Conference or Workshop Item:
Raza, Fakher and Kazakov, Dimitar Lubomirov orcid.org/0000-0002-0637-8106 (2019) Using ILP to Detect Anomalies in Pipelines. In: The 29th ILP conference, 03-05 Sep 2019, Plovdiv, Bulgaria.

The 29^{th} ILP conference, 3–5 Sep 2019, Plovdiv, Bulgaria.

Using ILP to Detect Anomalies in Pipelines

Fakher Raza and Dimitar Kazakov

Department of Computer Science
University of York, UK

Abstract. Simulated data generated from an accurate modelling tool can demonstrate real-life events. This approach mimics pipeline operation without the need for maintaining the original anomaly record that is already scarce in the pipeline industry. This synthetic data carries precise signatures and the shape of the curve depicts the type of alarm. Learning rules can be inferred from this parametric data and the examples are construed from the threshold levels. The issue is addressed by considering the method that lessens data handling and the associated complexity of the problem. Probabilistic ILPs can be the most appropriate candidate for classifying anomalies of this nature. A logic program addresses this issue by learning the parameters of a program given the structure (the rules), using the ability to incorporate probability in logic programming, interpreting the examples for target predicate, and refining background knowledge for the dissemination of discoveries. This synthetic data also develops a direct link with the ILP parameter learning for competing hypotheses. The modelling tool will receive feedback, check and tune the hypothesis until exclusivity is evolved. This will fall in the domain of closed-loop active learning [1].

Keywords: Pipelines, Anomaly, Hydraulics, Water Distribution, Inductive Logic Programming, ILP, Synthetic Data, Sensor Data

1 Introduction

In machine learning, the aim of the solution is essentially focused on a hypothesis that is dependable on examples. A number of search solutions are available for anomaly detection ranging from modest regression to a complex multivariate classification. Clearly, logic programs are the most appropriate candidate for classifying anomalies of this nature. ILP is valuable in many domains, and an attempt is made to utilise the concepts to another potential industrial candidate for ILP implementation. This paper addresses the anomaly detection issue by considering a technique that minimises the resource complexity of hypothesised logic programs. The nature of the problem is also associated with probability analysis of anomaly occurrences and therefore considered in this approach. The logic and probability fusion has recently evolved in machine learning advancements with encouraging results [2].

2 Background

In the recent past the oil, gas and water industries have suffered several major pipeline calamities [3] with severe impact on health, safety and the environment, which underlined the significance of intelligent technology for pipeline safety.

One of the unique aspects of this research is the use of simulated data generated from a hydraulic simulation model. This approach can mimic unplanned operations such as pipeline shutdown and pipe leaks or bursts. The synthetic data is generated for a variety of pipeline operation permutations including normal steady state, typical transients, severe operation, controlled and unplanned shutdown and any hazardous incident. Due to a sheer volume of data involved, a classifier is believed to be the best-suited approach. The solution requires probability logic to classify the anomaly location. The purpose of the approach is the detection of the location of anomalies which may be difficult to distinguish with the conventional methodologies.

3 The Simulator

VariSim^{TM} is an advanced hydraulic simulator for the pipeline industry that accurately simulates the steady-state and dynamic hydraulic behaviour of any fluid within a pipeline network. It achieves this through explicit equations of state for the fluid and the detailed simulation of operating equipment, control system or telemetry behaviour [4].

The objective is to ensure that any operation that can affect the hydraulic behaviour of a pipeline can be accurately modelled. The simulator works on the principle of constraints and the degree of freedom. Pressures, flows and temperatures are forced on some points and calculated on the other.

Network selection is based on the complexity of the pipe network. A large variety of pipeline network topology is installed worldwide i.e., a simple, moderate, branched or complex network. For this analysis, an A to B water pipeline network is selected. The entry point of the network is a pumping station feeding water to a terminal station, 10 km from the starting point. Pipe diameter is 24 inch and the material of the pipe is carbon steel.

A number of simulation scenarios and approaches are reviewed. A scenario may include burst at every pipe location, more than one location on a pipe, the variable flow rate for one location, pump trips as a result, and/or combination of all possible scenarios. 47 burst locations were selected with 11 different flow rate and operations. The speed of simulation and calculation steps are also be set in the simulator at this stage.

