not take into account the system dynamics or the memory effect. IMB method is ideally used with systems at steady-state.

$$\hat{x_t} = Y_t - VA^T (AVA^T)^{-1} AY_t \tag{4}$$

If we assume Y_t matrix contains the masses of input water and De-Super Heater (DSH), and the mass of output vapor, then ideally mass balance constraint should enforce $Ax = 0$, where matrix A represents the system input flow $+1$, output flow -1, and unused measurements 0. For the power plant system, Y_t is a 3-dimensional vector of these masses and a sample is shown in Table 1 and matrix $A = [+1\ +1\ -1]$. IMB method also uses a covariance matrix V, in order to find out how attributes vary together. Covariance matrices are obtained using historic data, where V is a $N \times N$ matrix ($N = 3$ for power plant and $N = 17$ for petrochemical plant) containing the variance-covariance of all the input-output attributes:

$$V = \begin{bmatrix} 104.998 & 1.825727 & 128.1891 \\ 1.825727 & 0.236576 & 2.163642 \\ 128.1891 & 2.163642 & 163.9264 \end{bmatrix}$$

Next, the data is reconciled using Equation 4 and $\hat{x}$ denotes the reconciled (or de-noised) data. Then, under the null hypothesis H_0 (H_0: The system is working, so there is no gross error), a statistical test analogous to global test is applied for detecting gross errors known as Chi-Squared test (X^2), which is a non-parametric statistical test corresponding to cumulative probability, exceeding the 95% criterion gross error is detected. Given r the vector of residuals of linear model that follows a normal distribution with zero mean and the covariance matrix V, the statistical global test is constructed. Statistics given by γ in Equation 5 follows a X^2-distribution with v degree of freedom, where v is the rank of matrix A. The 1×3 Matrix A used in Equation 4 in IMB method has rank 1 and the Chi-Squared test for this process has 1 degree of freedom and 95% confidence ($X^2_{95\%}(m) < 3.84$) corresponding to cumulative probability for desired confidence interval.

$$\gamma = r^T V^{-1} r \tag{5}$$

For the petrochemical plant, the same approach is used for DR-GED on raw values. As depicted in Figure 3, this plant has 17 flow sensors over 3 main branches of material flows and the corresponding sensor data streams. Each branch has its own mass balance consideration as follows: $1 = 2+3+4$, $3 = 5+6+7$, $4 = 8+\ldots+17$. Table 2 shows sample sensor measurements for Figure 3 and matrix A represents the mass balance equation ($V_{17\times17}$ not shown for brevity). The least squares estimation (LSE) method is used to obtain

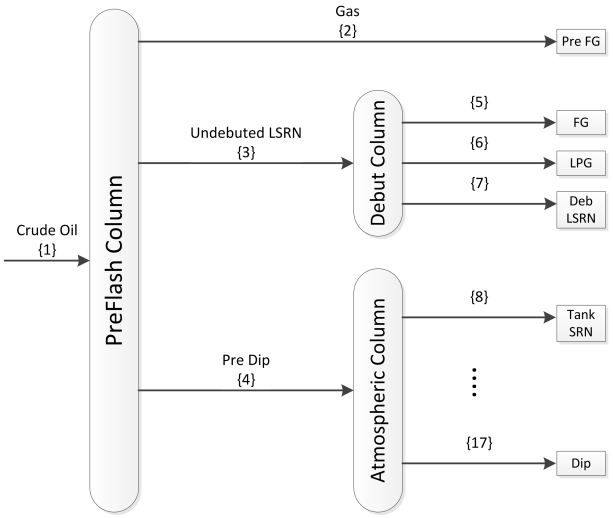

FIGURE 3. Petrochemical process showing crude oil input, preflash, debutanizer, atmospheric columns, and sensors (1-17) measuring input and output flows.

$\hat{Y}_t$ and a Chi-squared test with $rank(A) = 3$ degree of freedom is used for GED in this plant.

$$A = \begin{bmatrix} 1 & -1 & -1 & -1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & -1 & -1 & -1 \end{bmatrix}$$

2) CAPTURING SYSTEM MEMORY USING ARMA MODEL

Assuming plants are linear dynamical systems, their operation can be explained with the Auto-Regressive Moving Average (ARMA) model [41]. This model can capture the ''memory effect'' of the systems with respect to sudden changes in the input. For example, in the power plant any sudden change in the amount of water input might not reflect itself at the vapor output instantaneously, as some water/vapor gets stored in the system during heating. The ARMA model is described by the recursive Equation 6:

$$y_k + \alpha_1 y_{k-1} + \ldots + \alpha_n y_{k-n} = \beta_0 x_k + \ldots + \beta_m x_{k-m} \quad (6)$$

where y_k are the output, and x_k are the input of the system at time k. For the power plant, y_k is the vapor output and x_k is the water input, both in tons/hour. For the petrochemical processes, y_k is the various by-products (fuel, diesel, kerosene) and x_k is the crude oil input. Enforcing instantaneous mass balance (*i.e.* IMB) at the input and output might not work if the system is not in steady state. ARMA model captures this system buffering, thus ''memory effect''. The order of the model in Equation 6 is given by m and n parameters. The higher the order is, the bigger the memory of the system. In this system, discrete time is chosen, since the sensor values are obtained in discrete time intervals. A set of training data is chosen for fitting ARMA model and extracting the coefficients α_i and β_j to identify system order.

Then for a given system order and corresponding coefficients, the estimated model is trained and applied on test data. The system order that performs best on training data is selected, which resulted in $m = n = 1$ with both of our systems (power plant and petrochemical plant).

3) DREDGE

The time-varying Kalman Filter (KF) is a generalization of the steady-state models, *i.e.* systems with non-stationary noise covariance. Although the ARMA model takes into account the dynamics of the boilers, it does not take into account mass balance (*i.e.* all the water that goes in must eventually come out of the system). The upgraded ARMA model for this process requires to have a D.C. gain of 1 which is a smoothing technique in the moving-average model for reduction of random fluctuations in time-series. Given the plants' state obtained by ARMA model, power plant and petrochemical plant are converted into a state-space representation [40]. KF method tracks systems' dynamic state, however, mass balance is not considered. To incorporate system dynamics into mass balance constraints, we integrate Unity Gain constraint with the KF method and train these models over real refinery datasets; calling it DREDGE. Unity Gain constraint in ARMA model is defined with D.C. gain of 1 allowing imbalance instantaneously, but enforcing mass balance to be preserved in the long run. Equation 7 explains this additional linear constraint with respect to Equation 6.

$$-\alpha_1 - \alpha_2 - \ldots - \alpha_n + \beta_0 + \beta_1 + \ldots + \beta_m = 1 \quad (7)$$

ARMA model is applied with prior information and requires to solve a least squares problem subject to unity gain. The problem's formulation is as follows:

$$\begin{aligned} &Minimizing\ ||A\theta - b||^2 \\ &Subject\ to\ C^T\theta = d \end{aligned} \quad (8)$$

where unity gain considered in $C^T\theta$ is equal to one, and the C matrix is a vector of ones. Vector b is real output system and θ is to be estimated. The solution to this problem is obtained by solving the Lagrangian relaxation with optimality condition.

Thus, KF method is modified to use a constrained LSE step in estimating the system coefficients to enforce mass balance. Combining a Chi-squared test and unity gain to KF, we obtained a GED method that mass balance is incorporated in a time-varying process model. You can find details about our modifications and additions to basic KF in Appendix. Since DREDGE combines best of both worlds, we expect it to achieve higher GED accuracy. DR-GED methods introduced and proposed in this section are deployed inside the CEP engine as the first component of the data quality service.

B. GROSS ERROR CLASSIFICATION (GEC)

The Second component of the proposed software service is classification of detected errors called GEC. In the previous section, we addressed models for GED in detail and in this section, we move a step further by formally classifying sensor errors into four types using Complex Decision Trees (CDT), Neural Networks (NN) and K-Nearest Neighbors (KNN)

classifiers. CDT is a regression tree for modeling relationship between variables [42], KNN a density-based classification method that classifies data based on top-k nearest neighbor [43] and artificial NN is again used for pattern recognition and classification. These algorithms were trained with the sudden changes of mean and variance properties of measured data.

Different types of gross errors have their own natural data corruption behavior or characteristics. In relation to the physical defects of the sensors and their operational conditions, common types of gross errors include:

- Bias: Due to un-calibrated sensors, the received values contain a constant shift with respect to the correct value.
- Drift: Due to un-calibrated sensors, the data contains increasing or decreasing amounts of error with time.
- Precision Degradation (PD): The sensor plates may wear out or get dirty over time, resulting in received data from sensors containing errors resembling random noise around the nominal values.
- Failure: Due to sensor failure or measurement boundaries, the received data is constant or completely random.

