International Journal of Computer Applications (0975 – 8887)
Volume 184 – No. 52, March 2023

A Comparative Study of an AI Pipeline Monitoring System and Particle Swarm Optimization Technique in Predictive Monitoring Operations: Oil and Gas Pipeline Vandalism

Nlerum Promise Anebo
Computer Science & Informatics Department
Federal University Otuoke
Bayelsa State

Igbudu Kingsley Ezebunwo
Department of Computer Science
Rivers State University,
Port Harcourt, Nigeria

ABSTRACT
This work proposed an improved neural network model known as AI Pipeline Monitoring System for Predictive Monitoring of oil and gas installation vandalism threats. The system employed a sparse representative long-short-memory (SLSTM) learning network as part of a refinement to an existing feed-forward neural network. The system also uses a Gaussian membership function with a context-decision gate for detection and monitoring operations. In this paper the proposed system's efficiency is compared to that of the Particle Swarm Optimization Technique; a swarm intelligence algorithm that is emerging as an alternative to more conventional approaches for predictive monitoring operations. To test and evaluate the performance, dynamic simulations were performed using real-time dataset of most likely vandal behavior and the efficiency of the two systems in predictive monitoring operations. The results of simulation study showed impressive results and proves that the AI Pipeline Monitoring System is more preferred to the Particle Swarm System, because of its (AI Pipeline Monitoring System) continual long range context learning capability, which is a likely feature of most observed pipeline threat context-data.

Keywords
AI monitoring system, Long-short-term memory, Context-decision, Pipeline vandalism, Sparse distributed representations

1. INTRODUCTION
Crude oil theft has increased in recent years, as have continuous threats and sabotage to oil and gas installations by self-proclaimed militant groups, as well as offshore piracy in the world's oil producing regions. As a result of this act, there has been a significant loss of revenue and significant disruptions in gas supply to power industries, businesses, and public sector services [1].

This paper compares an AI pipeline monitoring, which is based on long short-term memory with a particle Swarm optimization anti-vandal's intelligent system for efficiency in oil and gas pipeline facility threat-claificaton and predictive monitoring operations. It is a context-based system that utilizes the concept of context management [2]. Contextual information (feature dataset) about potential threats to oil facility in [3] will be used in the study as shown in table 1. The system is expected to predict activities that exhibit abnormal behavior up to a predetermined threshold, then classify as threats to pipeline installations.

Table 1: Input/output feature data for the vandalism predictions.

Time (hrs)		6:00	8:00	10:00	12:00	14:00
Date						
20/08/2022	Pressure (bar)	17.14	16.00	18.14	20.28	19.78
20/08/2022	Vehicle Passing (kg)	20.22	20.78	17.99	19.22	21.22
20/08/2022	Manual Digging (inches)	16.26	18.87	20.39	19.67	18.19
20/08/2022	Machine Excavation	18.47	17.29	15.28	16.29	20.18
21/08/2022	Pressure (bar)	18.35	17.98	18.99	19.99	16.88
21/08/2022	Vehicle Passing	15.00	18.76	15.78	16.76	20.17
21/08/2022	Manual Digging	18.12	19.38	20.98	17.28	19.98
21/08/2022	Machine Excavation	17.23	18.88	14.45	20.55	19.27

2. RELATED LITERATURES
Recurrent neural networks (RNNs) are a class of artificial neural networks (ANNs) with an inherent time varying feature that allows for the continual processing of the weights of a

typical feed-forward network on different instances of artificial neurons through time [4]. A generalized version of an RNN is typically defined as a recursive neural network. With the current interest in mining streaming/or time varying data, RNNs play an important role as a candidate ANN model of the intelligent mind. In this section a brief overview of the development in RNNs is introduced. The idea of using RNNs was historically first introduced in [5], where the same weight structure is used for varying instances of the artificial neurons in an ANN with different time steps.

The biologically inspired computational search and optimization technique Particle Swarm Optimization (PSO) is based on the behavior of an insect colony or swarm, such as ants, termites, bees, and wasps, a flock of birds, or a school of fish [6].

The particles exchange information or good positions with one another and adjust their individual positions and velocities in response. This is done according to the following model [7]:

$$v_i^{k+1} = Wv_i^k + c_1r_1(pbest_i^k - x_i^k) + c_2r_2(gbest^k - x_i^k) \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+`} \quad (2)$$

Where:

- *C1* and *C2* are two positive constants.
- *r1 and r2* are two randomly generated numbers with a range of [0,1].
- *w* is the inertia weight.
- $pbest_i^k$ is the best position of the particle *i* achieved based on its own experience; k i pbest
- $gbest_i^k$ is the best particle position based on overall swarm's experience, k gbest
- *K* is the iteration index.

The field of threat identification and risk assessment in industries is an active one with recent progress made in both understanding and application-oriented approaches.

In [8], a fault diagnosis system for interdependent critical infrastructures was developed and applied to a synthetic benchmark.

Energy network (IEEE 30 bus model): This system used the Hidden Markov Model (HMM) – a state-based model that uses time as its operational learning parameter. Such models are useful as time-observers in fault-critical situations. Using the HMM model, the quantification of cyber-paths of critical infrastructures (CIs) was attempted. The cyber-paths include BGP routing protocols, SCADA servers, corporate networks etc. The paths are built into a state model that varies through all possible states (ergodic HMMs).

