Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol. 7, No. 3, September 2019, pp.1030-1040
Available online at: http://pen.ius.edu.ba

Intrusion detection system in gas-pipeline industry using machine learning

[1] **Ali Hasan Dakheel,** [2] **Awfa Hasan Dakheel ,** [3]**Haider Hadi Abbas**
[2]College of Basic Education, Department of math. & computer, Computer science, University of Babylon, Iraq
[3]Computer Technology Enginering Department, Al-Mansour University College (MUC), Iraq

Article Info

Received Feb 14, 2019

Keyword:

Gas-Pipeline
Machine learning
Optimization
Time Interval
Intrusion Recognition
classification
Feature Extraction

ABSTRACT

In this paper, we study about the plausibility of building up a total intrusion identification framework for gas pipeline industry utilized in present day man-made AI based frameworks to tell a gas controller of unexpected changes in pipeline working qualities, for example, weight, time interim, delta pipeline PSI and stream rate. This examination assesses the possibility for utilizing AI example of cautions strategies utilizing three able AI calculations, for example, Decision tree, K-Nearest Neighbor and Neural Network to recognize breaks in gas frameworks, like the SCADA rate of progress blend philosophy utilized by the risky fluids pipeline industry. The highlights were extricated from the dataset by evacuating the repetitive information too cleaning the information. The significant commitment to this work is by utilizing choice tree in three distinct degrees for example randomized, advanced and timberland just as utilizing the neural system with 3 layers, 20 units for each concealed layer, 20 preparing rounds and with 2 layers, 50 preparing rounds as appeared in down to earth some portion of this work executed in Matlab R2019a to recognize and foresee the potential assault in the gas pipeline industry. The idea of AI examination considered here shows guarantee, in light of the aftereffects of gas pipeline burst checking under the conditions tried. It can possibly be formed into a compelling crack checking technique, yet additionally testing under genuine world, complex-framework setups, in participation with appropriate AI demonstrating specialists, is expected to all the more likely comprehend the genuine practicality of this adjustment mechanized innovation. Since gases and fluids display diverse physical practices under changing weight and stream conditions an immediate relationship between's the viability in gas versus fluids frameworks can't be accurately expected that why AI would give the answer for variety in Pipeline PSI and complete delta pipeline PSI. For example, by-passing and back-feeding, and various other framework explicit conditions requiring redid arrangements utilizing AI.

Corresponding Author:

Ali Hasan Dakheel,
Computer Science,
University of Babylon, Iraq
Email: ali.dakeel19715@gmail.com

1. Introduction

The main focus of this study is to review historical rupture data from gas transmission operators to determine the effectiveness and feasibility of this application for natural gas pipelines and make recommendations based on the findings of possible attack in the gas pipeline industry which has to be detect through some automatic means like machine learning using some pre-existing publically available dataset. In addition, this effort demonstrates the industry's eagerness to collaborate to find a practical solution for natural gas attack detection that works well and can be effectively managed and maintained with reasonable effort and cost. Rapid detection of an information leak on natural gas pipelines is difficult, because of the compressible nature of natural gas and how it is transported through different means. Existing techniques such as Real Time Transient Models (RTTM) are capable of providing internal-based information leak detection on natural gas pipelines [3]. However, due to the engineering effort to configure and maintain these systems, along with the continuous changes that are common for interconnected natural gas transmissions networks, RTTMs may not be practical for all operations. Typical volume-balance Computational Pipeline Monitoring (CPM) systems used commonly on liquid pipelines are ineffective and prone to false alarms due to the physical nature of natural gas under pressure to predict the intrusion in gas pipeline industry. These applications are SCADA based (i.e. not requiring a hydraulic model), use existing pipeline instrumentation, and only require small enhancements to traditional SCADA to provide the logic needed for the monitoring of gas transmission pipelines for information leak or any kind of possible attack from outside in the industry. Therefore, there is a higher likelihood that the methodology would be adopted by the industry. This could help to ensure information reliability and scalability are reliably recognized and responded to as quickly as possible using some well-known techniques of machine learning to atomize this process of detecting the intrusion from outside in gas pipeline industry. SCADA systems often use a Rate of Change (ROC) alarm to notify controllers of sudden changes in pressures and flows as well as any kind of attack. By extending ROC alarm functionality to include a "pattern of alarms" concept, the resulting composite alarm can be used to identify leak events and ruptures while reducing false or nuisance alarms caused by normal operational activities.

Advanced alarm management best practices encourage the use of "pattern of alarms" techniques to reduce nuisance alarms and replace them with a more meaningful alarm that better relates to the variables being evaluated. The application being studied in this research project uses a pattern of rate of change alarms, and is referred to as "Rate of Change Combination". This combination of alarms works well with a classic pattern of measurement responses for a liquids pipeline rupture and attacks. This study will evaluate whether the attack/intrusion in the gas pipeline industry can be detected as provided in the publically available dataset using different machine learning based techniques, this combination of application is equally useful in detecting natural gas pipeline ruptures and attacks. In a liquids pipeline intrusion, the upstream pressure drops, the upstream flow increases while the downstream flow rate drops. Using each of these values as inputs to a simple detection method, or combinations of any two of these inputs as a composite attack prediction can result in detection in normal operations. When all three of them happen within a short time period of each other they represent the signature of an intrusion or attack from outside. With potentially several inputs configured for individual rate of change monitoring for attacks, the next step is to assign them to a specific pipe segment or region, so that when the rate of change conditions matches the defined logic for all three algorithms, the application generates a higher priority prediction to indicate that all the defined conditions that may indicate an information leak or attack have been detected in the provided dataset. We know you are challenged daily with complex projects that are increasingly risky. For this reason, Honeywell has revolutionized oil and gas automation and safety projects, and achieved capital savings by as much as 30% by taking automation off the critical project path. Parallel engineering and standard cabinet design along with three enabling technologies—Cloud Engineering, Universal Channel Technology and Virtualization—are key elements of Honeywell's Lean automation engineering methodology.