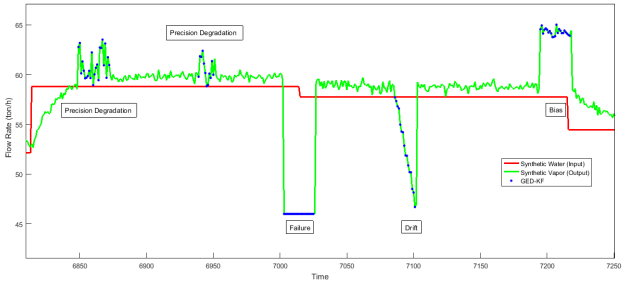

FIGURE 4. Different types of gross errors are inserted on top of refinery power plant data. Blue dots show that gross errors can correctly detected using KF. These errors are then classified by mean-variance tracking.

For comparison of GED accuracies of IMB, KF and DREDGE, and GEC accuracies of CDT, NN, KN, we generated a synthetic dataset with 1 million data samples whose properties were extracted from real data obtained from the power plant. Next, error events representative of different gross error types (Bias, Drift, PD, and Failure) were added to the synthetic data as exemplified in Figure 4. For each gross error type, we calculated the F-measure values of classifiers including CDT, NN, and KNN. CDT was modeled and evaluated by 5-fold cross-validation technique and Gini's diversity index for the split criterion. NN was trained and tested using scaled conjugate gradient back-propagation, which was a two-layer feed-forward network, with sigmoid hidden and softmax output neurons. Similar to CDT, KNN model was trained and evaluated by 5-fold cross-validation technique, and the Euclidean distance metric for 1 nearest neighbor. F-measure is a harmonic mean of precision (TP/(TP+FP)) and recall (TP/(TP+FN)) which provides a unified score for the classification quality evaluation.

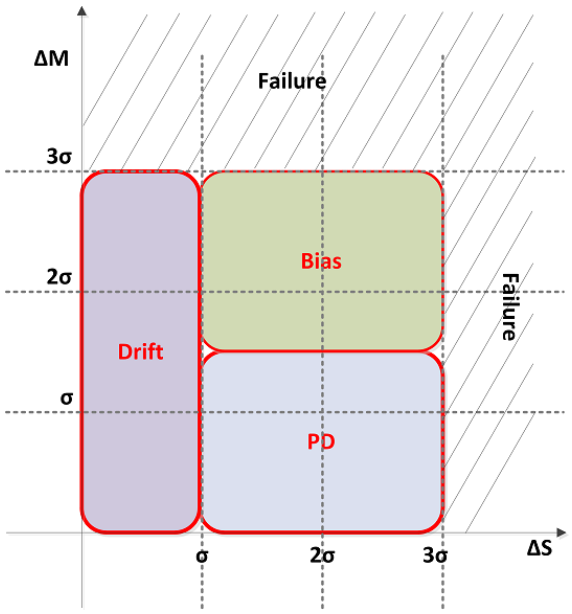

FIGURE 5. General categorization of gross error types with respect to changes in mean and variance of sensor data. The borders are extracted after training classifiers with real data.

For applying a supervised learning algorithm to learn gross error behavior, data should be labeled. But, our real data obtained from oil refinery does not have any extra information about types of gross errors. From the statistical studies of sensor data for GED purposes and the observation of changes in mean and variances, we developed a rule-based algorithm for labeling. The model was trained with mean (μ) and the variance (σ) values of offline data and obtained the model illustrated in Figure 5. This model is used for labeling data and by measuring the latest mean (μ) and the variance (σ) values of the time-series data in current sliding window and classify gross errors accordingly. Given one method of GED that detects a gross error event, if the mean changes up to 3σ ($\Delta M < 3\sigma$) and variance does not change drastically ($S < \sigma$), then the gross error is classified as Drift. If both the mean and variance change significantly ($1.5\sigma < \Delta M \& S < 3\sigma$), then it is classified as a Bias. If the mean does not change significantly ($\Delta M < 1.5\sigma$), but the variance increases ($\sigma < S < 3\sigma$), then is classified as Precision Degradation. Finally, if both variance and the mean changes significantly, it is classified as a Failure-Random. In case there is no movement at all, the sensor is classified as Failure-Dead.

The classification categories illustrated in Figure 5 are separated linearly without any overlap, but in real data, gross error types might have overlaps or some detected data points may contain more than one type of error. By training the classifiers using these rules, the unclassified or misclassified data points can be detected and their true classes can be extracted. These points are also revealed by accuracy evaluation of supervised learning algorithms and F-measure values.

C. SYSTEM OPERATIONAL STATE ANALYSIS IN REAL-TIME

Another aspect of real-time stream analysis and parameter estimation is the frequency of model update. In stream analysis the entire data is not available at all time, the past data may have a large volume, and the cyber model needs to be constructed using a window-based approach, *a*.k.a. sub-model identification. Yet, it is challenging to detect system's operational changes when the process is in a transient or drift mode from one state to another. In model reconstruction, a window should neither be too large to miss the patterns and operational modes, not too small to make frequent, unnecessary updates. The performance evaluation shows that smaller windows sizes are preferable because of lower CPU time and memory usage. An optimal window size can be computed using historical data analysis, but it does not necessarily require a fixed length and can be changed over time based on system's behavior [44].

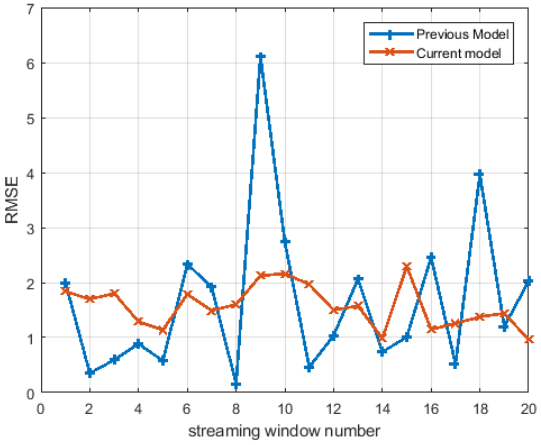

FIGURE 6. Model evaluation of pressure/temperature data.

Stream data context fluctuations in industrial systems is a critical indicator for system modeling and necessity for model updates. This information is extracted using the cleaned, high-quality data obtained form DR-GED process. Here, the system is modeled using the analysis described for GED and GEC in a window-based fashion, with a fixed window size of 180 data points (6 hours) in consecutive tumbling windows. By applying the model from previous window to current window and measuring the Root Mean Square Error (RMSE) from predicted model, the RMSE value is evaluated for operating state identification. We observe that when there is a drift in data context the RMSE increases dramatically as shown in Figure 6 window #9. When the system works in one steady-state, the previous model is applicable to current window; for example between window #6-#8. But when the RMSE increases suddenly, the system goes into a transient state and is an indication of state change. This extracted knowledge is interpreted as a requirement for model update as described in Algorithm 1. The validation of this extracted knowledge is tested using DBSCAN clustering method as described next. Note that in SystemModeTracker Algorithm,

Algorithm 1 SystemModeTracker

```
procedure
W : initial window size
offline:
M=Model(W) //predict model on steady-state data
online: DREDGE
for all windows W over stream do
    GED(W)
    RMSE_currtModel = RMSE(M)
    cluster = DBSCAN(W)
    if (RMSE_currtModel) > (RMSE_prevModel) &
        cluster>2 then
        detectNewMode
        M = updateModel(W)
        RMSE_prevModel = RMSE_currtModel
    end if
    GEC(W)
end for
```

we only use offline models to get the sensor stream started; after that models are trained and tuned online.

1) DBSCAN CLUSTERING

Real-time clustering methods can be used for detecting the system's operating states, where data points would be grouped in one cluster denoting the steady-state and formation of new clusters is interpreted as a drift to new operating conditions or emerging patterns. Note that the anomaly properties of transients will be different than steady-state modes. As such, a locality based outlier detection approach, without specifying cluster numbers in advance is beneficial. Density-based clustering methods are suitable for this evaluation, therefore we employed DBSCAN in this paper.

The applications for sub-model identification in real-time stream analysis are operating state identification and local outlier detection. DBSCAN algorithm can be used for discovering clusters in arbitrary shapes based on the density of data points. It requires two input parameters (Eps, $minPts$) where Eps-neighborhood for a point p is all neighbors within range Eps defined in Equation 9 that has more than $minPts$ data points [45].

$$N_{Eps} = \{q \in D \mid dist(p, q) \leq Eps\} \tag{9}$$

For all series of points q that are density-reachable from p one cluster is formed from connected points and, points that are not reachable are detected as outliers in a window-based analysis.