This approach used the probabilistic distance metric (PDM) for decision making and the torch framework (http://www.torch.ch//) for simulation experiments. Performance measures based on false positive and negative rates and the detection delay was shown to be encouraging. However, one drawback in using the HMM is its inability to fully account for the state space – as the capability of detecting a possible threat to a CI is a function of time and space. Bowties, a diagrammatic cause-and-effect-barrier model, have been studied as a risk/threat detection and monitoring mechanism for oil and gas and similar environments [9]. Their analysis of existing bow-tie schemes from the qualitative and quantitative bow-tie paradigms revealed some competing intelligent schemes based on fuzzy logic, Bayesian networks, and Boolean logical calculus.

Using an agent-based model and regression-based statistical design of experiments (doE), [10] were able to simulate pirate behavior that accounts for exploitation of marine environments in Somalia. With the Concept of Operations (CONOPS) agent, they tried to determine which factors of the Meteorology and Oceanography conditions (METOC) likely influence pirate behavior and increase the chances of a threat to existing facilities – these factors or parameters were then allocated more intelligent resources from a mobile pirate control system. However, they found out that using the doE approach with CONOPs for the different regression models did not present a clear-cut direction as to which parameters or conditions are more useful. In [11], the potentials of a Bayesian network for effective identification of threats due to piracy to oil industry infrastructure was investigated.

3. METHODOLOGY

An iterative object-oriented software engineering methodology known as Rational unified Process was used in accordance with the Action Research, which involves iterative refinement and redesigning of the practitioners' equirements. A software engineering process known as Rational Unified Process (RUP) offers a structured method for allocating tasks and responsibilities within a development organization. The RUP strives to produce high-quality software that meets the needs of its end users while adhering to a set schedule and budget [12]. To ensure that the process is continually updated and improved to reflect recent experiences as well as evolving and tried-and-true best practices, the RUP development team works in collaboration with customers, partners, Rational's product groups, and Rational's consultant organization. RUP activity emphasizes developing and maintaining models that are semantically rich representations of the software system being developed in addition to creating and maintaining models [13] RUP is a guide for using the Unified Modelling Language effectively (UML). Our ability to clearly communicate requirements, architectures, and designs is made possible by the industry-standard language known as the UML [14] The standards body Object Management Group now maintains the UML.

4. PREDICTIONS WITH AI PIPELINE MONITORING SYSTEM

Figure 1 depicts the architecture of the AI pipeline Monitoring System, which is a sequence learning recurrent neural network based on Long-Short Term Memory.

The system works as described in figure 1; the contextual information (feature dataset) is broken down into a numerical context prediction activity see Table 1; using the feature dataset module. The context information base on the help of sensors is fed to an LSTM block which learns a sequential representation of the context in the previous time step and then predicts the most likely sequences at the next time step. The predictions are then sent to a threat alert which flags predicted threat levels with high (abnormal) values. Figure 2 contains concatenated local contextual parameters which are incorporated into Pipeline Monitoring System as trafficability values using wireless sensor. The trafficability are values between zero and one, where zero indicates no threat and one indicates threat. The pipeline local contextual data includes:

- ✓ Pressure
- ✓ Vehicle Passing

- ✓ Manual Digging
- ✓ Machine Excavation

The sensed signals (context parameters) are fed into the context learning module (LSTM subsystem) through a multiplexer and are transformed and seen by the LSTM as words. The control subsystem (Prediction module) combines the individual trafficability values corresponding to each piece of contextual information into a value that would be used to indicate situations that pose danger and the ones that do not. The LSTM subsystem keeps track of the sensed signals in the memory module and then passes these signals to the control subsystem (Prediction module).

