A PREDICTIVE SYSTEM MODEL FOR OIL AND GAS PIPELINE MONITORING AGAINST POTENTIAL THREAT AND VANDALIZATION

V.I.E ANIREH PHD
AND
E.N. OSEGI

ABSTRACT

We propose an artificial intelligence monitoring system capable of performing predictive classification and pattern recognition on pipeline datasets. The predictive system is based on a sparse predictive Deviant Learning Algorithm (p-DLA) designed to synthesize a sequence of memory predictive data clusters for monitoring, control and for decision making purposes. The system uses two feature pipeline datasets and the pattern recognition monitoring ability of the DLA (p-DLA) is compared with a variant of an emerging machine learning algorithm, the Hierarchical Temporal Memory based on Cortical Learning Algorithms (HTM-CLA). The simulations study shows impressive results and validate the sparse memory predictive approach which underscores the sub-synthesis of a highly compressed and low dimensional knowledge discovery information system. It also shows that this proposed approach compares favourably with the HTM-CLA approach.

Keywords: Artificial intelligence; cortical learning; predictive classification; pattern recognition

1. Introduction

With the rise in militant activity and rogue behaviour in oil and gas regions around the world, oil pipeline disturbances is on the increase leading to huge losses in the economy of multinational operators and the countries where such facilities exist. The continual militancy and vandalism that occur in oil and gas installations have led to high levels of insecurity to industry operators disrupting their operations and as a consequence, bringing production operations to a halt. In line with the need to prevent/or reduce further disruptions to oil and gas operations, worldwide research in the area of oil/gas pipeline facility protection have resulted in the development of many proposals on how to tackle the oil and gas insurgency, in particular, in the Niger Delta region of Nigeria – the oil rich hub of West Africa. Popular among the list of proposed solutions is the use of real time monitoring systems based on several existing and future technologies. For example, the use of overhead and underwater surveillance has been briefly surveyed and wireless monitoring systems proposed in Obodoze (2012) and Obodoze et al (2015) respectively. This has the advantage of real time visualization and transmission but also adds
with it the complexity of processing vast amount of video or image data. Another interesting area is the use of artificial experts and robotics for smart monitoring and sensing in oil fields. The review in (Shukla & Karki, 2016a; Shukla & Karki, 2016b) speaks volumes. In (Castellanos et al 2011a; Castellanos et al 2011b) and in Jawhar et al (2013), the expert systems approach and robotics have been deployed for onshore pipeline analysis and underwater pipeline monitoring respectively. While some systems use wired means of detection, the benefits of wireless monitoring systems cannot be overemphasized (Henry & Henry, 2015).

However, very few literature exists in the area of predictive classification of pipeline data, particularly as it concerns the development and analysis of real world pipeline datasets. As have been shown in Cui et al (2016), Rodriguez-Cobo et al (2013), and in (Otahal & Stepankova, 2014), predictive systems using highly sparse cortical learning algorithms can prove useful in the detection of likely anomalies/or defects particularly in an online and unsupervised data monitoring system. Our contributions are as follows:

In this paper, we propose the use of a predictive memory system based on the Deviant Learning Algorithm (DLA); the memory is obtained using the sparse generative cortical learning algorithm designed for stream data processing and has been introduced earlier in Osegi & Anireh (2016).

Our aim is to validate the effectiveness of a new AI technique (the DLA) for predictive classification of two pipeline datasets that have been developed by other researchers (see Tan et al, 2016 & Lam, 2015). These datasets are chosen due to their interpretability and conformance to real world observations in oil and gas environments.

In addition, we compare the performances of the proposed AI approach with a variant of an emerging and widely applied AI approach the Hierarchical Temporal Memory based on Cortical Learning Algorithms (HTM-CLA).

2. Concept of Predictive Monitoring and Cortical-like Sparse Memory Generation

The idea of predictive monitoring using sparse-generative cortical learning approaches is not entirely new to the machine learning community. In Hawkins & Blakeslee (2007) the idea of memory predictive system was founded which engineered the development of variations of memory predictive architectures referred to now as Hierarchical Temporal Memory; these architectures have been developed in Hawkins et al (2010), Agrawal & Franklin (2014) and McCall & Franklin (2013) which is based on the cortical learning algorithms, and in George and Hawkins (2009) which is based on the Bayesian Belief Network (BBN). Sparse coding originally discovered and formally presented in Olshausen & Field (1996) provided the
necessary foundation for HTM cortical learning algorithms herein called HTM-CLA. While sparse coding presented the necessary foundations for cortical-like algorithms, predictive coding takes it a step further. As described in Huang & Rao (2011), predictive coding presents a unifying paradigm for explaining the functional properties of key neural tissue by actively predicting hidden causes of incoming sensory information; this forms the basis of most modern cortical-like predictive monitoring systems such as that proposed in this paper.