V. EXPERIMENTAL RESULTS

In this section, the GED accuracy results of IMB, KF, and DREDGE methods for different gross error types are discussed over synthetic data. Next, data reconciliation and error classification are applied over the real refinery dataset. Finally, the performance of different algorithms over different streaming window sizes are tested and compared to

TABLE 3. GED results of IMB, KF, and DREDGE for Bias-type gross error over synthetic data.

	Gross Erorr Existance	GE Prediction T	GE Prediction F
DREDGE	T	2755	0
	F	0	2756
KF	T	2741	14
	F	0	2742
IMB	T	2795	0
	F	40	2796

better understand the scalability and sustainability of these DR-GED techniques over fast moving sensor datasets.

A. RESULTS FOR SYNTHETIC DATA

Table 3 shows the GED accuracy results of IMB, KF, and DREDGE methods in detecting Bias type gross error over the synthetic dataset, to which $N = 2,755$ Bias events/epochs were synthetically inserted at random points. Logically, there will be $(N + 1)$ 2,756 periods where there is no gross error. Each Bias event contains a set of Biased values, the count of which randomly varies between 5 and 25 to provide statistical significance. A ''True (T)'' event signifies the existence and a ''False (F)'' event signifies the non-existence of a gross error in that period. As seen in Table 3, our DREDGE method gives the most accurate prediction results for Bias type gross events, where the 2,755 synthetically inserted events were all correctly detected (true positive-TP) and no misdetections (false positive-FP or false negative-FN) were recorded. However, the KF algorithm misses 14 of the Bias events. The IMB method is the least accurate in this case, where 40 ''non-gross error'' events were detected as gross error events. Since the DREDGE method takes into account both system dynamics and mass balance constraint, it performs the best. Note that an FP increases the total number of both gross error and non-gross error events, whereas an FN decreases both since we have time-series event data. The experiments are repeated for all 4 types of errors (Failure, Bias, Drift, Precision Degradation) and summarized results in Table 4.

Table 4 shows the precision and recall rates in confusion matrix for the GED methods over all gross error types. The precision of both KF and DREDGE are very high (99.54-100%), since they make little or no FP. Since IMB does not track the dynamics of the system it has lower precision ratios (81.3-98.57%) due to FPs. One interesting phenomenon is the relatively low recall rates for KF (78.05%) and DREDGE (90.8%) during detection of PD type errors. This happens because they both suffer from over-fitting their models and include precision degradation errors as regular, non-gross error events. In summary, we learned IMB can cause a lot of false alarms and DREDGE and KF methods are more dependable in their predictions compared to IMB. Therefore, KF-based methods are preferable for GED over time-varying systems found in refineries.

Applying GEC technique on top of GED methods, Table 5 and Table 6 show the F-measure, precision, and recall of all classifiers per error type. The F-measure values are between $0 \leq F \leq 1$ and the higher F-measure shows that the classification has a higher predictive power. In Table 5, we generally observe that the F-measure values of classifiers over synthetic data are higher than the real data. This can be attributed to the higher number and separation of gross errors inserted in the synthetic data, whereas in the real data the errors may be overlapped.

Over synthetic data, CDT has the highest F-measure values $0.994 \leq F \leq 1.0$ and the lowest values are achieved by KNN $0.965 \leq F \leq 0.995$. This shows that the classifiers can learn the labeled data and they have a high predictive power. Next, these classification algorithms are validated over real data.

B. RESULTS FOR REAL REFINERY DATA

After evaluating the accuracy of GED and GEC methods on synthetic data, these methods were validated on real data. IMB, KF, and DREDGE models were trained using a common dataset from a single day (8/1/2013) and tested again on another common dataset from the following day (8/2/2013). The visualization in Figure 7(a) shows the gross errors detected by KF for the flow rate measures (in tons/hour) on the two main lines (water-vapor) of the power plant (time = 2000 minutes). Note that, the transient jumps in data are marked $(*)$ by KF algorithm and there are not gross error marks in the steady-state regions of the data.

In Figure 7(b), we see the petrochemical material flow measurements and the gross errors detected therein. We first observe that GED can be applied on different lines simultaneously to identify and potentially locate which lines have which type of gross errors. We see some locality among the gross errors on different lines, but also some independence. This proves that we can detect, locate, differentiate, classify, and fix gross errors on different sensors for different industrial processes. For sensor 3, where $3 = 5 + 6 + 7$, we used reconciled values obtained from line 1, $1 = 2 + 3 + 4$ (Refer to Figure 2 for system details).

Our GED methods can also be applied among sensor measurements of different types such as temperature-pressure, temperature-flow, etc. Figure 8(a) and Figure 8(b) respectively show that, beyond the use of flow rates, water temperature-pressure and vapor temperature-pressure measurements can be utilized for GED purposes. Using multivariate data is beneficial if some gross errors are not detectable in one set of data, but can be extracted from among different sensor measurements.

Next, we applied the GEC technique over real data from the power plant and report results in Table 7. IMB declared 522 measurements as gross errors, which was significantly more than KF (215) and DREDGE (241). We know from Table 4 that IMB has a low Precision (81.36%) for detecting PD types, therefore the higher GED numbers can be attributed to higher FPs for PD.

TABLE 4. GED results for IMB, KF, and DREDGE over different gross error types on synthetic data.

	Failure		Bias		Drift		PD	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
DREDGE	100%	100%	100%	100%	100%	99.14 %	99.8%	90.8%
KF	100%	100 %	100%	99.49 %	100%	99.37%	99.54%	78.05%
IMB	91.40%	100%	98.57%	100%	95.71%	100%	81.36%	100%

TABLE 5. F-measure for GEC results of CDT, NN, and KNN over different gross error types on Real and Synthetic data.

	Failure		Bias		Drift		PD	
	Real	Synthetic	Real	Synthetic	Real	Synthetic	Real	Synthetic
CDT	0.958	0.994	1.0	0.996	0.992	1.0	1.0	0.998
NN	0.869	0.915	0.867	0.964	0.989	0.974	0.953	0.973
KNN	0.851	0.967	0.836	0.976	0.982	0.995	0.984	0.965

TABLE 6. GEC results for CDT, NN, and KNN over different gross error types on real data.

	Failure		Bias		Drift		PD	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
CDT	95.8%	95.8%	100%	100%	99.2%	99.2%	100%	100%
NN	90.9%	83.3 %	81.3%	92.9 %	99.3%	98.6%	96.8%	93.8%
KNN	86.9%	83.3 %	85.1%	82.1 %	99.2%	97.22%	100%	96.8%

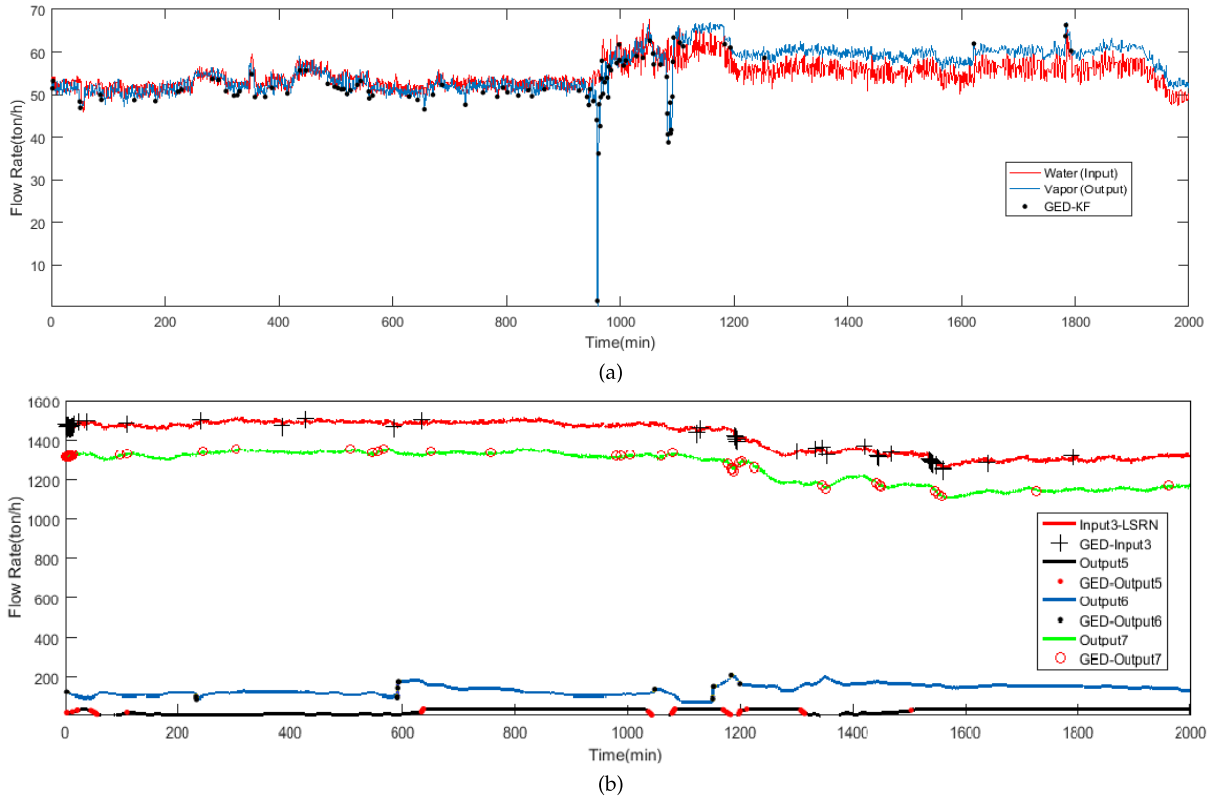

FIGURE 7. (a) GED for the water/vapor lines of the Power Plant data, (b) GED for debutanizer lines $3 = 5 + 6 + 7$ of the Petrochemical process data.

