Anomaly Detection in Data Streams: The Petrol Station Simulator

Anna Gorawska[1] and Krzysztof Pasterak[1]

Silesian University of Technology,
Faculty of Automatic Control, Electronics, and Computer Science,
Institute of Informatics,
Akademicka 16, 44-100 Gliwice, Poland
{Anna.Gorawska, Krzysztof.Pasterak}@polsl.pl

Abstract. Developing anomaly detection systems requires diverse data for training and testing purposes. Real measurements are not necessarily reliable at this stage because it is almost impossible to find a diverse training set with exactly known characteristics. The petrol station simulator was designed to generate measurements that mimic real petrol station readings. The simulator produces datasets with exactly specified anomalies to be detected via anomaly detection system. The paper introduces foundations of the simulator with results. The discussion section presents future work in the area of stream data extraction and materialization in the Stream Data Warehouse.

Keywords: petrol station simulator, petrol station, fuel leak detection, sensor miscalibration, statistical inventory reconciliation, data stream, stream data extraction, stream data warehouse

1 Introduction

The biggest danger that comes with storage of petroleum or other hazardous substances is contamination of groundwater supplies caused by leaks and spills [2,10]. The majority of petrol stations are not equipped with specialized appliances that prevent or detect leakages. Therefore, manual or automatic tank gauging [1,12], tank tightness testing [13,15], and inventory reconciliation techniques [5,14] are commonly used. In those methods the most important factors are frequency and detection accuracy, while as it was stated in [5] there is a variety of disturbing phenomena present in the inventory data that can negatively affect those factors. In reality problems described in [5] cannot be examined separately, as though they tend to overlap. As a consequence, detection accuracy may be extremely decreased or even a leakage may not be detected due to e.g. tank miscalibration [9].

The data gathered from a real petrol station usually does not contain hazardous anomalies, as though they appear very rarely. It is almost impossible to find a proper and diverse training set for anomaly detection purposes when relying solely on real data inventories. Thus, there was a need to implement a

petrol station simulator to generate datasets with specified characteristics and anomalies. In the following paper, the petrol station simulator is understood as a software designed to reproduce the majority of phenomena that may occur on a petrol station in terms of fuel appliances operation. As a result, it can behave as a virtual petrol station which emulates the real one in terms of measurement data delivery.

2 Requirements and Prerequisites

The purpose of the petrol station simulator is not being plainly a mock data source for test purposes. Our main goal was to create a reliable data source which can reproduce uncommon and rare situations, such as compound anomalies occurring on a real petrol station.

Since the designed petrol station simulator is intended to be used as a source of interesting data for analysis and anomaly detection, it has to model a real petrol station accurately. The compatibility ought to be achieved on four levels: behaviour, configuration, data, and anomaly-level.

Behaviour-level Compatibility. On a real petrol station fuel is stored in tanks and distributed to customers via nozzles that are installed in dispensers. A complex piping system connects tanks with specified nozzles, therefore the simulator is providing data according to determined schema of connections between petrol station's appliances, i.e. station infrastructure. Moreover, the petrol station simulator must generate fuel deliveries as cyclic events of refuelling the tank and simulate fuel purchase operations (*transactions*), that are consistent with dispensing fuel from the tank via nozzles.

All appliances in the petrol station infrastructure behave in a specific way that the petrol station simulator is mimicking. Behaviour-level compatibility ensures that output data are highly accordant to the real data and can be used as a substitute in a variety of applications.

Configuration-level Compatibility. Sometimes there is a need to simulate operation of a certain petrol station, i.e. with a specified infrastructure and parameters (e.g. capacity of tanks, delivery cycle). Configuration-level compatibility ensures that virtual components of the simulated petrol station comply with their real counterparts and if needed they can be fully parametrized enabling simulation of an arbitrary real petrol station.

Data-level Compatibility. Data acquired from a real petrol station has a specific format which can differ according to specific stations, just like applications that process the data can impose additional restrictions. Thus, the petrol station simulator has to produce data in a generic format that can be easily transformed to one required by an actual data recipient. By data-level compatibility we ensure that the petrol station simulator generates data that has exactly

the same format as the real inventory dataset. Moreover, since data from real petrol stations is continuously delivered to the processing system, the simulator has to implement stream output data production [4,6,11] or at least batch (file) generation.