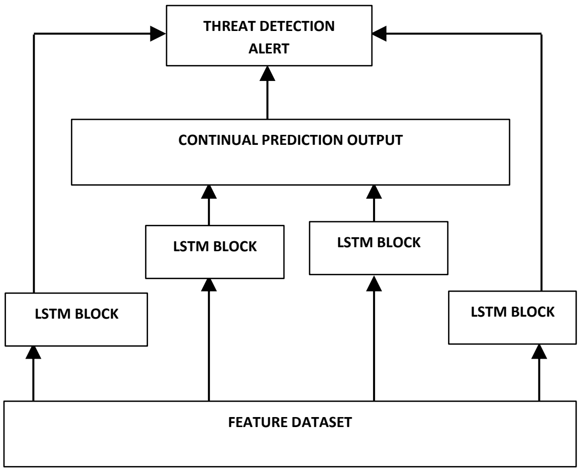

Figure 1: Proposed surveillance system architecture

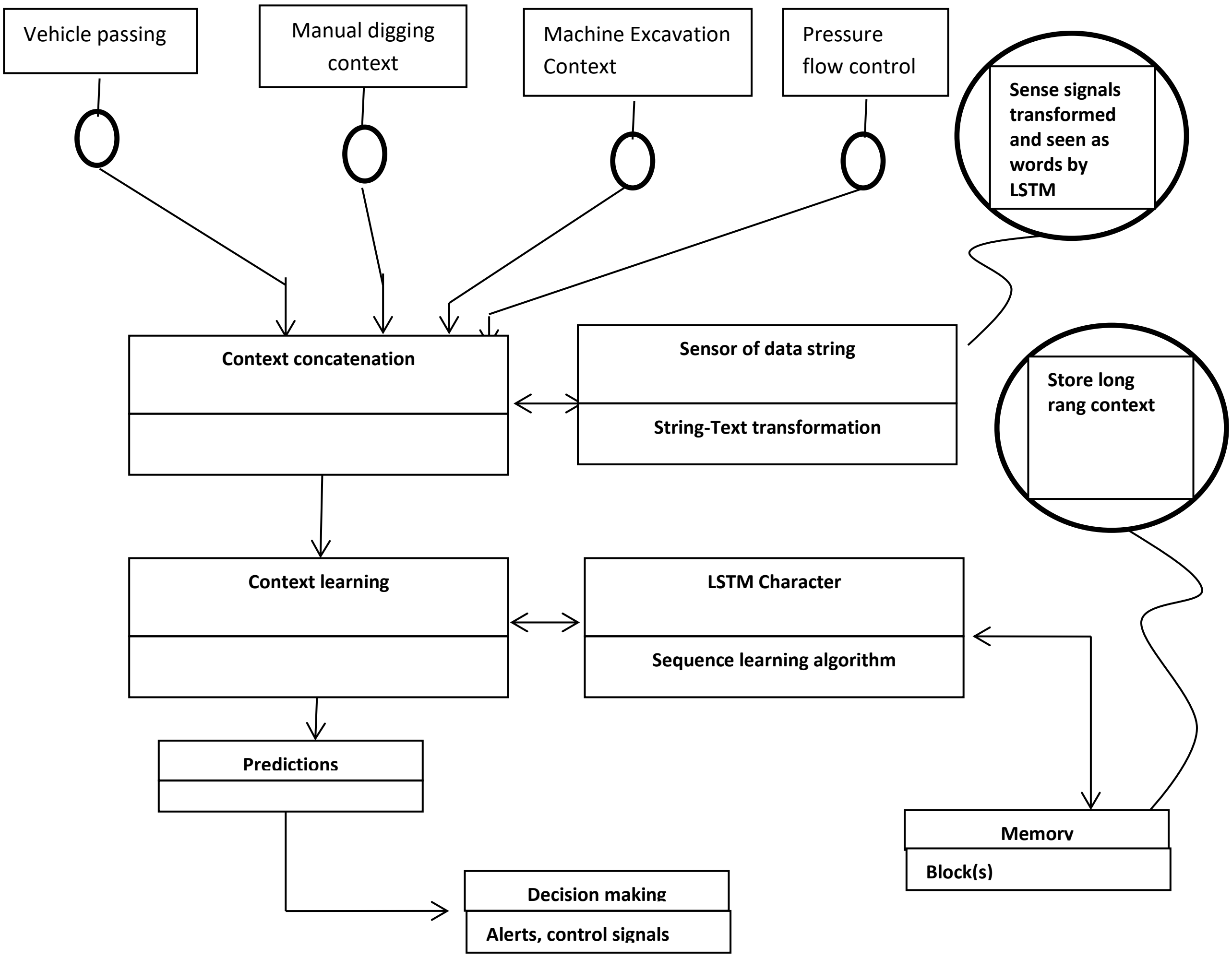

Figure 2: Concept-level situation aware design

The AI Pipeline Monitoring System, which is based on Long Short-Term memory, uses the Gaussian membership function in Simulink.

Gaussian built-in membership function Syntax:

$$Y = gaussmf(X, [sig\ C]) \tag{3}$$

The symmetric Gaussian function depends on two parameters ∂ and c as given by

$$f(X, \partial, C) = e^{\frac{-(x-c)^2}{2\partial^2}} \tag{4}$$

The parameters for gaussmf represent the parameters and c listed in order in the vector [sig c].

Where c is the mean and ∂ the standard deviation.

The deployment interface includes the following parts:

- Random Source: this part is made up of the following: -
    1) Predicted Parameters sensor blocks: this converts the real time parameters like pressure, vehicle passing, manual digging and machine excavation, etc. into electrical signal
    2) Display Sensor block: displays the numerical state of the sensed data
    3) Multiplexer block: that concatenates the entire sensed signal and synchronizes them through a transmission line to the LSTM subsystem.
- LSTM Subsystem: The LSTM subsystem passes these signals through different signal line (transmission line) to the control subsystem.
- Control Subsystem: the control subsystem predicts the sensor output based on the signal it received from the LSTM subsystem and displays a 1 to any predicted activity that poses threat and 0 to the activity that poses no threat.