2.1 HTM Cortical Learning principles

Hierarchical Temporal Memory (HTM) is originally a memory-prediction framework, and an emerging AI approach that is based on the notion that time plays a crucial role in the operation of neural networks (Hawkins & Blakeslee, 2007). In its current implementation, it is based on the cortical learning algorithms which are a suite of algorithms that favors the use of a more biological plausible paradigm. The core operations that make HTM interestingly unique are centered on the properties of sparse distributed representations (SDRs) being the fundamental data structure in HTM systems. The primary operations in HTM include:

- The activation and prediction of cell states/columns based on the notion of permanence
- Computation of a similarity metric called the “overlap”
- A learning rule for updating permanence's

As described in Cui et al (2016), the prediction and activation states at time step, t, are typically computed as given in (1) and (2) respectively:

\[
\pi_{ij}^t = \begin{cases} 
1 & \text{if } \exists \| D_{ij}^d \circ A^t \|_1 > \theta \\
0 & \text{otherwise} 
\end{cases}
\]  

(1)

\[
\alpha_{ij}^t = \begin{cases} 
1 & \text{if } j \in W^t \text{ and } \sum_i \pi_{ij}^t - 1 = 1 \\
1 & \text{if } j \in W^t \text{ and } \sum_i \pi_{ij}^t - 1 = 0 \\
0 & \text{otherwise} 
\end{cases}
\]  

(2)
The percentage of active columns typically follows a 2% rule for maximum fixed sparsity distribution (Hawkins et al, 2010). The assumption that the 2% rule holds and can be determined apriori remains a fundamental issue and is yet to be resolved by HTM researchers. More so, it is still not very clear if the brain uses this rule for learning sparsely. Nonetheless, the HTM model still works reasonably well in certain sequence learning tasks such as in time series related problems.

2.2 Skip-Sequence iterator for sparse-predictive memory systems

In this section, we introduce the concept of skip-sequence iterator for cost-effective generation of highly sparse memories. Skip-sequences are basically a form of drop-out where a portion of the input observation is sequentially skipped. This concept is illustrated pictorially in Figure 1. The skip-sequence approach creates an affordable, faster and hence less power-hungry mining process during data predictive learning which can help in reducing the dimensionality in data and hence avoid the “curse of dimensionality” in the DLA’s operation. Reducing the dimensionality in data has been proven to be a useful and very important task particularly when

\[
\text{where,} \\
\hat{D}_{i,j} = \text{an M×N binary matrix representing the permanence of a connected synapse belonging to a dendrite segment} \\
d = \text{a cortical segment} \\
i, j = \text{cell and column states} \\
A^t = \text{an M×N binary matrix denoting the activation state of the cortical network} \\
\varphi = \text{the cortical segment activation threshold or permanence} \\
N = \text{number of cortical columns} \\
\mu = \text{number of neurons (cortical cells) per column} \\
\% = \text{percentage of columns with most synaptic inputs} \\
The operation \|\hat{D}_{i,j} \circ A^t\|_{1,\alpha} \text{ is fundamentally referred to as the “overlap”. However, in the DLA, we use the absolute difference instead of the dot-product to compute a deviant overlap – this is accounted for by the Change Detection (CD) principle (Lieder et al, 2013a).} \\
\begin{align*}
\text{In order to improve on the HTM predictions, a Hebbian-like learning rule is applied to the dendrite segment:} \\
\Delta D_{i,j} = p^+ \left( D_{i,j} \circ A^{t-1} \right) - p^- D_{i,j} \\
\text{where, } p^+ \text{ and } p^- \text{ are permanence increments and decrements respectively.}
\end{align*}

The percentage of active columns typically follows a 2% rule for maximum fixed sparsity distribution (Hawkins et al, 2010). The assumption that the 2% rule holds and can be determined apriori remains a fundamental issue and is yet to be resolved by HTM researchers. More so, it is still not very clear if the brain uses this rule for learning sparsely. Nonetheless, the HTM model still works reasonably well in certain sequence learning tasks such as in time series related problems.

2.2 Skip-Sequence iterator for sparse-predictive memory systems

In this section, we introduce the concept of skip-sequence iterator for cost-effective generation of highly sparse memories. Skip-sequences are basically a form of drop-out where a portion of the input observation is sequentially skipped. This concept is illustrated pictorially in Figure 1. The skip-sequence approach creates an affordable, faster and hence less power-hungry mining process during data predictive learning which can help in reducing the dimensionality in data and hence avoid the “curse of dimensionality” in the DLA’s operation. Reducing the dimensionality in data has been proven to be a useful and very important task particularly when
the datasets grow in size and memory needs to be conserved (Cunningham & Byron, 2014). However, this comes with a price of lower predictive classification accuracies and hence, lowers the pattern recognition abilities of the DLA.