1.1 Lower capital and operating costs

With Honeywell's Experian PKS, every aspect of your operations can be fully integrated to eliminate unwanted downtime and improve performance over the lifetime of your assets. Together with our advanced software, smart instrumentation, accurate metering and regulating portfolio, and comprehensive range of products and systems for inventory management, pipeline, and terminal operations, Honeywell is transforming operations and business performance for oil and gas companies all over the world.

1.2 Seamless legacy SCADA migration

When your legacy SCADA becomes challenging to support or is restricting your business model, turn to Honeywell for seamless migration solutions for all legacy SCADA systems. Our advanced SCADA technology provides many advantages over legacy SCADA systems, including improved business fly, better compliance and lower costs. Honeywell regularly migrates competitors' SCADA platforms to Experian SCADA. We provide custom applications and integration with other applications and systems, and can bring this experience to benefit your enterprise.

1.3 Background

Data from actual pipeline ruptures and attacks were provided confidentially by dataset [4]. As the application being evaluated allows for four inputs that can be used collectively to identify a rupture or attack signature while reducing or eliminating false alarms, the request for data was to analyze real world pipeline data [1], that when used in combination, can confidently recognize the attacking condition. A single pressure point is of limited value, so we asked for "related" pipeline inputs, ideally upstream and downstream pressures and flow rates where possible. In cases where flow rate data was not available, we used multiple related pressure data points monitored as a group represented by delta pipeline PSI. Data provided included a brief description of the pipeline layout, such as where the inputs were relative to the rupture location. (i.e. PS1 is located approximately x miles upstream of attack location etc.) Additional pipeline information requested included fundamental details such as pipe alignment, nominal operating pressure and how the leak/rupture was determined. It should be emphasized here that the research focused on finding out if the machine learning application had the capability to detect the attack condition and not on the robustness of the application with respect to generating false positive alarms under certain operating conditions. As a result, the data collected from dataset was associated with specific attack incidents and not special operating conditions that potentially could have triggered a false positive as described in article [5]. The first step was to identify and configure a robust "rate of change" monitor for each of the inputs. Rate of change in SCADA has traditionally been "noisy," causing many false alarms due to the uncertainties of poll times, fast or interrogate scanning, and latency of the data. The study provides a more reliable rate of change evaluation by using a configurable number of samples versus a fixed time to evaluate the rate of change in the dataset after cleaning the data. The inputs are intended for pressure and flow in case of attacking situation, but the same algorithm could be used for any input: control valve position, compressor rpm, etc. With the Rate of Change Combination record fully configured, we start the SCADA polling of the points through a simulated remote telemetry unit (RTU) that reads the data provided as time-series data in real time. This enables us to represent live data in the machine learning application as it occurred in the rupture event and evaluate the effectiveness of the application as explained in article [5].

With the data being polled, each of the configured points are monitored by comparing the value of the first sample against the last value and dividing by the time span between them to calculate the rate of change of the process. These points are normally not seen, nor alarms generated, when alarm violations occur because the goal of the application is to eliminate the individual alarms and look for alarms that occur in the predetermined combination for the machine learning algorithms.

Table 1: The methodologies that have been used earlier suing different techniques of machine learning as a reference to gas pipeline industry for predicting the intrusions.

Method	Reference
Adaptive Vector Quantization	(Chaczykowski, M & Sumaili Akilimali , 2010) [1]
Follow-The-Leader (Fdl)	(Rios-Mercado, R.; Borraz-Sanchez et al., 2015) [5]
Fuzzy And Arima	(Pambour, K.A.; Bolado-Lavin, et al., 2016) (2016) [4]
K-nearest neighbor (KNN)	(Wang, P.; Yu, B.; Han, D.; Li, J.; Sun, D.; Xiang, Y et al., 2018) [8]
Probabilistic Neural Network (PNN)	(Gato, L.; Henriques, J et al., 2015) [7]
Self-Organizing Map (Som)	(Gato, L.; Henriques, J et al., 2015) [7]
Support Vector Machine (Svm)	(Sundar, K.; Zlotnik, et al., 2018) [9]

The table above presented many different methods that have distinct clustering and classification approaches, it can be found: from agglomerative techniques like the Adaptive Vector Quantization where there is no need to previously set a number of clusters; to partitioning methods such as K-nearest neighbor where the number of classes is necessary to be previously set. The fuzzy methods which states that each instance has a certain degree of belonging to a group, even they can belong to more than one group; instead of assigning the instances to a specific group. The unsupervised learning- based Support vector machine; the supervised ones like the neural networks and some statistical based like the multivariate statistics. Other more recent techniques are the Entropy-based or the Support Vector Machine [6,7].