The F-measure obtained from CDT model over real data is $0.958 \leq F \leq 1.0$ for each type of gross error. NN's F-measure for gross error classification is $0.867 \leq F \leq 0.989$. The F-measure achieved by KNN are $0.836 \leq F \leq 0.984$ as shown in Table 5. The lowest F-measure values belong to Failure and Bias which is related

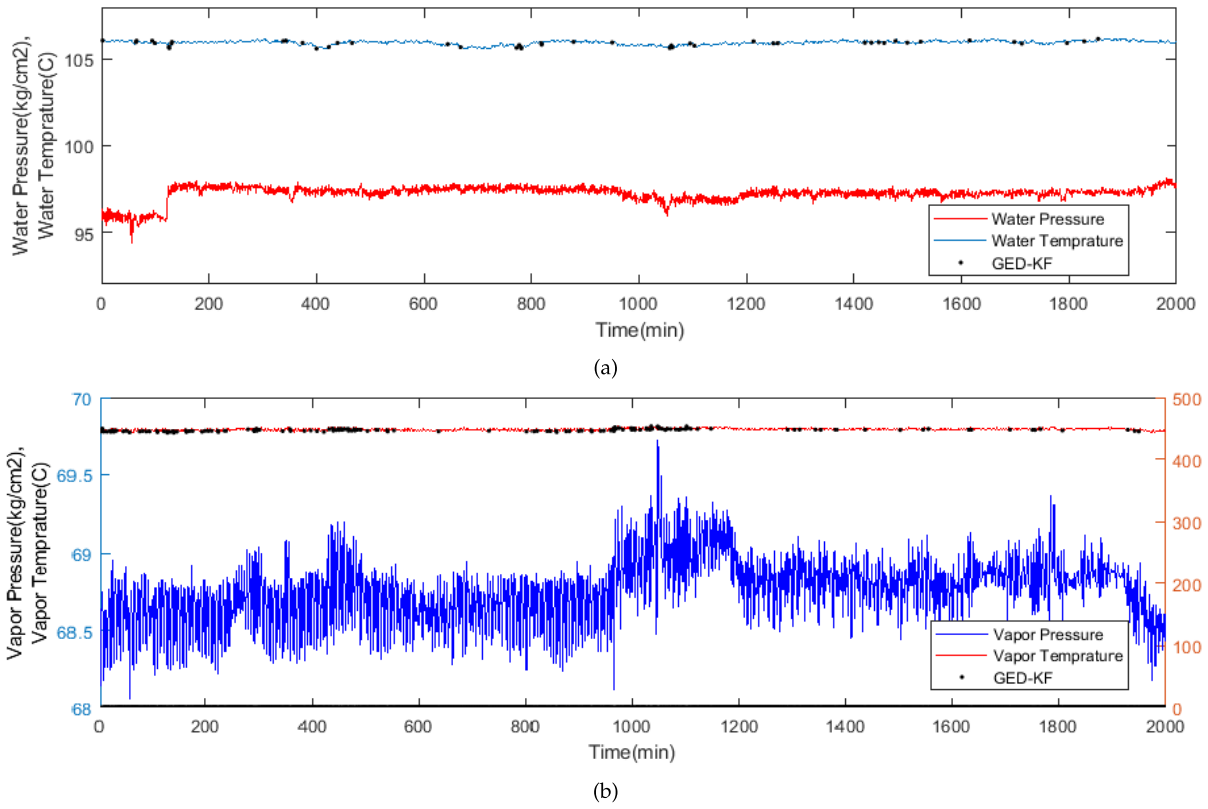

FIGURE 8. (a) Water temperature/pressure lines of power plant used for GED, (b) Vapor temperature/pressure lines of power plant used for GED, by DREDGE.

TABLE 7. GEC results of IMB, KF, and DREDGE over real dataset.

	Gross Erorr Existance	GE Prediction T	GE Prediction F
DREDGE	F	9754	0
	T	5	241
KF	F	9781	0
	T	4	215
IMB	F	9477	0
	T	1	522

to the overlaps between these two types due to their similar behavior. Table 6 also compares the GEC results, but reports the precision and recall details. Lower precision values can be attributed to FPs or ''misdetections'', whereas lower recall values can be attributed to FN or ''missed detections''. While it is desirable to have higher values in both precision and recall, having low recall values may have worse outcomes for oil refineries. As seen in Table 6, KNN may achieve a recall of 83.3% for Failure and 82.1% for Bias type errors. This means that the operators won't be informed 16-17% of the time when those sensor errors are happening, which is not acceptable. In comparison, the highest F-measure, precision and recall values are obtained by CDT. Also, the model's performance is an important concern in stream data processing. As a result, we trained and applied the CDT classifier on CEP engine because of its high accuracy and relatively lower time complexity (*i.e.* $O(logn)$).

Figure 9 shows classification of errors detected by IMB vs. KF-based GED methods. We observe in both Figures 9(a)-(b) that Water-Vapor flow rates cluster around 45-70 tons/hours creating a concentrated green region, which we will refer to as the normal range; GED methods detect and mark gross errors on top of this cluster. IMB works as a sharp, multi-linear classifier for error detection, where values above and below the normal ranges are declared as errors. CDT classifies those above the normal range as Drift, Bias, PD and errors below normal range as Failure types. As seen in Figure 9(b), KF-based models also detect the same Failure types, as detected by IMB in the lower region of cluster, but do not agree in most of the Drift, Bias, PD types (FPs) declared by IMB in the upper region; They agree on a few of the Bias types in this upper region. However, Drift and PD errors detected by KF-based models are more dispersed among the normal values. This is because KF and DREDGE can track Drifts and PDs that are a part of the dynamic system behavior even inside normal ranges.

The validated sensor data that is de-noised through DR-GED process can be used in modeling and other analysis process of the refinery and results of GEC is used for identification of sensor malfunction or system anomaly.