Anomaly-level Compatibility. An anomaly can be understood as an abnormal and potentially dangerous situation, which sometimes occurs during normal operation of a petrol station. Anomaly-level compatibility ensures that all possible phenomena are implemented in the simulator, especially anomalies. Among the most common anomalies we can distinguish: fuel leak from a tank or piping, fuel surplus (increase in fuel volume in a tank not connected to natural fuel behaviour [5]), water influx in a tank, level probe hang [5] or density mismatch, tank or dispenser meter miscalibration [9].

3 The Petrol Station Simulator

For the anomaly detection purpose, three main types of test data can be distinguished: data from real petrols station with detected and identified anomalies, data from real petrol stations with artificially applied anomalies, artificial data obtained from the petrol station simulator. Since obtaining fully analysed real inventories with confirmed anomalies is generally difficult to accomplish, the two other are common sources of data.

The petrol station simulator is generating artificial measurements that can be configured so that they resemble measurements from a particular petrol station. The second solution is a simple application that applies anomalies to real data.

The main purpose of generating data was to obtain idealized measurements with anomalies included – which is helpful in the early stages of developing any anomaly detection system. Then real data with anomalies applied was used later, to test the behaviour of anomaly detection system when operating in more realistic environment.

Since output measurements from the simulator and the anomaly applier are compatible with real inventories, they can be used interchangeably, i.e. anomaly applying software can be used to process data generated by the simulator.

3.1 Dispensing Transactions

In the petrol station simulator, *transaction* is a single dispensing operation consistent with refuelling a car via nozzle. The amount of fuel per transaction (V_{tr}) is calculated according to Equation 1, where $V_{tr_{avg}}$ represents *transaction average volume*, $V_{tr_{var}}$ *transaction volume variance*, and r is a uniformly distributed random value, such as: $r \in \langle 0, 1 \rangle$.

$$V_{tr} = 2r \cdot V_{tr_{var}} + V_{tr_{avg}} - V_{tr_{var}} \tag{1}$$

For simulation purposes we have adopted a *queue model* to enable presentation of different traffic intensities on a petrol station. Similarly as in real petrol

station, all available nozzles are organized in groups (dispensers). With each dispenser there is an associated waiting queue with newly incoming customers – fuel transaction precursors. The arrivals of customers are generated by a single source with a parametrized rate (i.e. *average number of transactions per hour*, *transaction average fuel volume*, *transaction volume variance*), knowing that customers usually prefer to select dispenser with the shortest waiting queue.

Let us assume that there are m dispensers and waiting queue for *i-th* dispenser (d_i) is denoted as q_i. The size of a single queue is n_i, the number of customers already queued – c_i, and the number of free slots in the *i-th* queue is $s_i = n_i - c_i$. The probability p_i for a customer to select an *i-th* dispenser is as follows:

$$p_i = \frac{s_i}{\sum_{j=1}^{m} s_j} \tag{2}$$

In order to emphasize significant differences between daily and nightly rate of transactions we have decided to use the *sinusoidal model* of fuel transactions intensity to present daily cycle of transactions. We have assumed that its maximum (equal to twice the *average number of transactions*) is at 3:00 PM and its minimum (equal to 0) is at 3:00 AM. Between aforementioned maximum and minimum values the intensity resembles a sine function.

The basic unit of time during simulation is one minute which refers to $\frac{2\pi}{60 \cdot 24}$, assuming that the whole day is equivalent to 2π. Since the maximum intensity is intended to be at 3:00 PM, the original time ought to be shifted by 3 hours ($3 \cdot 60$ minutes). The resulting intensity of transactions ($T_{avg}^{'}$) can be defined by the average number of transactions (T_{avg}) multiplied by the factor obtained from the sine function:

$$T_{avg}^{'} = T_{avg} \cdot \left(1 - \sin\left(\frac{t + 3 \cdot 60}{60 \cdot 24} \cdot 2\pi\right)\right) \tag{3}$$

3.2 Delivery

During normal operation the level of fuel stored in the tank is decreasing according to the amount of fuel that is dispensed from the tank during transactions. Therefore, to determine when deliveries must be triggered, it is necessary to define a *work area* for each tank, i.e. tank operating range between so called *low level* and *work level*. When the amount of fuel in the tank is smaller then the *low level* the delivery is triggered, as though the tank might destabilize with a small amount of fuel stored. The *work level* (WL) points to an upper limit of fuel stored in the tank. Filling the tank above this level may cause tank overfill, which might result in fuel spillage into the soil.