2. Methodology

The data set consists of 12 inputs that cover a thirty-minute time interval, updated once a minute. The dataset represents normal operations for seven minutes before the attack and 23 minutes after the attack. The pipeline PSI data supplied represents five interconnected main lines. In reality, this attack was recognized when the difference between the pressures between two of the interconnected lines exceeded a configurable alarm threshold using conventional machine learning functionality. Test configuration for dataset consisted of the setup of four selected pipeline inputs to be configured as the machine learning algorithms records to be used in combination as well as the process of feature extraction and converting format is shown in figure 1. The Matlab application provides a more reliable rate of change evaluation by using a configurable number of samples versus a fixed time to evaluate the rate of change used to compute the accuracy of detection for all three algorithms that have been used as a part of work.

- **ROC violation limit** is the value that the rate of change has to exceed in either negative or positive direction to create a data cleaning space for different values from dataset. The value needs to reflect the point chosen to use for region of consideration for the detection of attack.
- **ROC suppression limit** is the value that the rate of change has to exceed in either negative or positive direction to suppress the ROC calculation to predict the accuracy of detection.
- **Sample Count** is the number of samples that are used for the ROC calculation to predict the accuracy of detection.
- **Exceed limit time duration** is the amount of time a ROC stays in violation or suppression once it enters that state.

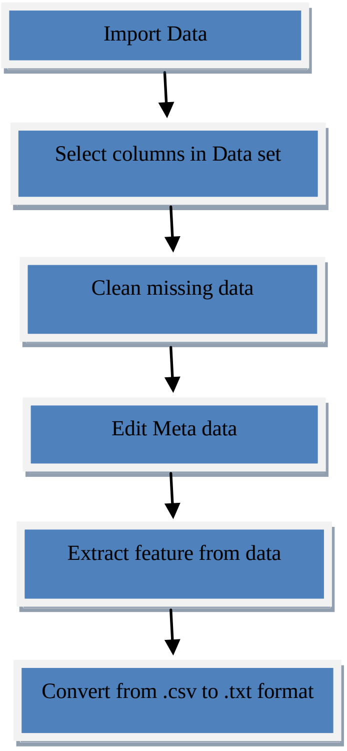

Import Data
Select columns in Data set
Clean missing data
Edit Meta data
Extract feature from data
Convert from .csv to .txt format

Figure 1: The process of extracting feature and converting its format.

All the individual algorithms consider detection any attack triggered at roughly the same time. Similarly, the rate of change was triggered in time interval seconds after the first indication at a pressure measurement was detected in pipeline PSI. This is understandable from looking at the pressure trend for the same four rate of change points in delta pipeline PSI. Dataset analysis indicates that the algorithms would have generated the rate of change alarm for the attacking event. This dataset does exercise the full capabilities of the application, as the simultaneous pressure drop off all inputs would be obvious to the gas controller using standard rate of change for attacking situation generation [8]; however, the application does ensure it will not be missed, and provides additional confirmation of the intrusion event. A first stage previous to the analysis, covers the data collection of the gas pipeline variation consumption of demand of gas and also the pipeline PSI and delta pipeline PSI characteristics, understanding how the data is presented and reshape it at convenience for the study. Once getting familiar with the dataset and having seen the problematic; a process of cleaning the data is carried out to remove the bad data (zeros, frozen, duplicated features). Closing this stage, a graphical representation is sought as a data exploration and visualization to go deeper on the data used for the study, and also help to check its validity. The pre-classification phase consists on doing a literature research of similar studies, selecting the most adequate format of the input data to allow a fair comparison among the gas industries load profile of the users, and processing the input data in that sense obtaining the proportional pipeline PSI usage per hour for each user, in percentages.

In the classification phase; a review on the existent classification techniques used for gas pipeline industry intrusion based data segmentation is done. K-nearest neighbor, decision tree plus random forest and neural network techniques are selected and applied individually in an iterative process, based on computational classification calculation in order to find the optimal number of classes with the appropriate number of members in each of them. Finally, a visual and statistical comparison of the results of each classification technique is performed to determine which solution is the most suitable [9]. The final phase, is the post-classification; which is devoted to obtain the final and desired intrusion segmentation by the redistribution of the different parameters of pipeline PSI as well as delta pipeline PSI. This is performed by a manual intervention done by the analyst applying visualization and statistical techniques to be able to find the outliers and reallocate them to a more appropriate group. To perform any data analysis the data quality is essential also the preparation of this data into the right format is the key to start the analysis sought. Hence, it is important to know beforehand the purpose of the analysis by having one or other purpose the procedure to go into the data processing may change completely. A good data preparation will also report better and more accurate results.