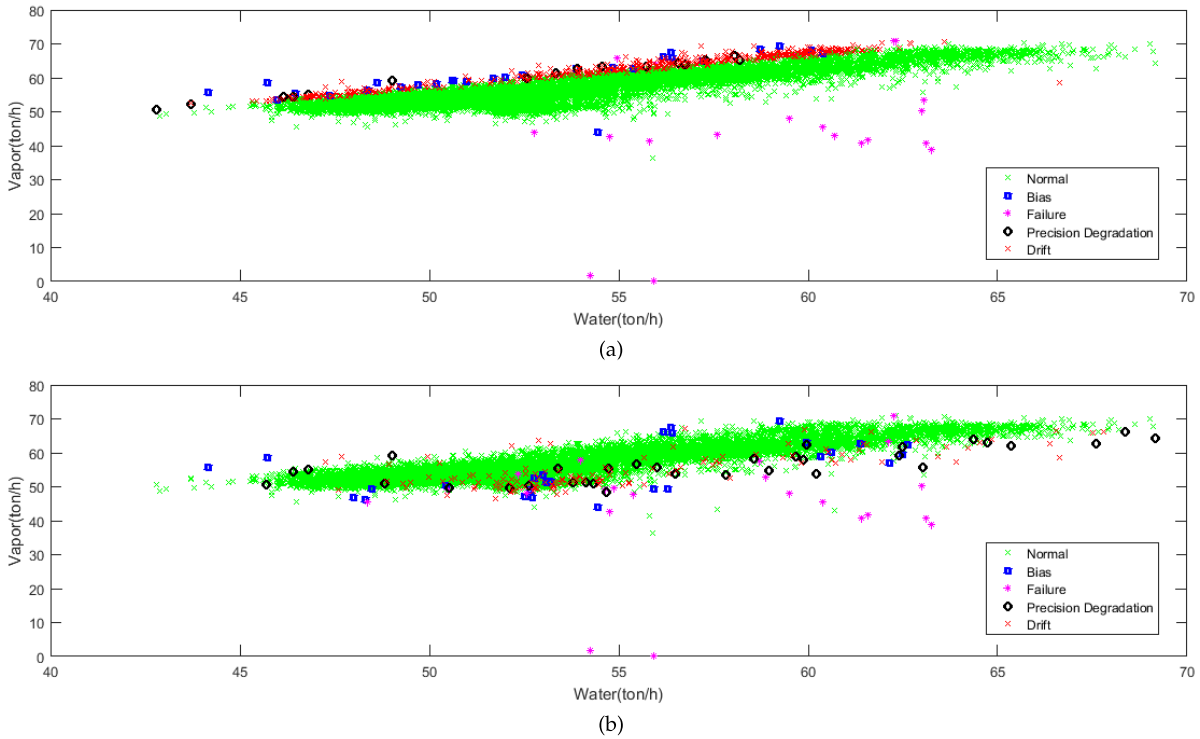

FIGURE 9. (a) GEC for errors detected by IMB, (b) GEC for errors detected by DREDGE.

C. COMPUTATIONAL PERFORMANCE EVALUATION

We learned that with careful selection of system analysis model (DREDGE, KF, or IMB), we can accurately detect and simultaneously classify gross errors. However, the question is whether these methods are sustainable over fast data streams. This section aims to provide an answer by detailed observation of performances for each method. The performance evaluation is done using an open-source CEP engine for Event Stream Processing, called ESPER. The system specifications are ESPER v.4.6.0 installed on RedHat Enterprise Linux Server 5.4 64-Bit OS, Java 1.7.0-05 and JRE v7 on an IBM HS22 Blade Server with Intel Xeon E5530 CPUs (8 cores, 2.40 GHz) and 24 GB DRAM. Data used in these experiments are 200,000 records from (Water, DSH, Vapor) flow sensors sampled every 60 seconds for about 5 months in the TUPRAS power plant. This data was replicated 5 times to form 1 million lines to better represent the real sensor loads. Every line has records of 3 flow sensors in power plant dataset and 17 flow sensors in petrochemical dataset. Therefore, we have 3 million sensor ''events'' in the first and 17 million ''events'' in the second dataset. We publicly provided a 1-month sample of this real data at OpenML datasets site [39] for academic use. Performances of four representative continuous queries (a, b, c, d described below) were evaluated at different window sizes {100, 250, 500, 1000} using a tumbling window type. Tumbling window is a sliding window, where the slide-size is equal to the window size. Continuous queries are:

(a) Select(*) from Boiler: This query returns all sensor data for Boiler's Water, DSH, and Vapor sensors. It is implemented as a reference query with the lowest computational and reference I/O loads.
(b) GEC: We measured the impact of GEC method that primarily uses statistical (stddev and average) library functions inside ESPER.
(c) - (d) GED (IMB and KF-based) methods using the JBLAS linear algebra library for matrix computations.

We start with performance (memory and CPU usage) analysis of queries using the water-vapor dataset from the refinery's power plant. Each experiment is repeated 5 times for each window size and the average values (as well as min and max) are depicted in Figure 10. Figure 10(a) shows the total memory used by different window sizes. For the Select(*) and GEC queries the memory usage is almost constant *w.r.t* the growth of window length as the data is quickly moved in and out of the window. However, for KF-based and IMB GED methods, memory consumption increases linearly *w.r.t* window size. Figure 10(b) shows the total CPU time consumed for GED by different methods over the power plant data. We see that it is possible to process 1 million rows on average in 30 seconds with a single core mapping to a rate of 100,000 events/sec (1 Million events/30sec $=$ 3 events/row $\times$ 33,333 rows/sec). The total time consumed for processing all data with Select(*) and GEC methods shows a slight growth *w.r.t.* window size. IMB time increases almost exponentially *w.r.t.* window size, whereas

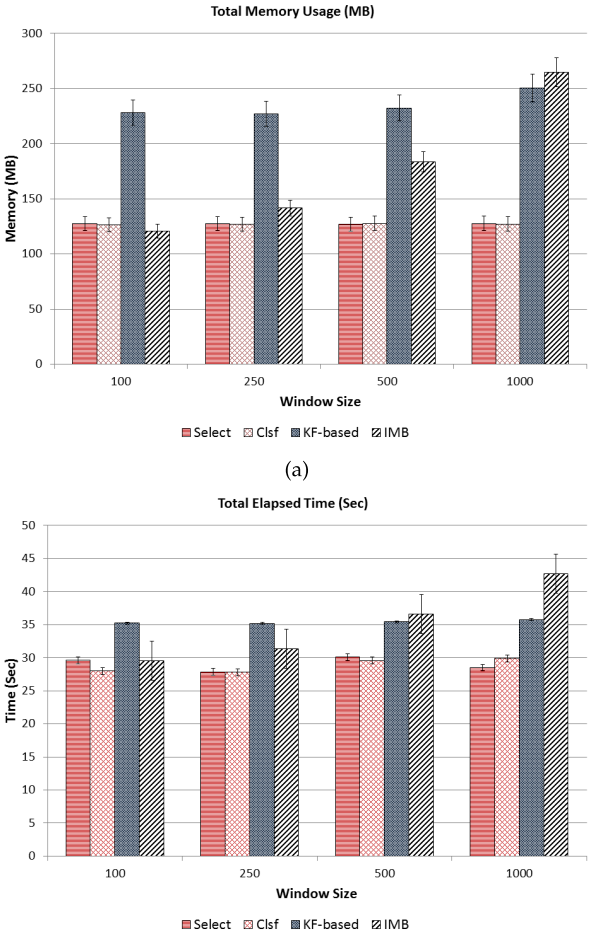

FIGURE 10. Power Plant: (a) computational loads of reference queries and GED algorithms with respect to total memory usage (MB) for different window sizes over streaming data, (b) total processing time (second) for different window sizes over streaming data.

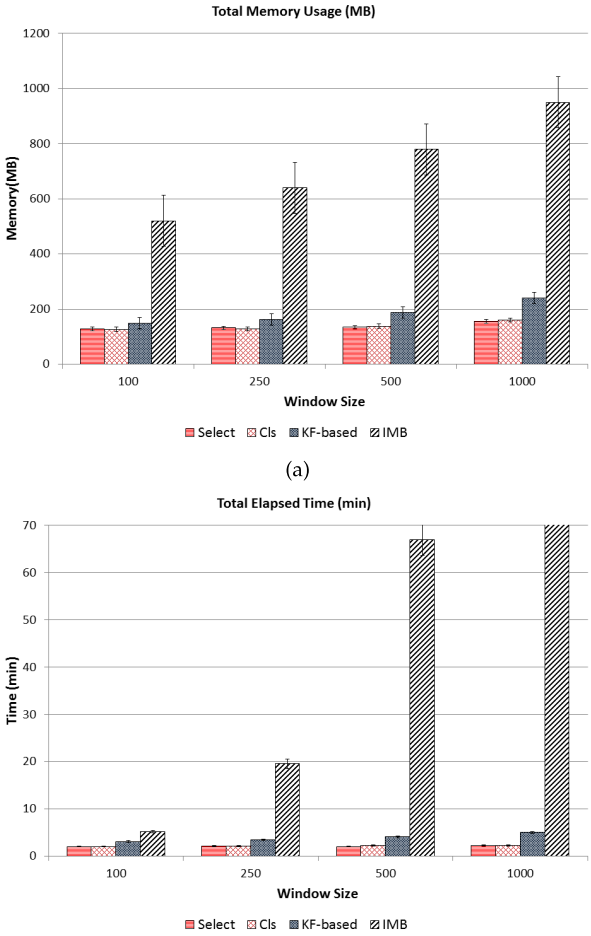

FIGURE 11. Petrochemical Plant: (a) computational loads of reference queries and GED algorithms with respect to total memory usage (MB) for different window sizes over streaming data, (b) total processing time(min) for different window sizes over streaming data.