In the petrol station simulator the size of a delivery, i.e. the amount of fuel that will be poured into the tank, is calculated as follows:

$$V_d = r \cdot V_{WL_{var}} + V_{WL} - V_{WL_{var}} - V_s \tag{4}$$

In Equation 4 V_{WL} denotes volume consistent with the tank's *work level*, $V_{WL_{var}}$ its variance, and V_s represents the amount of fuel currently stored in the tank.

In the petrol station simulator delivery is not an instantaneous event, i.e. the process of refuelling the tank takes time linearly proportional to the delivery volume from Equation 4 and *refuel speed*. The *refuel speed* is a simulation global parameter that determines delivery throughput.

In the petrol station simulator deliveries are triggered when the amount of fuel stored in the tank is approaching the *low level* value. In reality deliveries are mainly scheduled according to tank's fuel consumption and other statistics.

3.3 The Simulation Configuration

The simulation configuration starts with determining global parameters, i.e. source of configuration (file or a local database), destination for generated data (e.g. database, text file, data stream), *refuel speed*, *transaction volume variance*, and time frame of the simulation (*start* and *end time*). Then every single tank can be configured separately in terms of basic functional parameters:

- *Capacity* – capacity of a fuel tank,
- *Low Level* and its *Variance* – the minimal secure amount of fuel in a tank,
- *Work Level* and its *Variance* – the maximal secure amount of fuel in a tank,
- *Average Transactions/Hour* – average number of fuel transactions per hour,
- *Transaction Average Volume* – average volume of fuel transactions,
- *Calibration Table/Curve* – function that transforms height measurements into volume measurements [5,9].

With tanks parametrized, anomalies of four types can be applied: tank or pipe leak, surplus, probe hang [5]. Anomalies may overlap and there is no upper limit on the number of anomalies applied on one tank. It is possible to induce an anomaly only in a specific time frame of the simulation by defining *start* and *end* time. All anomalies, except probe hang, require specifying *volume* or *speed*, i.e. intensity of the applied anomaly. It can be defined either as the total volume or speed (in litres per hour).

All parameters can be imported from a database or a file. Real petrol station's configuration, when saved in a database, can be directly transferred into the simulator. Moreover, some of the imported parameters can be further edited. The simulator allows also to export its configuration to a file.

4 Results

Usually on real petrol stations it can be observed that during nights the intensity of transactions is generally lower than during daytime.

Figure 1 presents fuel volume measurements from a 12-day simulation where petrol station configuration was extracted from a real station. The tank presented

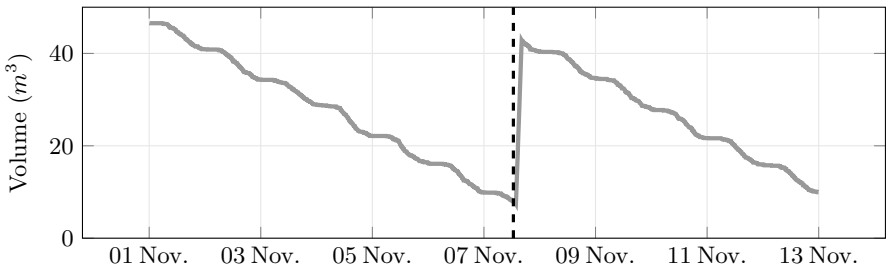

Fig. 1. Simulated fuel volume function

on Figure 1 can store a maximum of 49000 litres with a *work level* of 46550 litres. Starting fuel volume in the simulation was set to *work level*. For the simulated tank the *lower level* was set to 5000 litres – when the fuel level approached that value, delivery was triggered (marked with a vertical line on Figure 1). The daily cycle of transactions was retained which is visible as undulations on the volume chart.