3.1 Decision tree and random forest

Random forest is an ensemble machine learning method that that is typically used for classification and regression tasks. This ensemble learning method combines multiple uncorrelated decision trees and the predictions are obtained by combining the results from each decision tree. The combination of multiple classifiers reduces the model over-fitting, thus increasing the classification accuracy. The technique is used to re-sample the data and to generate different training sets for each classifier with replacement of the data [10]. A decision tree performs classification by recursively partitioning the data and is composed by three types of nodes: root node, internal node and leaf nodes (yes and no outcomes), as shown in Figure 2. The root and internal nodes represent a test on an attribute/feature and the branch represents the outcome of that test. The tree is partitioned until no more tests are required for the algorithm to perform the classification. The tree ends with the leaf (or terminal) node that contains the classification outcome to predict the data provided in dataset as intrusion based or attacking for given instance.

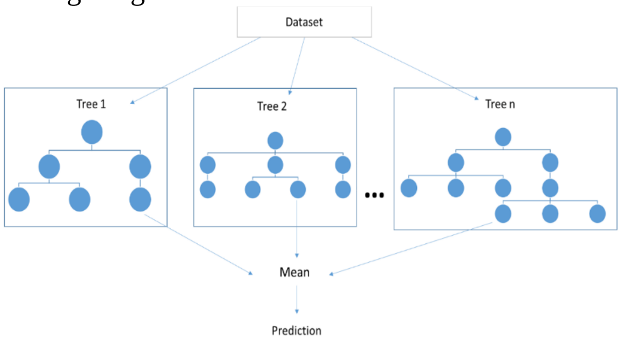


Figure 2: Decision tree and random forest combined hierarchy for the mean as well as prediction of intrusion.

3.2 K-Nearest neighbor

We pick the K-nearest neighbor classification algorithm to make an inhabitance probabilistic classes, which is then tried later on set. KNN is an efficient machine learning algorithm which is widely used for pattern recognition and classification problems such as face recognition, gene extraction and speaker identification and the structure of KNN is shown in Figure 3 being used in this work. It can be used to perform both linear and non-linear classification through kernel methods and performs well even with less training data samples [10]. The likelihood of a gas pipeline being under attack in a given situation and timespan is registered by partitioning the quantity of involved time interval by the absolute number of existent periods in the individual time and weekday of the arrangement set. For instance, if our characterization set has low PSI of pipeline than there is a chance of information leak, and if in this data set the pipeline was constantly delegated involved, at that point the likelihood of essence in this period during classification of potential attacks in gas pipeline industry. The K-nearest neighbor classification was then processed by applying the condition over every interim of the pipeline PSI as well as delta pipeline PSI from the classes as the value of K usually varies from 1 to 5 in this particular work.

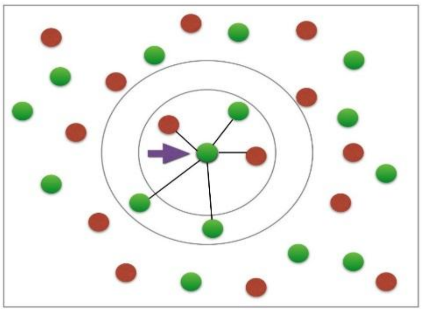


Figure 3: The K-nearest neighbor with value of k is 1, 3 and 5 in our case of study to classify the attack in gas pipeline industry.

3.3 Neural Network

Artificial neural network (ANN) is an artificial intelligence technique that is inspired in the human brain to solve problems. It is often used to solve both supervised and unsupervised classification problems (e.g. prediction of information leak and potential gas pipeline attack) and have been applied in many applications, such as: bankruptcy prediction, fault detection, speech recognition and product inspection [10]. The ANN structure is composed by three types of nodes: input, hidden and output nodes, as shown in Figure 4. The input nodes represent the nodes that receives the input information and emit signals to other nodes so, we processed it by given multiple instances as an input that feeds into the hidden layers (using 2-3 layers as hidden layers) following into the prediction of potential possible attack or any kind of information leak in dataset. The output nodes receive information from the network nodes and sends it as output to the environment. The hidden nodes are between the input and output nodes and do not interact directly with the external sources, i.e., do not receive information from the external environment and also do not outputs information to the environment to predict the potential attack or information leak in the industry.

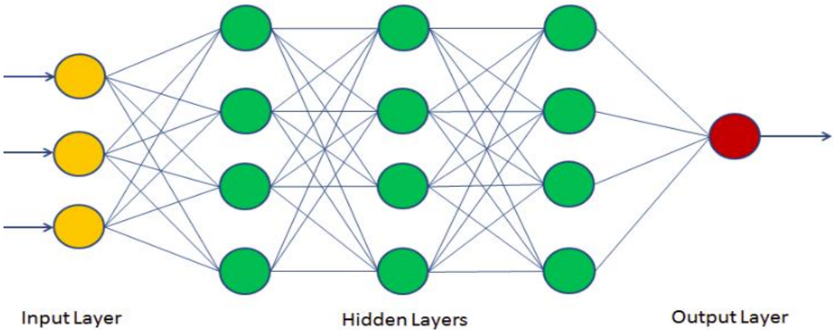


Figure 4: Illustrative structure of a neural network with approach using two hidden layers and multiple nodes processed through layer to predict the potential attack.

3.4 Prediction Evaluation

To evaluate the prediction performance of the probabilistic data provided in data set, it is first necessary to convert the probabilistic data into a binary probabilistic data for cleaning the data for processing and evaluation. This is done by choosing a cut-off value as our decision rule (or threshold). To this end, we first analyze the effect of choosing different cut-off values through the receiver operating characteristic for predicting the potential attack in gas pipeline industry. Then, we use the accuracy to evaluate the prediction performance.