KF-based only increases linearly. IMB method executes high-dimensional matrix computations for GED. At window size 1000, IMB takes 43 seconds to complete the task (50% slower than for window size 100). Therefore, we conclude that large window sizes are less preferable for GED since we do not want to miss important events due to delays in response. In Figure 11, we continue with performance (memory usage and CPU time) analysis of queries tested with petrochemical processing plant (17-lines) dataset and for different sliding window sizes. In Figure 11(a), we see that the memory consumption of Select(*), Classification, and KF-based increase slightly *w.r.t.* windows size, but these memory loads are not demanding compared to the memory capacity of our server. Similarly, their CPU processing times show a slight linear increase Figure 11(b) from 2 to 3 minutes. However, the processing time of the IMB algorithm grows exponentially as the window size increases from 5 minutes to 310 minutes for the 1000 window size; beyond our charts limits. Again, we conclude that smaller window sizes and use of KF-based are preferable.

D. USING DBSCAN FOR MODE CHANGE DETECTION

As the data stream arrives, DBSCAN algorithm is applied for operating state identification and outlier detection of current window. In Figure 12, the behavior of the system using water/vapor relation is studied. As shown in the figure, until 8th window data there is only one main cluster, but outliers are beginning to show the emergence of a second cluster. However, in 9th window, the data split into 3 clusters, which indicates a transient in the system. The data received in 10th and 11th window stays in the new steady-state mode. DBSCAN clearly enables detecting the outliers without any prior assumption about the distribution of data or any relationship among variables, and whether the system is working under a steady-state operational mode. If a transient happens, DBSCAN can detect the drift by partitioning data into more than one cluster of inliers and outliers.

In the meantime, the operation of the system is evaluated from other sensor measurements of the boiler, pressure and temperature values. As shown in Figure 12, the behavior of the system using flow-rate measurement is evaluated for

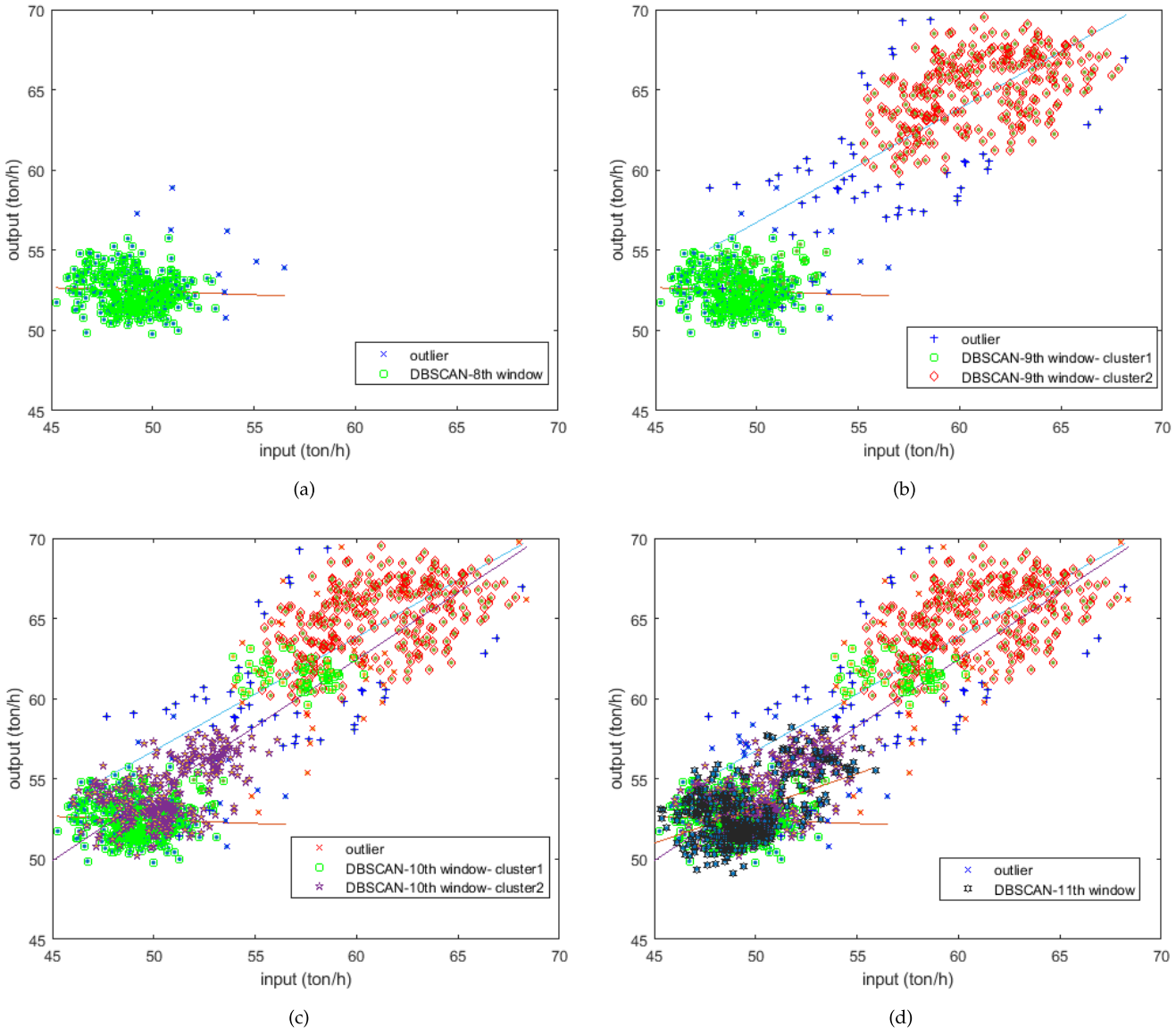

FIGURE 12. DBSCAN model on streaming sensor data water/vapor flow rate: in 8th window (ref. Figure 6) DBSCAN model shows the formation of one cluster of inliers indicating one operational state, as new data samples are received from window #9, data gets split into 2 main clusters and some outliers that indicates a drift in data context and that the system is in a transient mode. Windows #10-#11 shows that the system is operating under a new steady-state mode. (a) DBSCAN: 8th window. (b) DBSCAN: 9th window. (c) DBSCAN: 10th window. (d) DBSCAN: 11th window.

operating state identification. However, using other sensor data such as pressure and temperature as shown in Figure 13 we can observe the same behavior with more distinction. In window #8 the system is operating under one steady-state, and by receiving the new set of data in window #9 a transient is observable since the data is split into 3 different clusters. Windows #10-#11 approve this transition and stay in the new operating state.

E. DISCUSSION: SENSOR ERROR OR SYSTEM ANOMALY

One crucial question to answer is whether an outlier measurement would occur due to a sensor malfunction or system anomaly. The identification of these correlated high-level events could be difficult [46]. However, due to redundant sensors and laws of mass and energy preservation, the system can be monitored in multiple locations (in & out) as well as in multiple dimensions (flow, temperature, pressure) to differentiate sensor *vs.* system issues. In our experiments, we observed that the system will sometimes shift among regular operational modes and go through transient states, which get detected as gross errors in our scenarios. Consequently, in "bias" and "drift" types of gross errors, the system goes through a transient state until it reaches a new steady-state operation. In "precision degradation" type there is no steady-state and in "failure" type there is no sensor or system operation at all.

The system is declared as operating in a steady-state when the model from previous window that is applied on new