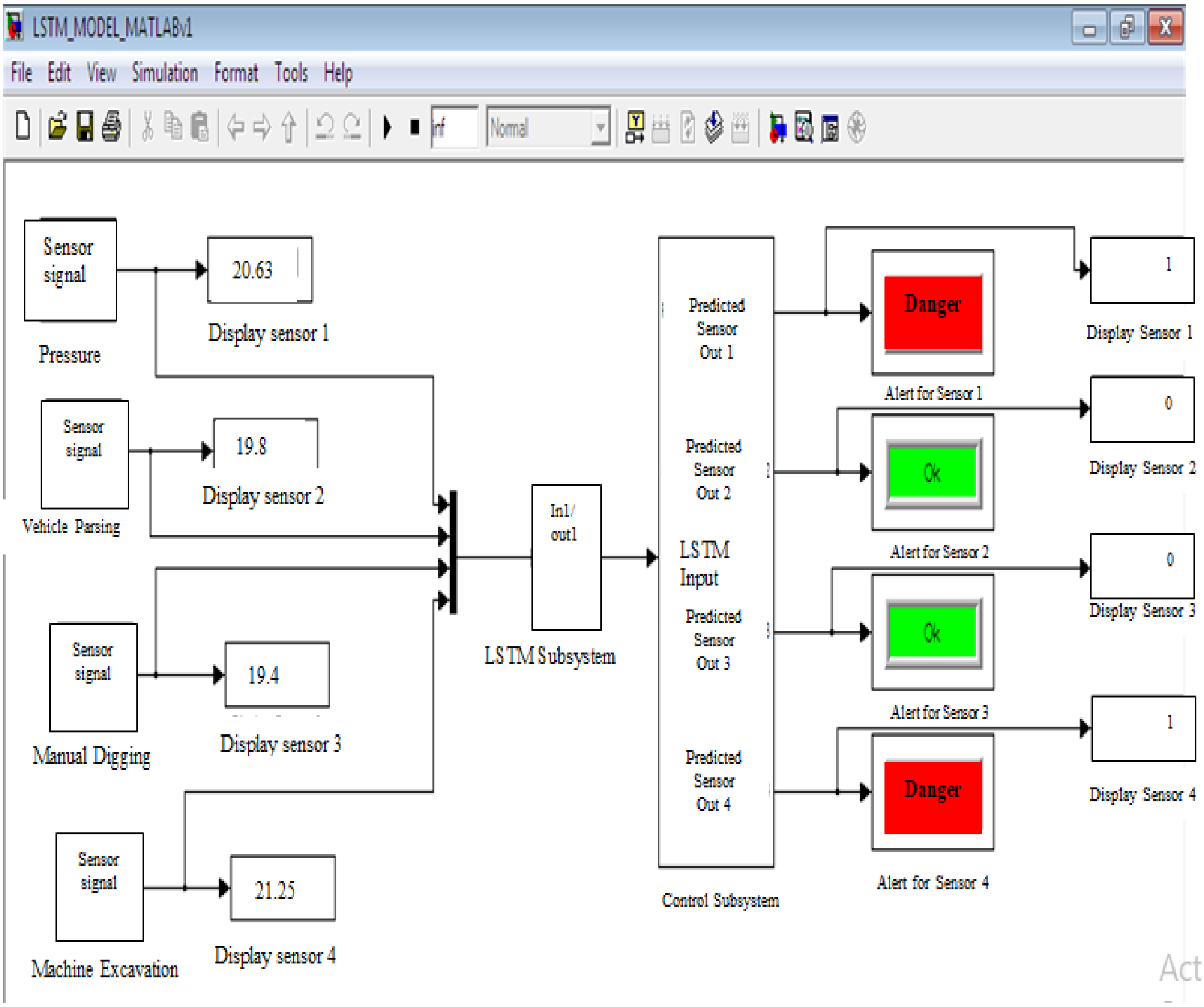

Figure 3: AI pipeline monitoring system user interface

4.1. AI pipeline monitoring systems simulation result

The AI system's efficiency was evaluated using a local pipeline contextual feature dataset. Tables 2 and 3 present the test results. After several runs of the AI Pipeline Monitoring system, the results show a simulation report. The simulation results show that when a specific activity is consistent with the targeted value of the trafficability data, it does not raise a red flag and instead sends a green signal indicating that the pipeline is safe; otherwise, it raises a red flag and sends a threat signal.

Table 2: Simulation results of the AI pipeline monitoring system

	Max Trial = 3			Hidden Unit Size = 20
Time(hrs)	Pressure (bar)	Vehicle Passing (kg)	Manual Digging (inches)	Machine Excavation (inches)
6:00	19.66	17.39	20.71	20.37
0:00	19.43	21.36	19.81	19.55
10:00	19.43	20.05	19.99	20.79
22:00	18.71	19.99	19.23	19.85
2:00	19.95	20.27	19.35	18.67
0:00	17.71	18.55	19.82	20.23
18:00	19.74	19.41	18.12	19.35
16:00	18.89	19.85	21.20	21.01
22:00	20.08	19.76	17.78	21.77
2:00	20.57	19.45	20.19	20.65
10:00	17.38	18.38	20.23	19.30
4:00	20.36	20.94	20.20	20.30
6:00	19.75	19.92	21.12	20.63
8:00	20.95	21.98	19.06	20.76
4:00	22.11	19.01	20.71	20.70
10:00	21.83	20.98	20.92	19.71
0:00	21.73	20.38	20.22	21.08
16:00	20.32	18.49	21.59	19.89
6:00	19.46	20.02	21.11	18.87
2:00	18.73	19.42	20.44	18.87
0:00	20.13	20.68	19.97	20.26

Table 3: Threat events prediction table based on simulation result

Contextual parameters	**Sensory Signal**	**Label**
Pressure	19.66	0 OK
Vehicle Passing	20.05	1 DANGER
Manual Digging	17.88	0 OK
Machine Excavation	16.78	0 OK
Pressure	22.71	1 DANGER
Vehicle Passing	19.78	0 OK
Manual Digging	21.65	1 DANGER
Machine Excavation	20.8	1 DANGER