3.5 Prediction Accuracy

After defining a threshold equal to process the data set through different algorithms, we consider that, whenever the probability of any possible attack in gas pipeline industry, in a certain time period, the pipeline is occupied in the respective period (and is unoccupied otherwise). Therefore, the probabilistic data presented in data set as a pipeline PSI is converted to a binary probabilistic data. To evaluate the prediction performance of our algorithm, we use the metric accuracy [11], given by equation (1), that we call here for convenience.

$$Prediction\ Accuracy = \frac{tp+tn}{tp+tn+fp+fn} \qquad (1)$$

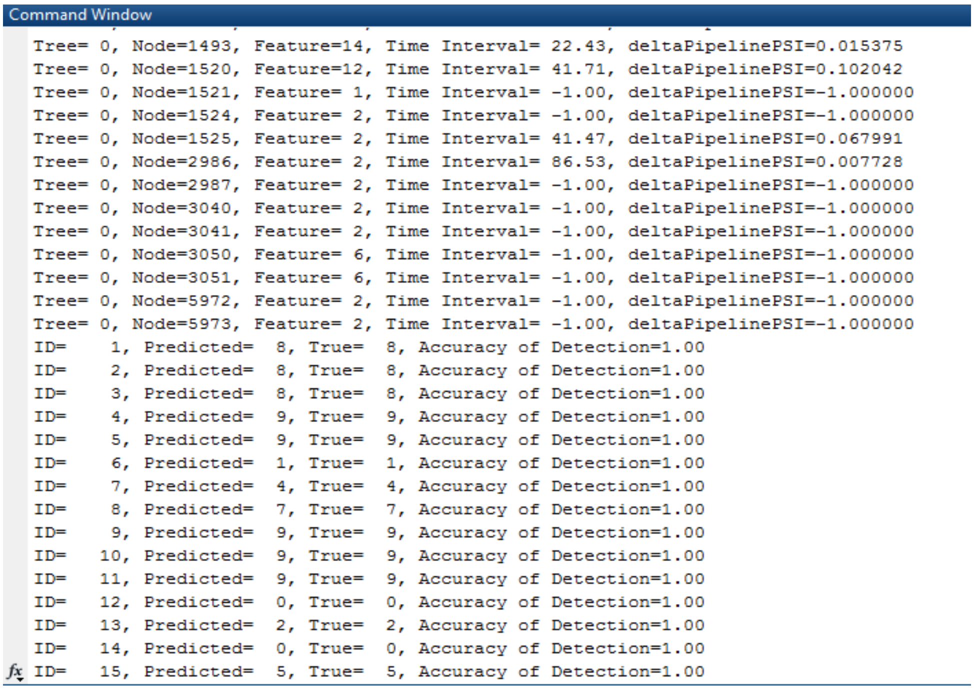
4 Results

In this work, by analyzing results of the three different algorithms performed in Matlab as a supplementary research to find ruptures, information leak and potential attacks in pipeline that were simulated through the application and reviewing the trends provided with the dataset, we can predict that the application would trigger an alarm on the rupture, information leak and potential attack events for the data provided, as pressure and pipeline PSI changes were significant. As such, we made a decision to limit the effort of fully configuring and inputting data to data set at this time. In hindsight, it is clear that the application can detect ruptures, information leak and potential attack in pipeline, which is surprising since the industry has been using rate of change methodology to detect ruptures, information leak and potential attacks for a long time. The application has the added benefit of eliminating many of the nuisance alarms by using such methodology. Datasets could have been investigated to prove this hypothesis further; however, at the time, it was not considered a priority for the study.