Fig 1. Illustration of two activities of Skip-Sequences; note that sequence 2 is skipped in the second activity leading to a sparse memory sequence-of-sequences

2.3 Learning Extent Theory (LET)

The learning extent theory is in line with the birth-death principle and is based on the hypothesis that learning performance increases as learning units' increase. Here, birth refers to the instantiation of a learning machine and the start of its learning operations. The primary units of learning in the DLA are the "symbolic integers". The integers along with a DLA processor form a sparse field of memory patterns. At birth only a few learning units (symbolic integers) is available during learning; but as the learning machine ages, the learning units increases and the machine starts to learn more complex tasks i.e. its learning performance appreciates due to a higher integer capture. However, in line with the death principle, at a certain learning extent (threshold) the machine should start to degrade in performance with reducing learning units until the machine dies off i.e. performance degrades to a point where learning is no longer effective. This process of variational learning can also be interpreted in terms of cortical operation where the integer learning units is replaced with the cortical synapses.

3. Systems Architectural Modelling

The proposed systems architectural concept for predictive classification of pipeline datasets is shown in Figure 2. It is based on the DLA developed in Osegi & Anireh (2016) and consists of the following key units:
- The Pipeline data unit which serves as container for pipeline dataset; this may also serve the purpose of sensor-input unit for real time data acquisition tasks.
- An Integer Encoder for transforming the pipeline input dataset into a mixed-integer sparse distributed representation (SDR).
- A Sequential Pre-prediction unit is for initial mixed-integer pre-processing of the incoming sensory input. This generates a sequence of SDRs in the memory space.
- A Sequential Post-prediction unit that performs further pre-processing of the Sequential Pre-prediction.
- An optional Backward Additive Deviant Computing (BADC) Unit that allows extrapolations to be made on the DLA’s post-predictive memories.
- A classifier for evaluating the predictive classification performance of the entire system.

Fig 2. Architectural concept of proposed system (Source: Authors design)

3.1 Sequential Predictions

Sequential predictions follow a two-stage processing regime. First, the sparse set of input data is first captured into a memory store by computing a set of mismatch operations based on response functions Lieder et al (2013a). The response functions modeling these predictions are given in (4) to (6) and are all mimicked by the DLA software.

\[ g_1 = \begin{cases} 
1, & \text{if } u_i \neq x_i \\
0, & \text{otherwise} 
\end{cases} \]  \hspace{1cm} (4)

\[ g_2 = |u_i - x_i| \]  \hspace{1cm} (5)

\[ g_3 = u_i - x_i \]  \hspace{1cm} (6)
where,

\[ x_t = \text{the non-observable generative internal state}, \]

\[ u_t = \text{a sensory input}. \]

Then, the sparse memory is processed dynamically online using the DLA in a temporal manner. In order to perform extrapolations, the DLA uses the backward additive deviant computing formula to compute an aggregated expression as:

\[
K_{\text{seq}}^t = \sum_{j=1}^{n} \frac{|K_t^j - K_{\text{seq}}^j|}{n} \quad (7)
\]

where,

\[ K_t^n = \text{nth memorized sequence chunk at time, } t \]

\[ K_{\text{seq}}^j = \text{memorized sequence chunks at time steps of } t \]

\[ n = \text{number of previously memorized sequence chunks} \]

Using (7), the DLA’s numeric prediction can then be computed as:

\[
K_p^t = K_{\text{seq}}^j + K_t^n \quad (8)
\]

1. Experimental Results and Discussions

4.1 Experiment 1

Simulation experiments have been performed using the pipeline feature datasets developed in Tan et al (2016) and Lam (2015). The data obtained from Tan et al (2016) is an example of a very small feature dataset with 9 exemplars and two attributes comprising three linguistic threat attributes and 10 feature vectors; the threat attributes include: Manual Digging, Machine Excavation and Vehicle Passing; a numeric target attribute (labelled a 1 or 0) is intentionally embedded in each exemplar to indicate the presence (indicated by a 1) or absence (indicated by a 0) of a threat. The dataset obtained from Lam (2015) is a much larger one with 190 exemplars and 6 attributes one of which represents the target threat attribute; this attribute was transformed
into a numeric equivalent of 1 (indicating presence of a threat) and 0 (indicating absence of a threat).

The pattern recognition prediction results of experiments using the smaller and larger pipeline datasets are shown in Table 1. As can be seen, the DLA and HTM-CLA gave very accurate results (100% accuracy) using the first dataset (pipeline (tan)_data). In the second dataset (pipeline (lam)_data), the HTM-CLA outperformed the DLA after two trials. However, for the first trial, HTM-CLA gave similar result as DLA.

The results of the DLA for the second dataset under different percentage sparsities are shown in Table 2. From the results it is obvious that the DLA still performs reasonably well attaining accuracies of around 70%.