Obviously there are plenty of cases when the daily cycle of transactions may not be accurate. When a petrol station is located near highways, especially in border zones, the freight traffic is significant without regard to time of a day.

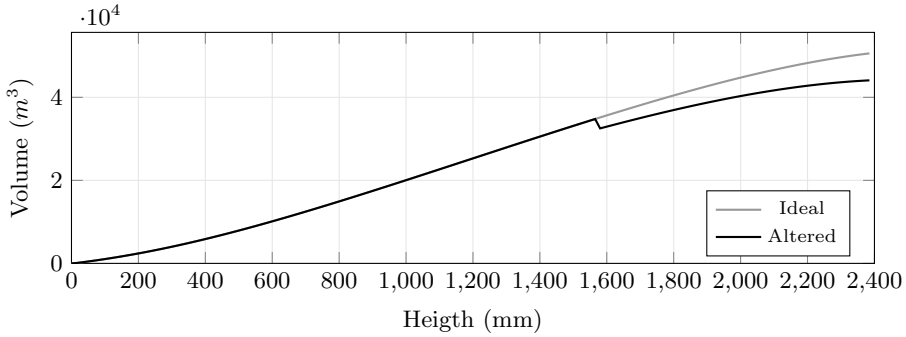

Fig. 2. Calibration curve – ideal and manually altered

On Figure 2 calibration curve from the same tank is presented in its ideal form (i.e. correctly recalculating height of the fuel level to corresponding volume in m^3) and altered in the upper half of the tank. While Figure 1 shows measurements recalculated using the ideal calibration curve, Figure 3 presents the very same tank but in the simulation when the altered calibration curve was used. Course of the volume function in the second case seems similar; however, maximum value is equal to 34 480 litres, while *work level* was set to 46550 litres.

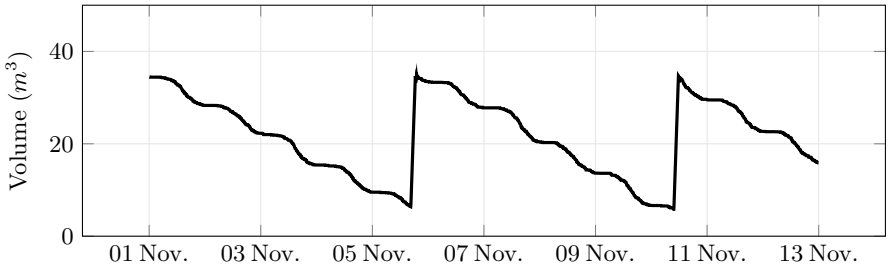

Fig. 3. Fuel volume in the tank affected by the alterations to the calibration curve

The changes to the calibration curve were applied from mentioned 34 480 litres and according to them all the volumes above that level are lowered comparing to those calculated with ideal calibration curve.

In terms of anomaly detection, one of the crucial aspects to the simulator and associated anomaly applier is possibility of applying leakage. Figure 4 presents the same dataset as on Figure 1 with and without a leak. Using the anomaly applier, the leak of 30 litres per hour was applied on the data from simulation no. 1 (Figure 1) starting from the 5^{th} day. The change is clearly visible from the 6^{th} day where CV functions diverge.

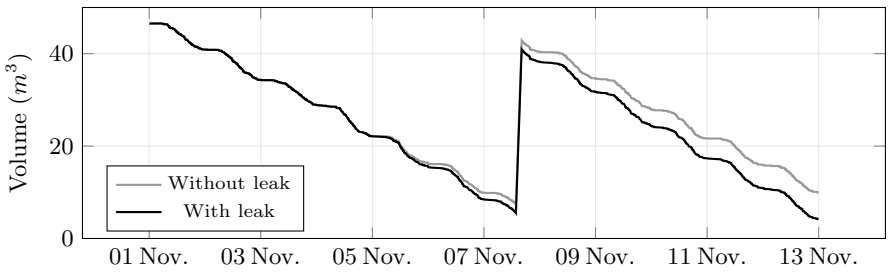

Fig. 4. Simulation no. 1 – data with and without a leak of 30 litres per hour

Unfortunately, the comparison of leak applied to previously simulated data and the new simulation with leak configured seams pointless at this time. Figure 5 presents results of two separate simulations – with a leak applied on the simulated data (the same as on the fig. Figure 4) and another simulation with the same parameters including a leak of 30 litres per hour applied from the 5^{th} day. Simulations are always unique and it is not possible to produce the very same data.