4.1 Decision Tree and Random Forest

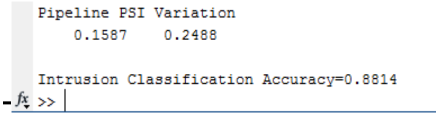
As can be seen from the accuracy of detection in Figure-5 ID’s with respect to the instances of data set, there was some time between the start of the data and the actual rupture, information leak or any kind of intrusion incident. During this time period the pipeline PSI is also measured using decision tree and random forest methodology for the flow measurement to detect any kind of intrusion in the dataset instances which exceeded the violation limit and generated an internal prediction of pipeline PSI variation and intrusion classification accuracy using decision tree and random forest methodology as seen in Figure-6.

```
Command to run:
Run the main.m file
   main(dtree , 'features_training.txt', 'features_testing.txt', 'randomized', 50)
   main(dtree , 'features_training.txt', 'features_testing.txt', 'optimized', 50)
   main(dtree , 'features_training.txt', 'features_testing.txt', ‘forest’, 50)
```



```
Tree= 0, Node=1493, Feature=14, Time Interval= 22.43, deltaPipelinePSI=0.015375
Tree= 0, Node=1520, Feature=12, Time Interval= 41.71, deltaPipelinePSI=0.102042
Tree= 0, Node=1521, Feature= 1, Time Interval= -1.00, deltaPipelinePSI=-1.000000
Tree= 0, Node=1524, Feature= 2, Time Interval= -1.00, deltaPipelinePSI=-1.000000
Tree= 0, Node=1525, Feature= 2, Time Interval= 41.47, deltaPipelinePSI=0.067991
Tree= 0, Node=2986, Feature= 2, Time Interval= 86.53, deltaPipelinePSI=0.007728
Tree= 0, Node=2987, Feature= 2, Time Interval= -1.00, deltaPipelinePSI=-1.000000
Tree= 0, Node=3040, Feature= 2, Time Interval= -1.00, deltaPipelinePSI=-1.000000
Tree= 0, Node=3041, Feature= 2, Time Interval= -1.00, deltaPipelinePSI=-1.000000
Tree= 0, Node=3050, Feature= 6, Time Interval= -1.00, deltaPipelinePSI=-1.000000
Tree= 0, Node=3051, Feature= 6, Time Interval= -1.00, deltaPipelinePSI=-1.000000
Tree= 0, Node=5972, Feature= 2, Time Interval= -1.00, deltaPipelinePSI=-1.000000
Tree= 0, Node=5973, Feature= 2, Time Interval= -1.00, deltaPipelinePSI=-1.000000
ID=    1, Predicted=  8, True=  8, Accuracy of Detection=1.00
ID=    2, Predicted=  8, True=  8, Accuracy of Detection=1.00
ID=    3, Predicted=  8, True=  8, Accuracy of Detection=1.00
ID=    4, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=    5, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=    6, Predicted=  1, True=  1, Accuracy of Detection=1.00
ID=    7, Predicted=  4, True=  4, Accuracy of Detection=1.00
ID=    8, Predicted=  7, True=  7, Accuracy of Detection=1.00
ID=    9, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=   10, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=   11, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=   12, Predicted=  0, True=  0, Accuracy of Detection=1.00
ID=   13, Predicted=  2, True=  2, Accuracy of Detection=1.00
ID=   14, Predicted=  0, True=  0, Accuracy of Detection=1.00
fx ID=   15, Predicted=  5, True=  5, Accuracy of Detection=1.00
```

Figure 5: The decision tree and random forest accuracy of detection with multiple nodes and trees formed as well as time interval extracted from the features extracted from dataset for optimized mode.



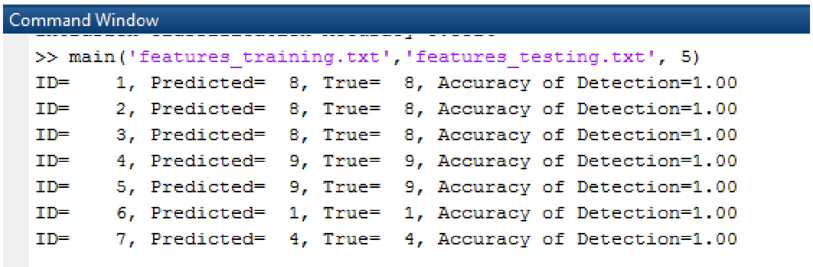
```
Pipeline PSI Variation
    0.1587    0.2488

Intrusion Classification Accuracy=0.8814
fx >>
```

Figure 6: The pipeline PSI variation and intrusion classification accuracy for gas pipeline dataset using decision tree and random forest.

4.2 K- Nearest neighbor

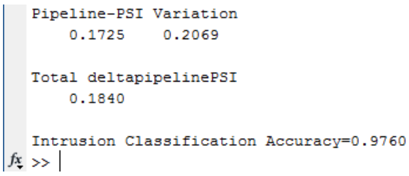
As can be seen from the accuracy of detection in Figure-7 ID's with respect to the instances of data set, there was some time between the start of the data and the actual rupture, information leak or any kind of intrusion incident. During this time period the pipeline PSI is also measured using K nearest neighbor for the flow measurement to detect any kind of intrusion in the dataset instances which exceeded the violation limit and generated an internal prediction of pipeline PSI variation, total delta pipeline PSI and intrusion classification accuracy using K nearest neighbor methodology as seen in Figure-8.

Command to run:
Run the main.m file
main('features_training.txt','features_testing.txt', 1)
main('features_training.txt','features_testing.txt', 3)
main('features_training.txt','features_testing.txt', 5)



```
>> main('features_training.txt','features_testing.txt', 5)
ID=    1, Predicted=  8, True=  8, Accuracy of Detection=1.00
ID=    2, Predicted=  8, True=  8, Accuracy of Detection=1.00
ID=    3, Predicted=  8, True=  8, Accuracy of Detection=1.00
ID=    4, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=    5, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=    6, Predicted=  1, True=  1, Accuracy of Detection=1.00
ID=    7, Predicted=  4, True=  4, Accuracy of Detection=1.00
```

Figure 7: The k-nearest neighbor accuracy of detection with number of k is set to 5.