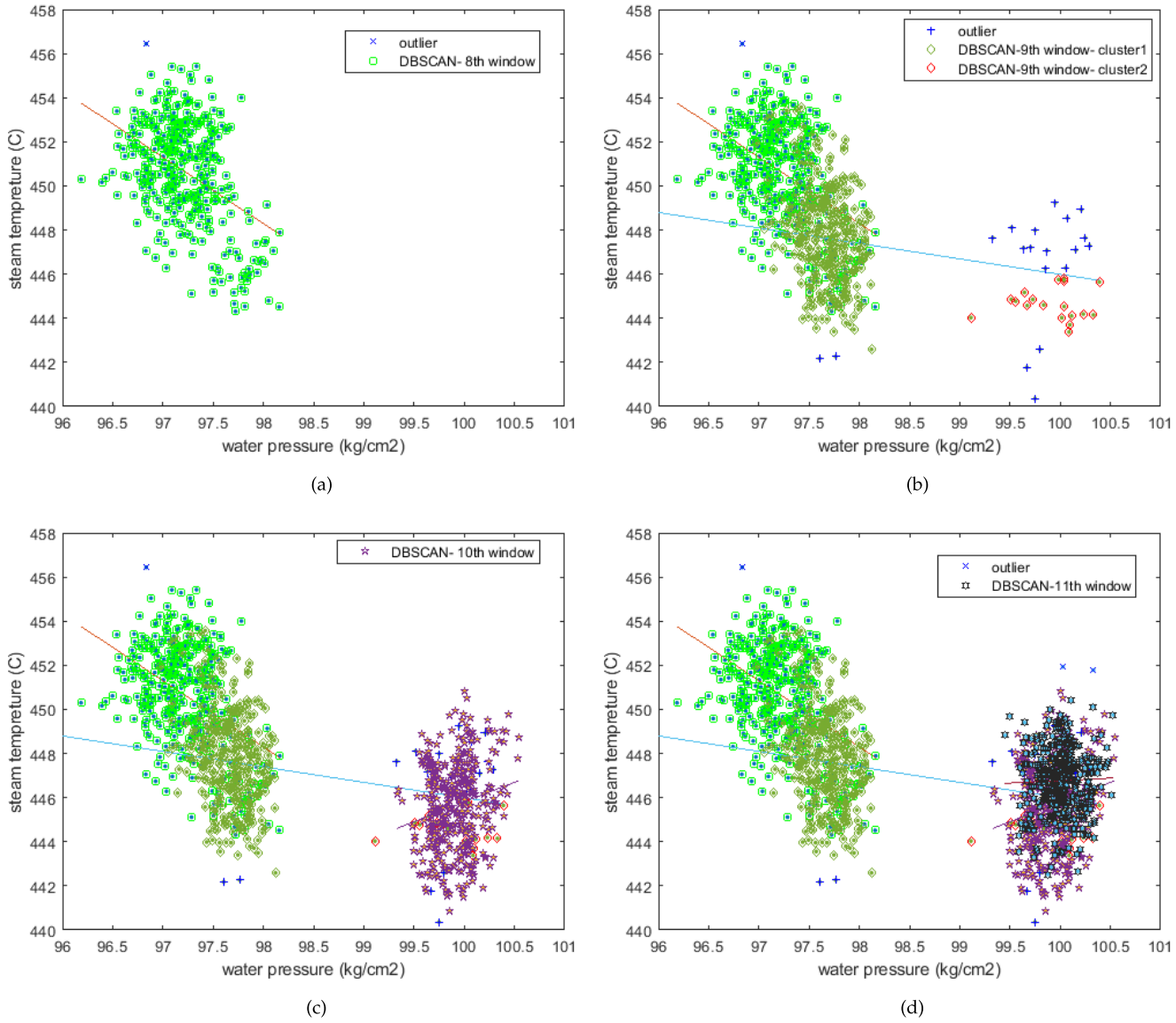

FIGURE 13. DBSCAN model on streaming sensor data water-vapor pressure/temperature: similar to flow rate, in 8th windows DBSCAN model shows formation of one cluster of inliers indicating one operational state, in window #9, data gets split into 2 main cluster and some outliers that is a distinctive indication of drift in data context and the system goes into transient mode, requiring a model update. Windows #10-#11 show that the system is operating under a new steady-state mode. (a) DBSCAN: 8th window. (b) DBSCAN: 9th window. (c) DBSCAN: 10th window. (d) DBSCAN: 11th window.

window data fits well by RMSE value evaluation. However, a new operational mode formation can be identified when the prediction error of the previous model on current window increases dramatically. The operational mode identification compared and is confirmed by DBSCAN clustering method.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we addressed problems associated with erroneous sensor readings in oil refineries and we proposed a real-time data validation, gross error detection (GED) and classification (GEC) service, leveraging tools from statistics, signal processing, data mining and a CEP engine integrated with cyber-physical systems. For comparison of GED and GEC accuracies as well as computational performances, we obtained time-series data from the power and petrochemical plants of an oil refinery. We found that our proposed DREDGE method has accurate error detection (99.1-100%) and sustainable performance at smaller window sizes. IMB method had lower accuracy results and its performance degraded exponentially with increasing window size. After comparison of three gross error classification (GEC) methods, we found the Complex Decision Tree (CDT) to have the highest precision and recall values (95.8-100%), where KNN had the lowest recall values (e.g. 82.1%) that would be unfit for oil refineries and their safety requirements. Therefore, we implemented CDT technique into CEP engine for real-time GEC.

Our approach combines data cleaning via gross error detection, steady-state system modeling, real-time operational mode identification and model updates, all at once. This study is applicable for many data-driven systems dealing with sensor streams to ensure data quality and improve system modeling. Devices in interconnected CPS can communicate for more accurate system monitoring and avoid total failure by predicting and detecting abnormal behavior of the system. We believe that our study has the potential to bridge the gap between generic big data software available in the market and real challenges faced by oil refineries: detecting erroneous data and sensors with high accuracy (no false positives or negatives) over high volume stream data. Error detection and classification such as ours provide a data quality improvement for better system modeling and isolation of faulty processes. In future work, we plan to integrate models from our system into the real refinery, apply techniques from deep learning [47], real-time model updates, and apply the proposed architecture to other production plants.

APPENDIX
KALMAN FILTER

Kalman filter is a state space model used for tracking and parameter estimation formulated with the system Equation 10 where x_{n+1} is the state vector at time n that is transformed to y_n the measurement vector. A, B, C are state transition and measurement matrices and w_n and v_n are white Gaussian noise with zero mean.

$$\begin{aligned} \textit{System state model}: \ & x_{n+1} = Ax_n + Bu_n + w_n \\ \textit{Measurement model}: \ & y_n = Cx_n + v_n \end{aligned} \tag{10}$$

KF has two phases: prediction for projecting current state obtaining *a priori* estimate as shown in Equation 11, and correction step for obtaining *posteriori* estimate by incorporating actual measurement into the a priori estimate as $\hat{x}_{n|n}$, P is the estimate error covariance, Q and R are covariance matrices and, M_n is the KF gain [4]. The prediction and correction steps are executed recursively as follows:

Prediction step

$$\begin{aligned} \hat{x}_{n+1|n} &= A\hat{x}_{n|n} + Bu_n \\ P_{n+1|n} &= AP_{n|n}A^T + Q_n \end{aligned} \tag{11}$$

Correction step

$$\begin{aligned} \hat{x}_{n|n} &= \hat{x}_{n|n-1} + M_n(y_n - C\hat{x}_{n|n-1}) \\ M_n &= P_{n|n}C^T(CP_{n|n-1}C^T + R_n)^{-1} \\ P_{n|n} &= (I - M_nC)P_{n|n-1} \end{aligned} \tag{12}$$

Given the observed input data, the system state is tracked in a KF. The "innovation" is computed in the correction step of the KF, which is subjected to a Chi-Squared test. If the test fails, a gross error is detected. KF is used for random error detection and data reconciliation in sensor data and an adapter is required to turn its output prediction error (called *innovations*) into a GED tool, similar to the statistical global test used in IMB. Constructing a statistical test in dynamic linear systems is possible by utilizing the properties of KF innovations (*i.e.* output prediction error), which is computed in the correction step of KF. Innovations have normal distributions with expected values and a covariance matrix V_k given by Equation 13, where $P_{k|k-1}$ is *a priori* estimate covariance, C is observation model and Q_k is noise covariance matrix [4].

$$V_k = CP_{k|k-1}C^T + Q_k \tag{13}$$

Equation 14 is used to obtain a γ value, where V_k is innovation covariance and v_k is innovation residual at time k. The γ value follows a Chi-squared distribution with 1 degree of freedom and if it exceeds the criterion 95% corresponding probability for the desired confidence interval, the test fails and the gross error existence, as well as its location, are detected [4].

