Condition Monitoring System for Internal Blowout Prevention (IBOP) in Top Drive Assembly System using Discrete Event Systems and Deep Learning Approaches

Nadia Saad Noori[1], Filippo Maria Bianchi[1], and Tor Inge Waag[1]

[1] *NORCE Norwegian Research Center, Bergen, Norway*
{*nano, fibi, towa*}*@norceresearch.no*

ABSTRACT

Offshore oil drilling is a complex process that requires careful coordination of hardware and control systems. Fault monitoring systems play an important role in such systems for safe and profitable operations. Thus, predictive maintenance and monitoring operating conditions of drilling systems are critical to the overall production cycle. In this paper, we are addressing the topic of condition monitoring of a critical part in the process of oil drilling, the Internal Blowout Preventer (IBOP) system in the top drive assembly in offshore oil drilling. In our work, we aim to design an intelligent system for monitoring the health of IBOP system using discrete event systems (DES) based control method in combination with multivariate time series classification deep learning method. The proposed system comprises two stages: 1) produce IBOP system logical behaviour analysis using Hierarchical Colored Petri Nets (HCPN) approach; 2) develop an activity detection or a classifier module using reservoir computing framework for classification of multivariate time series for activity monitoring and fault detection for the top drive assembly.

The combination of these methods would enable automation of monitoring and early detection of incidents during drilling operations. We present the preliminary results of a model in Petri Nets used to simulate a monitoring system for IBOP valve in top drive assembly and activity classification of activities relevant to IBOP condition monitoring. The effects of failure rate and repair time of each component on system performance are to be researched at a later stage.

1. INTRODUCTION

The internal blowout preventer (IBOP) system located in the topdrive assembly [1] functions a support mechanism to the main blowout preventer system (BOP) in Oil drilling opera-

tions. However, during the exploration phase, it provides the first line of defence to prevent blowouts and ensure safe working conditions for drilling activities in offshore deep-water operations. Failures of blowout preventer valves could lead to catastrophic accidents causing damage to human lives and the environment. The explosion of the deep-sea oil drilling rig Deepwater Horizon and the consequent oil spill off the coast of Louisiana on April 20, 2010, is a recent example of the impact of such incidents. The blowout prevention system of the Deepwater Horizon rig was believed to have been faulty before the blowout, or it might have been damaged because of the accident causing the failure to isolate the well before and after the explosions (Harlow, Brantley, & Harlow, 2011) (Skogdalen, Utne, & Vinnem, 2011). Thus, an intelligent automaton of monitoring and predictive maintenance of operating conditions are critical to the safety and profitability of oil and gad sector (Sayda & Taylor, 2006) (Cai et al., 2012).

IBOPs are designed to operate in a very harsh operational condition associated with high pressure and drilling fluids containing heavy metal particles and cuttings. Figure 1, includes a diagram of the topdrive assembly and an example of a damaged IBOP valve due to erosion caused by the pressure and the drilling mud fluid. Despite the critical role of the BOP and IBOP systems, few studies addressed the reliability of blowout preventer (BOP) and in particular, IBOP systems have not been widely studied according to a literature review conducted by the researchers. Some studies focused on producing statistical process for determining reliability and failure rate necessary to accomplish the maintenance goal (Shanks, Dykes, Quilici, Pruitt, et al., 2003; Holand & Rausand, 1987) Other studies addressed rig downtime and relation to BOP system failure rates (Fowler, Roche, et al., 1993).

In this work, we to design an automated system for monitoring the health of IBOP system using a subset of data collected from the drilling stack. For the HCPN system, we used signals like active pressure, open and closing of BOP valves, drilling speed. For the machine learning (ML) activity de-

[1]Oilfield Glossary: topdrive, Schlumberger Limited - https://www.glossary.oilfield.slb.com/en/Terms/t/topdrive.aspx
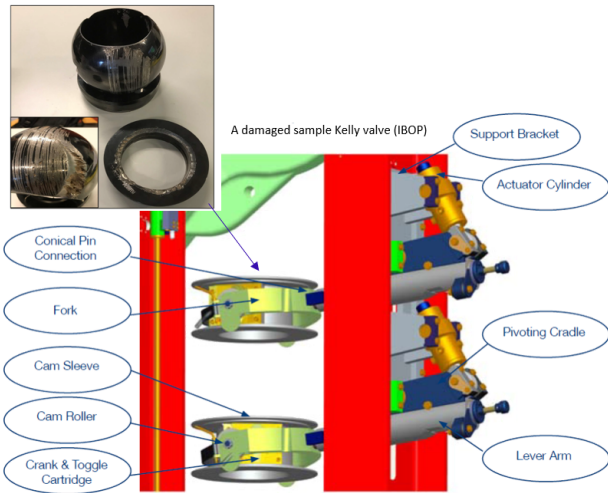
Figure 1. Example of damaged IBOP valve

tection system, we used signals to identify drilling activities relevant to the IBOP operations (e.g. drilling and connection and mud circulation).

2. METHODOLOGY

Offshore oil drilling is a complex process that requires careful coordination of hardware and control systems. Relying on one method to efficiently automate the monitoring and prediction of any process/component involved the oil & gas production is impossible. Thus, in this research work, we propose a combination of two methods from both control systems and data science to address the complexity of the system and data at hand, where Petri-nets used to capture the system behaviour and ML used to automate activity detection using the vast amount of data acquired from the sensors.