Table 1
Classification accuracies of DLA and HTM-CLA using small and large benchmark datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DLA</th>
<th>HTM-CLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>pipeline(tan)_data</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>pipeline(lam)_data</td>
<td>99.33</td>
<td>99.33</td>
</tr>
<tr>
<td>pipeline(lam)_data(^a)</td>
<td>99.33</td>
<td>100.00</td>
</tr>
</tbody>
</table>

\(^a\)Accuracy obtained after two trials

Table 2
Predictive classification accuracies of DLA with skip-sequence using the large benchmark dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DLA (2%)</th>
<th>DLA (5%)</th>
<th>DLA (10%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pipeline(lam)_data</td>
<td>69.33</td>
<td>73.33</td>
<td>72.67</td>
</tr>
</tbody>
</table>
4.2 Experiment 2

In these set of experiments, we validate the performance of the DLA and HTM-CLA using three set of tests. In the three tests, we define an expectation exemplar in the dataset which have not been seen by both algorithms. In the first two tests our expectations are positive i.e. a threat is indicated with a high state of numeric value equal to 1. The last test has a negative expectation i.e. threat value equal to 0 indicative of a no-threat scenario. The results of these experiments are depicted graphically in Figures 3-5 for the DLA and Figures 6-8 for the HTM-CLA. From the figures shown, the blue blobs represent the state (positions) of the signal. Here, signals at 1 indicate a threat while signals at 0 indicate the absence of a threat.

From the results, it is obvious that both algorithms are capable of multi-predictions, with the HTM-CLA obtaining a more precise visual representation than the DLA. However, both algorithms fared very well as expected.

The DLA's predictions of threat are centred on the first and last prediction sequences (see Figures 3-4). However, for the third test, the threat signal was incorrectly predicted at the last sequence.

![Fig. 3 Threat signals and the corresponding sequence(s) as predicted by the DLA for the 130th exemplar.](image)
Fig. 4 Threat signals and the corresponding sequence(s) as predicted by the DLA for the 178\textsuperscript{th} exemplar.

Fig. 5 Threat signals and the corresponding sequence(s) as predicted by the DLA for the 200\textsuperscript{th} exemplar.
Fig. 6 Threat signals and the corresponding sequence(s) as predicted by the HTM-CLA for the 130th exemplar.

Fig. 7 Threat signals and the corresponding sequence(s) as predicted by the HTM-CLA for the 178th exemplar.
Fig. 8 Threat signals and the corresponding sequence(s) as predicted by the HTM-CLA for the 200th exemplar.

1. Conclusion

The abilities of a novel machine learning (ML) algorithm – the Deviant Learning Algorithm (DLA), in performing predictive classification of onshore pipeline incidence/threat datasets have been demonstrated in this research paper. The model has been shown to be comparable in memory prediction abilities with a variant of another proven online cortical-like ML algorithm – the Hierarchical Temporal Memory based on Cortical Learning Algorithms (HTM-CLA) which is capable of multiple predictions. The DLA also shows promising accuracies of around 70% or higher even with higher sparsity (skip-sequence). This has the advantage of lower cost in sequential memory processing when performing predictive monitoring operations.

Further effort/work on DLA will focus on a more efficient methodology for a real-time embedded predictive pipeline monitoring system with the hope that this will yield an improved set of results. This in turn unlock the full potentials of sparse memory predictive systems for real-time monitoring tasks.
Acknowledgement

The Authors will like to thank the anonymous reviewers and colleagues whose suggestions were very valuable in the development of this paper.

Source codes for the DLA and HTM-CLA simulations and dataset are available at the MATLAB central website:

www.matlabcentral.com

APPENDIX A: PARAMETERS FOR THE HTM-CLA AND THE DLA

Table A.1: KEY HTM PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desired local activity</td>
<td>3</td>
</tr>
<tr>
<td>Minimum overlap</td>
<td>90</td>
</tr>
<tr>
<td>Initial permanence value</td>
<td>0.21</td>
</tr>
<tr>
<td>Number of Monte-Carlo Runs</td>
<td>50</td>
</tr>
<tr>
<td>Maximum Number of sequences considered</td>
<td>150</td>
</tr>
</tbody>
</table>

Table A.2: KEY DLA PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning extent</td>
<td>255</td>
</tr>
<tr>
<td>Time limit (s)</td>
<td>70</td>
</tr>
<tr>
<td>Initial permanence value</td>
<td>0.00</td>
</tr>
<tr>
<td>Store threshold</td>
<td>120</td>
</tr>
<tr>
<td>Maximum Number of sequences considered</td>
<td>150</td>
</tr>
</tbody>
</table>
REFERENCES


McCall, R., & Franklin, S. (2013). Cortical learning algorithms with predictive coding for a systems-level cognitive architecture. In Proceedings of the second annual conference on advances in cognitive systems (pp. 149-166).