$$\gamma = v_k^T V_k^{-1} v_k \tag{14}$$

REFERENCES

[1] R. K. Perrons and J. W. Jensen, "Data as an asset: What the oil and gas sector can learn from other industries about 'big data,'" *Energy Policy*, vol. 81, pp. 117–121, Jun. 2015.
[2] E. Lughofer and M. Sayed-Mouchaweh, "Autonomous data stream clustering implementing split-and-merge concepts–towards a plug-and-play approach," *Inf. Sci.*, vol. 304, pp. 54–79, May 2015.
[3] *TUPRAS-Refinery*. Accessed: Aug. 24, 2018. [Online]. Available: http://tupras.com.tr/en/rafineries
[4] S. Narasimhan and C. Jordache, *Data Reconciliation & Gross Error Detection: An Intelligent Use of Process Data*. Houston, TX, USA: Gulf Publishing Company, 2000.
[5] C. Harrison *et al.*, "Foundations for smarter cities," *IBM J. Res. Develop.*, vol. 54, no. 4, pp. 1–16, 2010.
[6] P. C. Evans and M. Annunziata, "Industrial Internet: Pushing the boundaries of minds and machines," Gen. Electr., Boston, MA, USA, White Paper, 2012. [Online]. Available: http://www.ge.com/docs/chapters/Industrial_Internet.pdf
[7] J. Gertler, "Fault detection and diagnosis," in *Encyclopedia of Systems and Control*. London, U.K.: Springer, 2015.
[8] M. V. Shcherbakov, A. Brebels, N. L. Shcherbakova, V. A. Kamaev, O. M. Gerget, and D. Devyatykh, "Outlier detection and classification in sensor data streams for proactive decision support systems," *J. Phys., Conf. Ser.*, vol. 803, no. 1, p. 012143, 2017.
[9] D. Jankov, S. Sikdar, R. Mukherjee, K. Teymourian, and C. Jermaine, "Real-time high performance anomaly detection over data streams: Grand challenge," in *Proc. 11th ACM Int. Conf. Distrib. Event-Based Syst.*, 2017, pp. 292–297.
[10] A. Varga, *Solving Fault Diagnosis Problems*. Springer, 2017.
[11] B. Tang and H. He, "A local density-based approach for outlier detection," *Neurocomputing*, vol. 241, pp. 171–180, Jun. 2017.
[12] C. Guarnaccia, L. Elia, J. Quartieri, and C. Tepedino, "Time series analysis techniques applied to transportation noise," in *Proc. IEEE Ind. Commercial Power Syst. Eur. Environ. Elect. Eng.*, Jun. 2017, pp. 1–6.
[13] G. Krempl *et al.*, "Open challenges for data stream mining research," *ACM SIGKDD Explorations Newslett.*, vol. 16, no. 1, pp. 1–10, 2014.
[14] M. Cruz, M. Bender, and H. Ombao, "A robust interrupted time series model for analyzing complex health care intervention data," *Statist. Med.*, vol. 36, no. 29, pp. 4660–4676, 2017.
[15] G. Reikard, S. Haupt, and T. Jensen, "Forecasting ground-level irradiance over short horizons: Time series, meteorological, and time-varying parameter models," *Renew. Energy*, vol. 112, pp. 474–485, Nov. 2017.
[16] A. Vasebi, É. Poulin, and D. Hodouin, "Dynamic data reconciliation in mineral and metallurgical plants," *Annu. Rev. Control*, vol. 36, no. 2, pp. 235–243, 2012.
[17] A. Rafiee and F. Behrouzshad, "Data reconciliation with application to a natural gas processing plant," *J. Natural Gas Sci. Eng.*, vol. 31, pp. 538–545, Apr. 2016.

[18] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 657–667, Jan. 2015.
[19] *Apache Hadoop Project*. Accessed: Aug. 24, 2018. [Online]. Available: http://hadoop.apache.org
[20] ESPER Espertech Inc. *Event Stream Intelligence*. Accessed: Aug. 24, 2018. [Online]. Available: http://www.espertech.com
[21] C.-L. Yang *et al.*, "Streaming data analysis framework for cyber-physical system of metal machining processes," in *Proc. IEEE Ind. Cyber-Phys. Syst. (ICPS)*, May 2018, pp. 546–551.
[22] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions," *Neurocomputing*, vol. 239, pp. 39–57, May 2017.
[23] E. Olmezogullari and I. Ari, "Online association rule mining over fast data," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, Jun. 2013, pp. 110–117.
[24] E. C. do Valle, R. de Araújo Kalid, A. R. Secchi, and A. Kiperstok, "Collection of benchmark test problems for data reconciliation and gross error detection and identification," *Comput. Chem. Eng.*, vol. 111, pp. 134–148, Mar. 2018.
[25] Z. Zhang, Y.-Y. Chuang, and J. Chen, "Methodology of data reconciliation and parameter estimation for process systems with multi-operating conditions," *Chemometrics Intell. Lab. Syst.*, vol. 137, pp. 110–119, Oct. 2014.
[26] Z. Zhang and J. Chen, "Simultaneous data reconciliation and gross error detection for dynamic systems using particle filter and measurement test," *Comput. Chem. Eng.*, vol. 69, pp. 66–74, Oct. 2014.
[27] S. Guo, P. Liu, and Z. Li, "Data reconciliation for the overall thermal system of a steam turbine power plant," *Appl. Energy*, vol. 165, pp. 1037–1051, Mar. 2016.
[28] B. Cai *et al.*, "Multi-source information fusion based fault diagnosis of ground-source heat pump using Bayesian network," *Appl. Energy*, vol. 114, pp. 1–9, Feb. 2014.
[29] G. Ruan, P. C. Hanson, H. A. Dugan, and B. Plale, "Mining lake time series using symbolic representation," *Ecol. Inform.*, vol. 39, pp. 10–22, May 2017.
[30] X. Jiang, P. Liu, and Z. Li, "Data reconciliation and gross error detection for operational data in power plants," *Energy*, vol. 75, pp. 14–23, Oct. 2014.
[31] W. Zhang, M. Hirzel, and D. Grove, "AQuA: Adaptive quality analytics," in *Proc. 10th ACM Int. Conf. Distrib. Event-Based Syst.*, 2016, pp. 169–180.
[32] E. Lughofer, M. Pratama, and I. Skrjanc, "Incremental rule splitting in generalized evolving fuzzy systems for autonomous drift compensation," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 4, pp. 1854–1865, Aug. 2017.
[33] Z. Zhu, G. Geng, and Q. Jiang, "Multi-scenario parameter estimation for synchronous generation systems," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 1851–1859, May 2017.
[34] A. Shaker and E. Lughofer, "Self-adaptive and local strategies for a smooth treatment of drifts in data streams," *Evol. Syst.*, vol. 5, no. 4, pp. 239–257, 2014.
[35] R. R. Rajkumar, I. LEE, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. 47th Design Autom. Conf.*, 2010, pp. 731–736.
[36] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015.
[37] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.
[38] J. Wan, H. Cai, and K. Zhou, "Industrie 4.0: Enabling technologies," in *Proc. Int. Conf. Intell. Comput. Internet Things (ICIT)*, Jan. 2015, pp. 135–140.
[39] *Turkish Petroleum Refineries, Boiler Data*. Accessed: Aug. 24, 2018. [Online]. Available: https://www.openml.org/d/41170
[40] L. Zhang and X. Peng, "Time series estimation of gas sensor baseline drift using ARMA and Kalman based models," *Sensor Rev.*, vol. 36, pp. 9–34, Jan. 2016.
[41] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.
[42] L. Breiman, *Classification and Regression Trees*. Evanston, IL, USA: Routledge, 2017.
[43] G. O. Campos *et al.*, "On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study," *Data Mining Knowl. Discovery*, vol. 30, no. 4, pp. 891–927, 2016.
[44] A. Khodabakhsh, I. Ari, M. Bakir, and S. M. Alagoz, "Stream analytics and adaptive windows for operational mode identification of time-varying industrial systems," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, Jul. 2018, pp. 242–246.
[45] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, p. 19, 2017.
[46] A. Akbar *et al.*, "Real-time probabilistic data fusion for large-scale IoT applications," *IEEE Access*, vol. 6, pp. 10015–10027, 2018.
[47] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.

ATHAR KHODABAKHSH (S'15) received the B.Sc. degree in software engineering from the Computer and Electrical Engineering Department, Islamic Azad University, Zanjan, Iran, in 2005. She is currently purusing the Ph.D. degree with the Computer Science Department, Özyeğin University, Istanbul, Turkey. She is also a Teaching and Research Assistant at Özyeğin University. Her fields of interest include data science, data mining, cloud computing, and distributed systems.

ISMAİL ARİ received the Ph.D. degree from the University of California Santa Cruz in 2004. He was with HP Labs, Palo Alto, CA, USA, until 2009. From 2013 to 2018, he was the Deputy General Manager of Teknopark Istanbul and the Vice President of TUBITAK (The Scientific and Technological Research Council of Turkey). He is currently an Assistant Professor with the Computer Science Department, Özyeğin University. He has publications and patents in the fields of big data, cloud computing, and networked storage systems. He was a recipient of several IBM Faculty Awards, the EU Marie Curie Award, the TUBITAK Career Award, and the Rector's Merit Award.

MUSTAFA BAKİR is currently pursuing the Ph.D. degree with the Computer Science Department, Gebze Technical University, Kocaeli, Turkey. He is also a Manager with the Process Improvement and Software Department, Tüpraş Refinery, where he is responsible for the company's digital transformation.

ALİ OZER ERCAN (S'02–M'07–SM'15) received the B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, CA, USA, in 2002 and 2007, respectively. From 2007 to 2009, he was with the Berkeley Wireless Research Center, University of California at Berkeley, Berkeley, CA, USA, for post-doctoral studies. He joined Özyeğin University, Istanbul, Turkey, in 2009, as an Assistant Professor. His research interests are in the areas of signal and image processing, and wireless communication networks. He was a recipient of the FP7 Marie Curie International Reintegration Grant and the TUBITAK Career (3501) Award. He served as the Publications Chair of the IEEE 3DTV Conference (3DTV-CON) in 2011 and the Technical Program Co-Chair and Publications Chair of the IEEE Signal Processing and Communication Applications Conference in 2012.