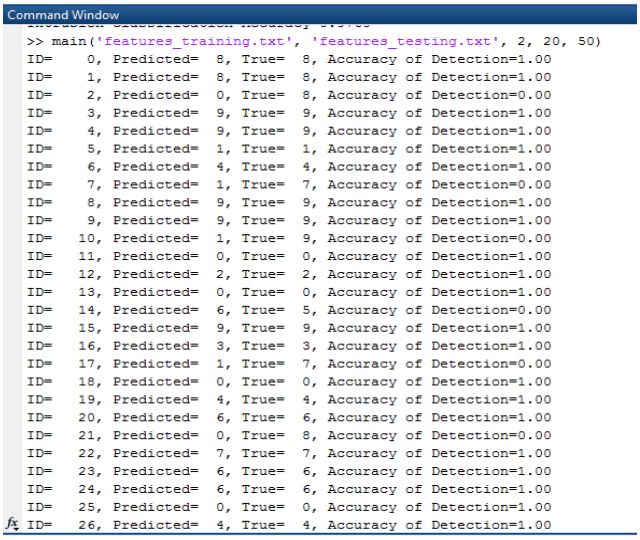
```
Pipeline-PSI Variation
    0.1725    0.2069

Total deltapipelinePSI
    0.1840

Intrusion Classification Accuracy=0.9760
>>
```

Figure 8: The pipeline PSI variation, total delta pipeline PSI and intrusion classification accuracy for gas pipeline dataset using k-nearest neighbor.

4.3 Neural Network

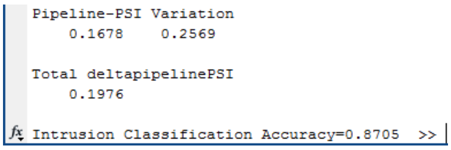
As can be seen from the accuracy of detection in Figure-9, ID’s with respect to the instances of data set, there was some time between the start of the data and the actual rupture, information leak or any kind of intrusion incident. During this time period the pipeline PSI is also measured using neural network methodology for the flow measurement to detect any kind of intrusion in the dataset instances which exceeded the violation limit and generated an internal prediction of pipeline PSI variation, total delta pipeline PSI and intrusion classification accuracy using neural network methodology as seen in Figure-10.

Command to run:
Run the main.m file
main('features_training.txt', 'features_testing.txt', 2, 20, 50)
where the argument format is : "main(training file, testing file, layers, units, rounds)"



```
>> main('features_training.txt', 'features_testing.txt', 2, 20, 50)
ID=    0, Predicted=  8, True=  8, Accuracy of Detection=1.00
ID=    1, Predicted=  8, True=  8, Accuracy of Detection=1.00
ID=    2, Predicted=  0, True=  8, Accuracy of Detection=0.00
ID=    3, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=    4, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=    5, Predicted=  1, True=  1, Accuracy of Detection=1.00
ID=    6, Predicted=  4, True=  4, Accuracy of Detection=1.00
ID=    7, Predicted=  1, True=  7, Accuracy of Detection=0.00
ID=    8, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=    9, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=   10, Predicted=  1, True=  9, Accuracy of Detection=0.00
ID=   11, Predicted=  0, True=  0, Accuracy of Detection=1.00
ID=   12, Predicted=  2, True=  2, Accuracy of Detection=1.00
ID=   13, Predicted=  0, True=  0, Accuracy of Detection=1.00
ID=   14, Predicted=  6, True=  5, Accuracy of Detection=0.00
ID=   15, Predicted=  9, True=  9, Accuracy of Detection=1.00
ID=   16, Predicted=  3, True=  3, Accuracy of Detection=1.00
ID=   17, Predicted=  1, True=  7, Accuracy of Detection=0.00
ID=   18, Predicted=  0, True=  0, Accuracy of Detection=1.00
ID=   19, Predicted=  4, True=  4, Accuracy of Detection=1.00
ID=   20, Predicted=  6, True=  6, Accuracy of Detection=1.00
ID=   21, Predicted=  0, True=  8, Accuracy of Detection=0.00
ID=   22, Predicted=  7, True=  7, Accuracy of Detection=1.00
ID=   23, Predicted=  6, True=  6, Accuracy of Detection=1.00
ID=   24, Predicted=  6, True=  6, Accuracy of Detection=1.00
ID=   25, Predicted=  0, True=  0, Accuracy of Detection=1.00
ID=   26, Predicted=  4, True=  4, Accuracy of Detection=1.00
```

Figure 9: The neural network accuracy of detection with training and testing on features training dataset, with 2 layers, 50 training rounds.



```
Pipeline-PSI Variation
    0.1678    0.2569

Total deltapipelinePSI
    0.1976

Intrusion Classification Accuracy=0.8705  >>
```

Figure 10: The pipeline PSI variation, total delta pipeline PSI and intrusion classification accuracy for gas pipeline dataset using neural network.

5 Discussion

As an outlook, Suggested Improvements to make this application more beneficial for rupture monitoring, predicting the information leak and any future potential attack in industry include:

- Automating the trend data so that the user does not need to manually configure the trends for each input data set for extracting the features. The application should automatically configure trends for the points in the record to extract the feature automatically without any human involvement.
- Creating a warning alarm, so that when part of the data record is in violation, the pipeline controller would get a rupture, information leak or intrusion warning. The controller would then use the machine learning based algorithms trend to watch the changes in the segment, potentially recognizing a rupture, information leak and intrusion before all the logical conditions in the evaluations have been met. This is useful when there is a long distance between the upstream and downstream input pipeline PSI for processing [12-15].
- Currently the application has its own configuration of three algorithms rather than reusing standard configuration for each point found in machine learning based algorithms. This means that points are duplicated within these three algorithms [16]. This should be eliminated going forward so that the standard configuration for each point in algorithms are also used by the application for better performance in detecting any kind of intrusion in the gas pipeline industry.