This stage involves the design and deployment of the data generating approach within the simulation software tools. The simulator allows the scripting mechanism to store a timeline of the events to be simulated. The event configuration is the time location and the volume of the anomaly. Associated flow, pressure or temperature constraints are also set. The simulator runs unsupervised to induce anomalies at specified timings and pressure trends are collected as time series data in the form of csv files.

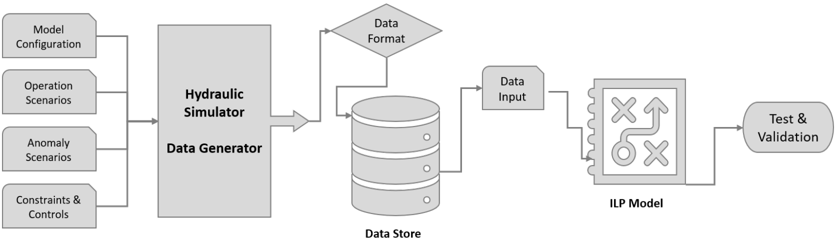

Fig. 1. Data Flow Diagram

Time series data output includes a large volume of calculation points at small time steps. It is imperative to process this data to extract useful patterns for the classification process. Each trend generated from the simulator is processed to extract deviation value which is unique for every sensor at one time. This value is termed as a deviation of a sensor in the event of an anomaly.

4 ILP Model

In this paper, a library called SLIPCOVER is used for structural learning of probabilistic logic programs. SLIPCOVER searches the promising clauses, dividing them into clauses that are intended to be predicted and then clauses for background predicates, with a discriminative approach. The clause search starts from a set of bottom clauses generated as in Progol (Muggleton 1995) [5], and find refinements by monitoring log-likelihood (LL) as heuristic gain. Next, a greedy search is initiated in the space of theories to combine each clause for a target predicate to the existing theory. To the very end, the parameter learning with EMBLEM is performed on the best target theory and the clauses for background predicates.

In this solution, a series of anomaly patterns are identified with the help of Probabilistic Inductive Logic Programming (PILP) routines. The shape of the sensor data in the event of an anomaly is processed logically as a learning example for the prediction models. ILP analysis enhances the understanding of the problem to a common audience by presenting logical examples without compromising the prediction accuracy. The approach collaborates well with other hypothesis-based techniques and therefore augments the potential use of ILP to wide-ranging anomaly detection problems.

Another test was performed with principal component analysis (PCA). The testing platform is SWI-Prolog. The predicate performs the EM algorithm with the assumption that there are 3 clusters and that the four features of the data set are independent given the cluster. The accuracy of clustering was tested with PCA of the data set using the first two principal components (those that account for the highest amount of variability in the data). The result is plotted with the colour of points indicating their class. Comparison plots are drawn, one with the original data set and the other showing assigned clusters computed by the EM algorithm. Both groups are compared for similarity. The program is then tested with test data saved as a subset of the original data set. A total of 20 values were tested for 3 different locations. All data points appeared in their original unseen clusters.

5 Results and Findings

SLIPCOVER program was tested on a test set with a list of terms representing clauses and folds. This returns the log-likelihood of the test examples. The Area Under the ROC curve reflects the accuracy of the learning algorithm.

The PCA analysis has been performed on a full set of original data. Mean and variance of individual data is taken from the overall mean and variance. The individuals are classified into the most probable cluster and the cycle is repeated on the new values. Ultimately, two groupings, the original data and the final PCA results are compared. The two groups present a similar distribution. Test data points are shown distinctly well within their expected clusters.

6 Conclusions

The approach adopted in this paper can be applied to anomaly detection problems in a number of ways. For example, the use of multiple signatures for pinpointing the anomaly location requires extensive data generation from the simulator. This approach can be optimised, for example, by making use of database queries for a larger network to augment the output quality and performance. To recap, we believe that the proposed methodology in this paper breaks new ground in the anomaly detection domains for understanding the value of synthetic data generation and use of machine learning based on logic programming.

References

1. Bryant, CH & Muggleton, S.S.M.: Combining active learning with inductive logic programming to close the loop in machine learning. (1999)
2. Riguzzi, F.: Foundations of probabilistic logic programming. (2018)
3. PHSMA, U.S.: Incident database. United States Department of Transportation. (2019)
4. Fox, J.A.: Hydraulic analysis of unsteady flow in pipelines. The Macmillan Press Limited (1977)
5. Bellodi, Elena & Riguzzi, F.: Structure learning of probabilistic logic programs by searching the clause space. Theory and Practice of Logic Programming. (2013)