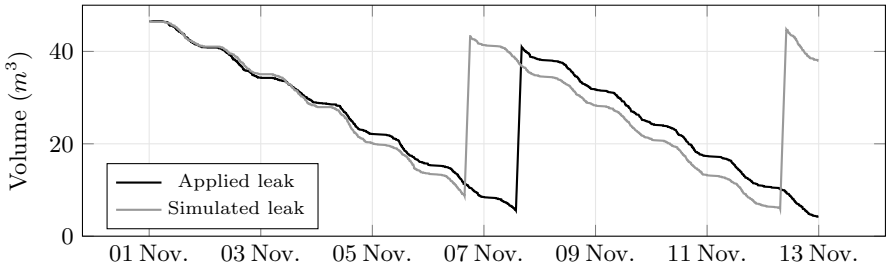

Fig. 5. Simulations no. 1 and no. 2 with leaks of 30 litres per hour

5 Discussion

The process of designing and implementing any software product, such as e.g. anomaly detection system for liquefied petroleum stations, requires proper and reliable data for training and testing purposes. The petrol station simulator has been designed and implemented in order to fulfil this requirement – it generates datasets with various anomalies applied, as well as without them (plain data). Nevertheless, there are still some areas in which the petrol station simulator has to be improved. They are not crucial for the simulator to comply with the requirements and prerequisites (i.e. four compatibility levels) described in section 2; however, they can significantly ameliorate the detailed accordance with a real data source.

Issue 1: Simulation Time Management – Stream Data. In the simulator the internal time resolution is fixed – tempo of a simulation process cannot be arbitrary adjusted. We want to distinguish two distinct times: internal (simulation) time, which controls the whole simulated world (e.g. the frequency of measurements), and the real time, which is observable to the user and affects the duration of a simulation process. Thanks to that we will achieve the most important feature – mimicking a petrol station in real time by producing streams of data [4,6,7,11].

Issue 2: Physical Modelling of Leaks From a Tank. Currently, when a fuel leak from a tank is simulated, from each measurement of stored fuel volume the constant amount of fuel is subtracted. In reality the intensity of a leak directly depends on the fuel pressure exerted on the bottom of a tank. It means that the more fuel tank contains, the more intensively it leaks. Moreover, when a leaking point is located on the side wall of the tank on a certain height, it causes fuel to leak only when the fuel surface is above that point. Thus, according to [1] we propose the variable tank leak model, when the intensity of a leak is a function of the amount of fuel stored in the tank.

Issue 3: Nozzle Miscalibration. On a real petrol station fuel flow meters located in nozzles may miscalibrate over time, which leads to significant corruption of

dispensed fuel measurements. This problem can be successfully modelled using the *linear model* of nozzle calibration: when a particular flow meter suffers from miscalibration, it increases or decreases its measured volume by a constant coefficient.

Issue 4: Fuel Thermal Model. Temperature changes affect significantly the behaviour of fuel on real petrol stations. Due to thermal expansion phenomenon, fuel expands when heated and contracts when cooled. Although, during seasonal changes in weather temperature slow changes in stored fuel volume can be noticed, it is delivery that may trigger more rapid effect. When relatively warmer or colder fuel is being delivered to the tank, it mixes with fuel already present in the tank causing the resulting mixture to rapidly change its temperature and volume. Some time after delivery (usually in few days) fuel tends to return to its normal temperature (accordingly to current weather conditions). Currently, the simulator generates data with a fixed fuel temperature – 15 Celsius degrees. We propose the aforementioned fuel thermal model to be implemented.

Issue 5: Petrol Station Infrastructure Modelling. Currently, the simulator can simulate behaviour of a particular real petrol station by setting various parameter values, e.g.: number of tanks, tank volume or calibration curve. All these virtual devices are fixed, which means that currently the simulator cannot simulate e.g. manifolded tank systems [1]. All internal devices should be individually modelled and simulated, allowing the user to arbitrary build a custom virtual petrol station.