Table 2: The accuracy of detection compared with the previous work.

Algorithms	Optimal number of hidden neurons	Optimal feature set	Detection Accuracy (%) – *classification set*	Prior accuracy [8] (%)
KNN	NA	2	97.60	93.82
Neural Network	2	2	87.05	86.31
DT and RF	NA	2	88.14	86.29

6 Conclusion

The objective of this work is to study the feasibility of developing a fully automated system for detecting any raptures, information leak and any kind of potential attacks in gas pipeline industry by analyzing the data set provided by Tommy Morris. The best accuracy was provided by the k-nearest neighbor among all three algorithms to predict the intrusion in gas pipeline industry. In other words After analyzing sample rupture, information leak and intrusion based data provided by dataset of gas pipeline industry, we are able to conclude that the Matlab application that comprises of three main machine learning algorithms to predict any kind of intrusion in the probabilistic data set, with further development, may have the potential to recognize ruptures on natural gas pipelines, as well as address the recommendation that suggests providing automatic SCADA system trend data alarms of this type would improve controller recognition of abnormal conditions (such as pipeline ruptures, data leaks and memory loss in the servers), notify the controller to examine the condition and take the appropriate action to respond to it. In addition to the Matlab application that is being analyzed as part of this study, other techniques have been developed that may be worth evaluating for natural gas leak detection. One simple approach that may prove useful is an “alarm bracketing” or clamping method that allows the gas controllers to activate an alarm bracket for a pipeline (group of hydraulically related pressure inputs, pipeline PSI and delta pipeline delta PSI). These read the current pressures for all the points in the bracket group after feature extraction, and creates an operating envelope that matches the operating conditions, and provide an intrusion classification when pressures and pipeline PSI deviate from this envelope without any operational reason. This provides a “simple to implement” rupture monitoring, information leak and intrusion detection application that uses common machine learning based techniques by compressing the dataset for pressures, flows and rate of change. As the program logic is designed for multiple point input to intrusion detection and prediction in gas pipeline industry.

References

[1] Chaczykowski, M. Transient flow in natural gas pipeline—The effect of pipeline thermal model Adaptive Vector Quantization. Appl. Math. Model. 2010, 34, 1051–1067.

[2] Nguyen, H.; Chan, C. Optimal scheduling of gas pipeline operation using genetic algorithms. In Proceedings of the Canadian Conference on Electrical and Computer Engineering using Machine Learning, Saskatoon, SK, Canada, 1–4 May 2012.

[3] Zlotnik, A.; Chertkov, M.; Backhaus, S. Optimal control of transient flow in natural gas networks using machine learning. In Proceedings of the 54th IEEE Conference on Decision and Control, Osaka, Japan, 15–18 December 2015.

[4] Pambour, K.A.; Bolado-Lavin, R.; Dijkema, G.P.J. An integrated transient model for simulating the operation of natural gas transport systems. J. Nat. Gas Sci. Eng. 2016, 28, 672–690.

[5] Rios-Mercado, R.; Borraz-Sanchez, C. Optimization problems in natural gas transportation systems using Follow-The-Leader (Fdl): A state-of-the-art review. Appl. Energy 2015, 147, 536–555.

[6] Behrooz, H.; Boozarjomehry, R. Modeling and state estimation for gas transmission networks using Machine Learning Algorithms. J. Nat. Gas Sci. Eng. 2015, 22, 551–570.

[7] Gato, L.; Henriques, J. Dynamic behaviour of high-pressure natural-gas flow in pipelines using Probabilistic Neural Network and Self-Organizing Map. Int. J. Heat Fluid Flow 2015, 26, 817–825.

[8] Wang, P.; Yu, B.; Han, D.; Li, J.; Sun, D.; Xiang, Y.; Wang, L. Adaptive implicit finite difference method for natural gas pipeline transient flow. Oil Gas Sci. Technol using Neeural Network. Dec-2018.

[9] Sundar, K.; Zlotnik, A. State and parameter estimation for natural gas pipeline networks using transient state data. IEEE Trans. Control Syst. Technol. 2018, 99, 1–15.

[10] Durgut, I.; Leblebicioglu, K. Optimal control of gas pipelines via infinite-dimensional analysis. Int. J. Numer. Methods Fluids 2016, 22, 867–879.

[11] Cortinovis, A.; Mercangoz, M.; Zovadelli, M.; Pareschi, D.; de Marco, A.; Bittanti, S. Online performance tracking and load sharing optimization for parallel operation of gas compressors. Comput. Chem. Eng. 2016, 88, 145–156.

[12] Wen, K.; Xia, Z.; Yu, W.; Gong, J. A new lumped parameter model for natural gas pipelines in state space. Energies 11 June 2017.

[13] B. Durakovic, “Thermal Performances of Glazed Energy Storage Systems with Various Storage Materials: An Experimental study”, Sustainable Cities and Society, vol. 45, pp. 422-430, 2019.

[15] B. Durakovic, “Design for Additive Manufacturing: Benefits, Trends and Challenges”, Periodicals of Engineering and Natural Sciences (PEN), vol. 6, pp. 179–191, 2018.