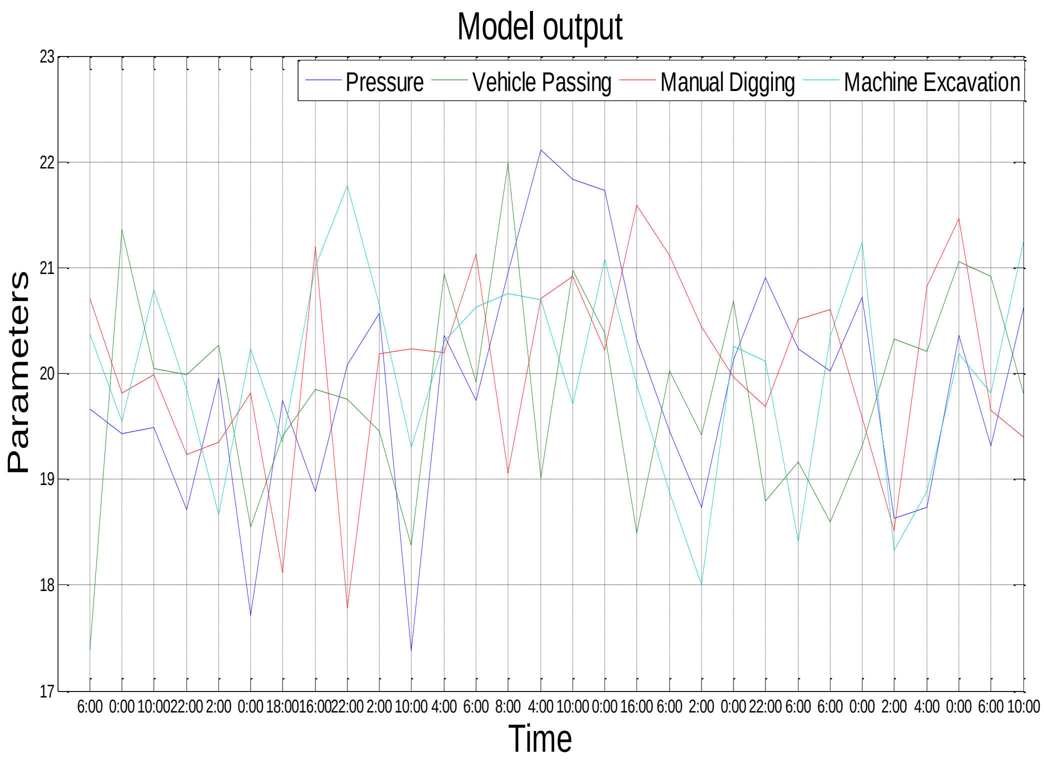

Figure 4: Graph of the predicted parameters against time

5. PREDICTION WITH PARTICLE SWARM OPTIMIZATION TECHNIQUE

An evolutionarily based artificial intelligence called Particle Swarm Optimization Technique (PSO) is used to categorize feature data gleaned from a real-world pipeline sensor detection system. Figure 3 depicts the architecture of the PSO Predictive Monitoring System. The PSO system consists of feature dataset blocks with pre-processed threat detection parameters obtained from real-time phase-shifted optical time domain response systems (ϕ - OTDR). The input/output specifications for the PSO decision system are listed in table 1, and they include pressure, vehicle passing, manual digging, and machine digging activities. These same data are also used in the AI Pipeline Monitoring System. These data are fed into the PSO-classifier module, where the PSO algorithm optimizes the crucial parameter(s) to create a model of the threat classifier. Any activity that continues over time and approaches a restricted value or a predetermined threshold based on the activity being performed at a specific time should be detected.

5.1 Particle swarm optimization simulation results

Table 4 lists the simulation results following several PSO system runs. According to the simulation results, a particular activity would not be predicted to be hazardous to an oil and gas pipeline facility if it is consistent with the threshold value of the trafficability data and instead would give a signal of green indicating that the pipeline is safe.

Table 4: Particle swarm optimization system simulation results

	Max Iteration = 3			Population Size = 20
TIME (hrs)	Pressure(bar)	Vehicle Passing(kg)	Manual Digging (inch)	Machine Excavation (inch)
6:00	17.99	16.45	18.22	17.89
0:00	15.77	15.66	15.43	19.56
10:00	15.12	20.21	14.87	19.17
22:00	19.25	18.16	14.32	17.05
2:00	19.83	15.67	16.43	14.65
0:00	18.26	15.54	16.23	16.15

18:00	15.78	16.43	14.99	15.87
16:00	15.54	15.00	18.14	16.18
22:00	15.76	17.09	16.67	16.87
2:00	18.87	16.89	15.56	14.45
10:00	15.45	18.78	14.56	16.89
4:00	20.24	15.60	16.79	17.11
6:00	15.45	15.15	15.07	15.00
8:00	14.54	14.67	16.17	18.56
4:00	20.22	15.87	15.12	14.65
10:00	16.06	16.73	15.53	19.76
0:00	17.87	15.45	15.71	19.32
16:00	18.56	16.89	16.27	20.57
6:00	16.00	15.87	15.76	15.46
2:00	16.45	14.78	15.25	16.43
0:00	16.45	17.76	17.65	16.88
22:00	15.00	15.17	14.81	16.72
6:00	15.76	14.75	16.87	17.25
10:00	16.00	16.72	17.44	19.99
0:00	17.77	15.43	15.43	19.87
2:00	18.76	15.43	16.65	16.34
4:00	14.54	15.76	15.87	14.34
0:00	20.45	16.43	17.40	21.54
6:00	19.60	15.43	16.21	17.53
10:00	14.56	19.17	17.43	21.27