6 Conclusions

Issues 2-5 (presented in section 5) address problems connected directly to the production of the most accurate datasets. However, it is issue no. 1 that refers to the most important future feature of the simulator. The main goal was to develop a reliable data source not only for the anomaly detection purposes. The future version of the simulator will be used in testing and development of the Stream Data Warehouse, especially the stream ETL process [3] and stream materialized aggregate list [8].

Acknowledgments

The authors would like to thank Professor Marcin Gorawski from Silesian University of Technology, Poland for support and mentoring, and undergraduate students Krzysztof Zagórski and Marek Bajorek for their collaboration during the implementation phase.

References

1. EN 13160-5. Leak Detection Systems - Part 5: Tank Gauge Leak Detection Systems, 2005.

2. S. Erkman. Industrial Ecology: an Historical View. *Journal of Cleaner Production*, 5(1):1–10, 1997.
3. M. Gorawski and A. Gorawska. Research on the Stream ETL Process. In *Beyond Databases, Architectures, and Structures*, volume 424 of *Communications in Computer and Information Science*, pages 61–71. Springer International Publishing, 2014.
4. M. Gorawski, A. Gorawska, and K. Pasterak. A Survey of Data Stream Processing Tools. In *Information Sciences and Systems*, pages 295–303. Springer International Publishing, 2014.
5. M. Gorawski, A. Gorawska, and K. Pasterak. Liquefied Petroleum Storage and Distribution Problems and Research Thesis. In *Beyond Databases, Architectures and Structures - 11th International Conference, BDAS 2015, Ustroń, Poland, May 26-29, 2015, Proceedings*, volume 521 of *Communications in Computer and Information Science*, pages 540–550. Springer, 2015.
6. M. Gorawski and P. Marks. Towards Reliability and Fault-Tolerance of Distributed Stream Processing System. In *Dependability of Computer Systems, 2007. DepCoS-RELCOMEX '07. 2nd International Conference*, pages 246–253, June 2007.
7. M. Gorawski and P. Marks. Towards Automated Analysis of Connections Network in Distributed Stream Processing System. In J. Haritsa, R. Kotagiri, and V. Pudi, editors, *Database Systems for Advanced Applications*, volume 4947 of *Lecture Notes in Computer Science*, pages 670–677. Springer Berlin Heidelberg, 2008.
8. M. Gorawski and K. Pasterak. Research and Analysis of the Stream Materialized Aggregate List. In A. Amine, L. Bellatreche, Z. Elberrichi, E. J. Neuhold, and R. Wrembel, editors, *Computer Science and Its Applications*, volume 456 of *IFIP Advances in Information and Communication Technology*, pages 269–278. Springer International Publishing, 2015.
9. M. Gorawski, M. Skrzewski, M. Gorawski, and A. Gorawska. Neural Networks in Petrol Station Objects Calibration. In *Algorithms and Architectures for Parallel Processing: ICA3PP International Workshops and Symposiums, Zhangjiajie, China, November 1820, 2015, Proceedings*, volume 9532 of *Lecture Notes in Computer Science*, pages 714–723. Springer International Publishing, Switzerland, 2015.
10. M. Sigut, S. Alayón, and E. Hernández. Applying Pattern Classification Techniques to the Early Detection of Fuel Leaks in Petrol Stations. *Journal of Cleaner Production*, 80:262–270, 2014.
11. M. Stonebraker, U. Çetintemel, and S. Zdonik. The 8 Requirements of Real-time Stream Processing. *SIGMOD Rec.*, 34(4):42–47, Dec. 2005.
12. United States Environmental Protection Agency. Standard Test Procedures for Evaluating Leak Detection Methods: Automatic Tank Gauging Systems. Final Report, 1990.
13. United States Environmental Protection Agency. Standard Test Procedures for Evaluating Leak Detection Methods: Non Volumetric Tank Tightness Testing Methods. Final Report, 1990.
14. United States Environmental Protection Agency. Standard Test Procedures For Evaluating Leak Detection Methods: Statistical Inventory Reconciliation Methods. Final Report, 1990.
15. United States Environmental Protection Agency. Standard Test Procedures for Evaluating Leak Detection Methods: Volumetric Tank Tightness Testing Methods. Final Report, 1990.