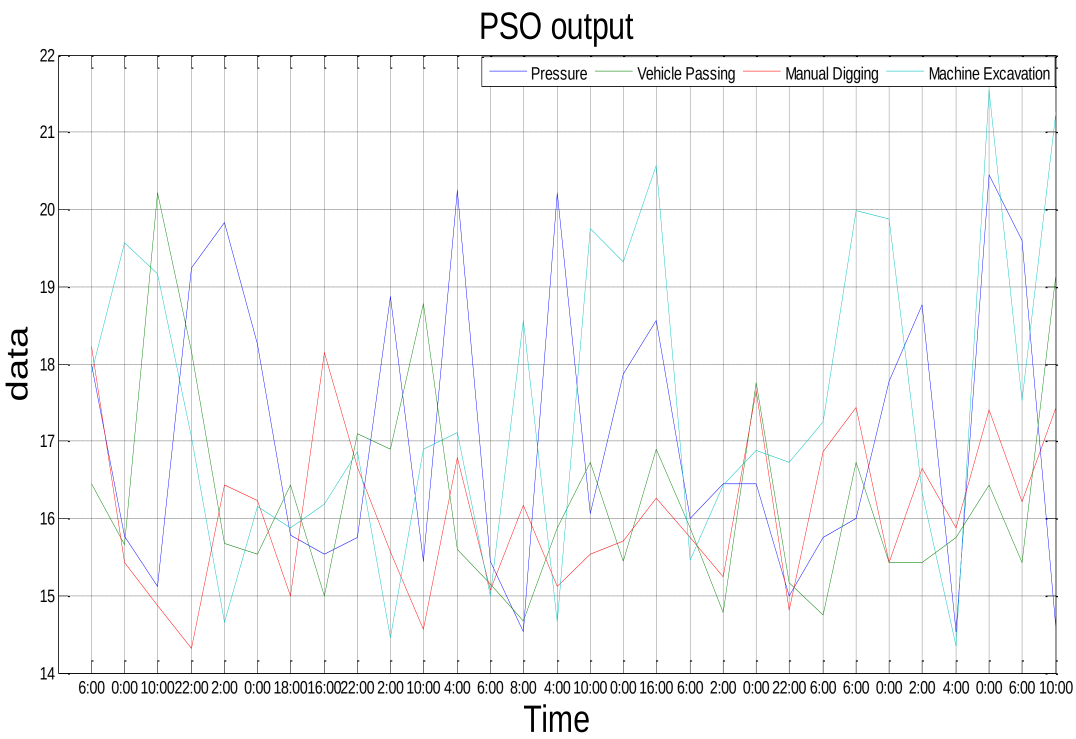

Figure 5: Graph of predicted parameter against time using pso system

6. COMPARING SIMULATION RESULTS OF AI PIPELINE MONITORING SYSTEM AND PARTICLE SWARM OPTIMIZATION TECHNIQUE

Table 5: Comparison between simulation results of the AI pipeline system and PSO system

	PSO	LSTM	PSO	LSTM	PSO	LSTM	PSO	LSTM
Time (hrs)	Pressure (bar)		Vehicle Passing (kg)		Manual Digging (inch)		Machine Digging (inch)	
06:00	17.99	19.66	16.45	17.39	18.22	20.71	17.89	20.37
00:00	15.77	19.43	15.66	21.36	15.43	19.81	19.56	19.55
10:00	15.12	19.49	20.21	20.05	14.87	19.99	19.17	20.79
22:00	19.25	18.71	18.16	19.99	14.32	19.23	17.05	19.85
02:00	19.83	19.95	15.67	20.27	16.43	19.35	14.65	18.67
00:00	18.26	17.71	15.54	18.55	16.23	19.82	16.15	20.23
18:00	15.78	19.74	16.43	19.41	14.99	18.12	15.87	19.35
16:00	15.54	18.89	15.00	19.85	18.14	21.2	16.18	21.01
22:00	15.76	20.08	17.09	19.76	16.67	17.78	16.87	21.77
02:00	18.87	20.57	16.89	19.45	15.56	20.19	14.45	20.65
10:00	15.45	17.38	18.78	18.38	14.56	20.23	16.89	19.37

04:00	20.24	20.36	15.60	20.94	16.79	20.2	17.11	20.3
06:00	15.45	19.75	15.15	19.92	15.07	21.12	15.00	20.63
08:00	14.54	20.95	14.67	21.98	16.17	19.06	18.56	20.76
04:00	20.22	22.11	15.87	19.01	15.12	20.71	14.65	20.70
10:00	16.06	21.83	16.73	20.98	15.53	20.92	19.76	19.71
00:00	17.87	21.73	15.45	20.38	15.71	20.22	19.32	21.08
16:00	18.56	20.32	16.89	18.49	16.27	21.59	20.57	19.89
06:00	16.00	19.46	15.87	20.02	15.76	21.11	15.46	18.87
02:00	16.45	18.73	14.78	19.42	15.25	20.44	16.43	18.78
00:00	16.45	20.13	17.76	20.68	17.65	19.97	16.88	20.26
22:00	15.00	20.9	15.17	18.79	14.81	19.69	16.72	20.12
06:00	15.76	20.23	14.75	19.16	16.87	20.51	17.25	18.41
10:00	16.00	20.02	16.72	18.59	17.44	20.6	19.99	20.35
00:00	17.77	20.72	15.43	19.32	15.43	19.58	19.87	21.24
02:00	18.76	18.63	15.43	20.33	16.65	18.51	16.34	18.33
04:00	14.54	18.74	15.76	20.21	15.87	20.82	14.34	18.89

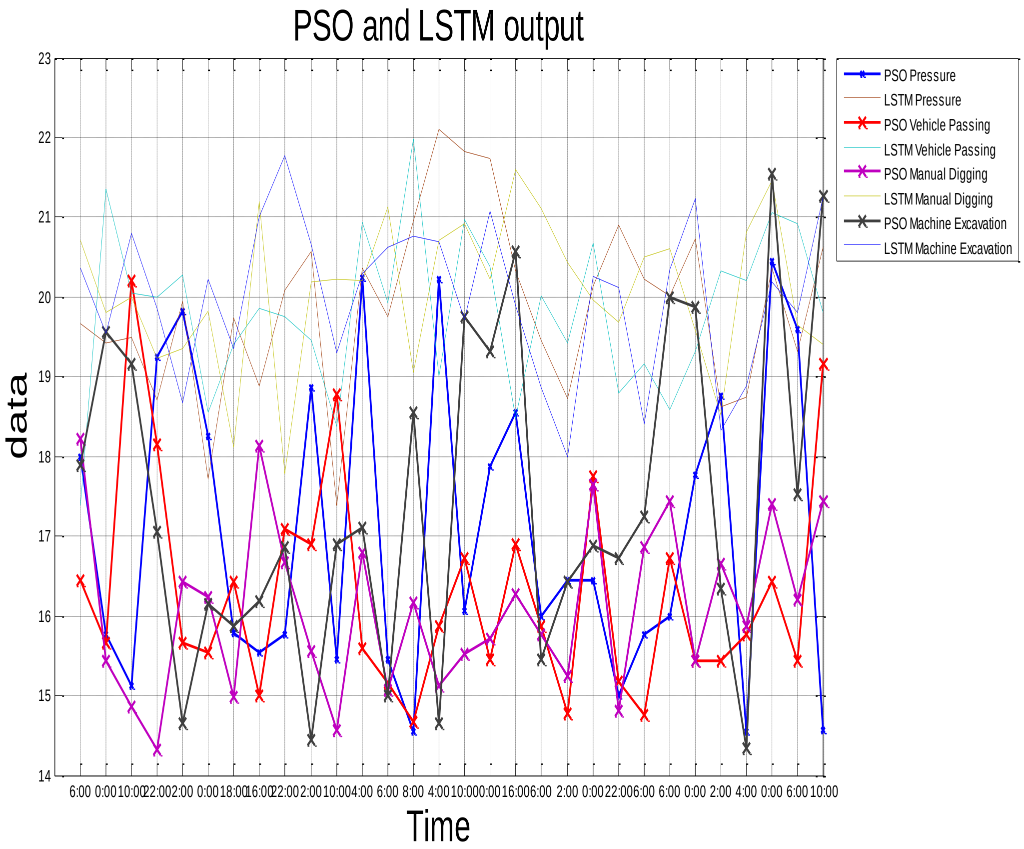

Figure 6: Graphs of the compared simulation results of the LSTM AI pipeline system and PSO system

The contextual feature dataset was tested and run in both systems to evaluate their performance and efficiency in predictions. The simulation results of the two systems are tabulated in table 5. Figure 6 show the graph of simulation results of the particle swarm optimization technique as recorded in table 5. The PSO system uses Max iterations = 3 and Population size = 20, and in case of the LSTM AI Pipeline Monitoring System uses Max trial = 3 and Hidden unit size =20.

A plot of the compared simulation results of the PSO and LSTM of figure 6 is discussed here. The bolded lines with asterisks on it represent the PSO result, whereas the normal tiny lines represent the LSTM results. the blue line indicates the rising and dropping of pressure wave at different time interval for PSO and brown for LSTM respectively, at 6:00 am the pressure wave as predicted by PSO is 17.99 bar, LSTM is 19.66, at 0:00 am the wave Predicted by PSO is 15.77 bar, LSTM is 19.43 bar, at 10:00 am the pressure dropped as predicted by PSO is 15.12 bar and LSTM is 17.38 bar, all of which indicate no threat because they fall between the range of the set point. At 4:00 pm the pressure rises as predicted by PSO is 20.24 bar, LSTM is 22.11 bar and at another interval of 00:00 am pressure rise by PSO prediction is 21.71 bar and that of the LSTM is 21.71 bar which are above rising threshold and indicates threat to the pipeline. The red line with asterisks and the normal tiny sky-blue line as shown in figure 4.4 indicate the different weight exerted on the pipeline as vehicles passing along the oil pipeline area at different time interval as predicted by PSO and the LSTM respectively. At 6:00am the weight signal as predicted by PSO to the pipeline system is 16.45 kg and LSTM is 17.39 kg which indicates no threat, at 0:00 am weight by PSO signal is 15.66 kg indicting no threat and LSTM is 21.36 kg indicating threat and at the time interval of 8:00 am the vehicle weight by PSO is 14.67 kg indicting no threat to the oil pipeline and LSTM is 21.98 kg, indicating a threat to pipeline. This continues for the other parameters as shown in figure 5. Comparison result shows that the PSO holds great advantages in terms of its use of an optimization routine, high accuracy, and ability to model complex non-linear decisions. However, results show that the PSO system is deficient in its training time, that is, it has a slow training time and cannot learn continual long-range context, which is a likely feature of most observed pipeline threat context data; a major issue that the proposed Oil and Gas AI Pipeline Monitoring System has come to address.

7. CONCLUSION

Based on comparison results, it was observed that the Oil and Gas AI Pipeline Monitoring System, which is based on the LSTM, holds great promises as a future neural network model for predictive monitoring operation if properly planned. The ideas of advanced machine learning recurrent neural networks such as the one proposed here can lead to better neural models for diverse tasks. Thus, it is desirable that researchers shift from using existing simple neural network architectures to more sophisticated ones.

Further work will be on integration of the AI Pipeline Monitoring model into real time hardware, as the system has not been integrated into real time hardware, hence, it may not be obvious if it will perform as expected. Finally, the output is not symbolic i.e., cannot be interpreted as a mathematical expression yet.

8. AUTHOR'S PROFILE

Conceptualization, N.P.A and I.K.E investigation, N.P.A., methodology, N.P.A and I.K.E. writing-original draft, and N.P.A and I.K.E..; writing-review and editing.

8.1 Funding

This research received no external funding.

8.2 Conflicts of Interest

The authors declare no conflict of interest.

8.3 Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
SLSTM	Sparse Representative Long-Short-Memory
RNNs	Recurrent Neural Networks
ANNs	Artificial Neural Networks
PSO	Particle Swarm Optimization
HMM	Hidden Markov Model
CI	Critical Infrastructures
BGP	Border Gateway Protocol
SCADA	Supervisory Control and Data Acquisition
PDM	Probabilistic Distance Metric
CONOPS	Concept of Operations
METOC	Meteorology and Oceanography conditions
RUP	Rational Unified Process
UML	Unified Modeling Language
OMG	Object Management Group
LSTM	Long-Short Term Memory
OTDR	Optical Time Domain Response

9. REFERENCES

[1] Mathias, D. and Igbudu, K.E., (2018). Application of Particle Swarm Optimization Technique in Monitoring of Oil Theft and Oil facility in the Niger Delta. Nigerian Journal of Oil and Gas Technology 3(2), 110-119

[2] Nlerum, P. A. and Edward E. O. (2020): Resource Management in a Pervasive Computing Environment. Asian Journal of Research in Computer Science, AJRCOS.54523 5(1) 29-39.

[3] Nwiabu, N. D., & Igbudu, K. E. Oil and Gas Pipeline Monitoring using Artificial Neural Network. International Journal of Computer Applications, 975, 8887.

[4] Bengio, Y., Goodfellow, I. J., and Courville, A. (2015). Deep learning. An MIT Press book in preparation. Retrieve from http://www.iro.umontreal.ca/∼bengioy/dlbook.

[5] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation (No. ICS-8506). California UNIV SAN DIEGO LA JOLLA INST FOR COGNITIVE SCIENCE.

[6] Sunita, S., Arindam, R. and Bipul, S. P. (2013). Application of Particle Swarm Optimization in Data Clustering: A Survey: International Journal of Computer Applications 65(25), 0975 – 8887

[7] Kusakana, K., and Vermaak, H.J. (2016). A survey of Particle Swarm Optimization Applicationsfor Sizing Hybrid Renewable System: Journal of Computational and Theoretical Nanoscience 8(11), 172-183

[8] Ntalampiras, S., Soupionis, Y., and Giannopoulos, G. (2015). A fault diagnosis system for interdependent critical infrastructures based on HMMs. Reliability Engineering and System Safety, 138(3), 73-81.

[9] Mathias, D. and Igbudu, K.E., (2018). Application of Particle Swarm Optimization Technique in Monitoring of Oil Theft and Oil facility in the Niger Delta. Nigerian Journal of Oil and Gas Technology 3(2), 110-119

[10] Esher, L., Hall, S., Regnier, E., Sánchez, P. J., Hansen, J. A., and Singham, D. (2010, December). Simulating pirate behavior to exploit environmental information. In Simulation Conference (WSC), Proceedings of the 2010 Winter (pp. 1330-1335).

[11] Nuka, D. N. and Igbudu, K. E., (2017) Oil and Gas Pipeline Monitoring using Artificial Neural Network: International Journal of Computer Applications 180(1), 0975 – 8887

[12] Jacobson, I., Grady B., and Jim, R. (1999) Unified Software Development Process- Introduce the main concepts of iterative and incremental development 4(3), 12-26

[13] Grady, B., and Addison-Wesley (1995) Object Solutions: Managing the Object-Oriented Project 2(4), 15-23

[14] Kruchten, P., Capilla, R. and Duenas, J.C. (2009). The Decision view's role in software architecture practice, IEEE software 26(2), 36-42

[15] Torch framework (http://www.torch.ch//)