2.1. IBOP monitoring process using Hierarchical Colored Petri Nets (HCPN)

A popular approach was adopted for modelling interacting processes, that is the Petri-nets and its two derivatives, coloured Petri-nets and hierarchical coloured Petri-nets (HCPN) (Peterson, 1977; Bruno & Marchetto, 1986). Petri nets modelling is used to model processes and procedures in many areas including disaster and emergency management. The Petri-nets method was used as a tool to evaluate different procedures and process flow in order to improve the disaster and emergency management systems (Holloway, Krogh, & Giua, 1997; Bruno & Marchetto, 1986). Even though Petri nets are useful graphical-tools to represent elements of complex systems, the classical method lacked the capacity to represent multiple resources and multiple layers of simultaneous coordination processes. Such requirement was crucial to our model because response operations involved a variety of re-

sources and multiple layers of simultaneous coordination actions. Therefore, an extension of the classical Petri nets was adopted to help model the complex systems of coordination in response operations (Karmakar & Dasgupta, 2011). The coloured and hierarchical Petri-nets were used to describe the complex resources and simulate the different tasks carried out inside a complex process. With such capacity to represent diversified resources and hierarchical operations, we were able to capture and simulate operations with different levels of complexity.

The use of classical CPN for large and complex systems can produce a model that is less readable and complicated to trace. Luckily, hierarchical CPNs offered features, which enabled a modular and multi-layer representation of large complex systems. Hierarchical CPN modelling became a great asset to have to model complex systems such as oil drilling operations (Sayda & Taylor, 2006).

Therefore, choosing HCPN approach provided us with a tool for simulation and graphical visualization of dynamic discrete process and provides means to identify bottlenecks, deadlocks and optimization parameters.

In the work we use coloured and hierarchical Petri nets that are offered by the CPN-Tools software package (http://cpntools.org/) (Fowler et al., 1993; Cai et al., 2012; Holand & Rausand, 1987; Kristensen, Christensen, & Jensen, 1998). The CPN-Tools software is a tool for editing, simulating and analyzing un-timed and timed hierarchical coloured Petri nets. CPN-Tool is the result of a research project, the CPN2000 project at the University of Aarhus, sponsored by the Danish National Centre for IT Research (CIT), George Mason University, Hewlett-Packard, Nokia, and Microsoft (Jensen, 1991; Ratzer et al., 2003).

The CPN-Tools offered a valuable tool-set to visually and mathematically model the complex processes of the drilling operation and hence the condition monitoring processes needed (Moreira, da Silva, Almeida, & Ramalho, 2015). This unique combination of graphical and mathematical representations and a programming language allowed the creation of sophisticated models without having to lose its detailed aspects.

HCPN were used to represent the different levels of operations related to the IBOP operations and the use cases related to Tripping-In, Tripping-Out, Drilling and Connection. For the safety check, there are main two routines relevant to our work, "Pressure test" and "Maintenance" routines. IBOP valve maintenance routine is recommended to be carried out at 10- 12 weeks of normal operations. However, this duration differs from one vendor to other and rig operators. The hi-level of the IBOP operations is shown in Figure 2. The system will alert operators for 1) necessary pressure tests and maintenance routines (as recommended by the supplier or based on Mud Pressure measurement);2) operating the valve while pressurized during drilling or connection activities, and issue

Sensor 1: IBOP OPEN/CLOSE, Sensor 2: Mud pressure, Sensor 3: Drilling RPM, Sensor 4: DDM torque,
Sensor 5: Hook POS w/out Compensation; Sensor 6: Hook POS w/ Compensation, Sensor 7: Hook load
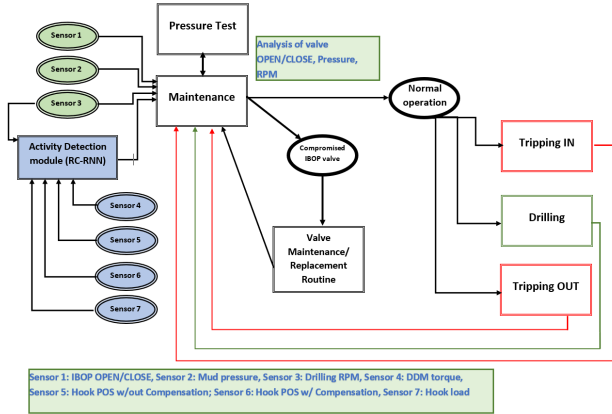
Figure 2. Hi-level of the logical analysis of the IBOP monitoring system

necessary warnings; 3) operating the valve beyond the recommended operational thresholds.

2.2. Dataset for HCPN system

The IBOP valves are not designed to operate while pressurized (except in an emergency), thus, sensor data for mud pressure, IBOP OPEN and CLOSE signals and drilling operation were used as inputs for the system to record the opening and closing of the valve. The combination of mud pressure, IBOP status (OPEN/CLOSED) and activity detection inputs to the maintenance control unit in figure 2, will be used to detect pressurized operation (based on Mud Pressure measurements and operational thresholds provided by the valve manufacturer) and keep a record of the use of the valve. The activity detection module acts as a soft sensor providing the type of ongoing operations, e.g. drilling, tripping-in, tripping-out or drilling and connections.

2.3. Activity detection with time series classification

Detection of different drilling activities is framed as a multivariate time series (MTS) classification problem, which consists of assigning each MTS to one of a fixed number of classes. The problem has been tackled by approaches spanning from the definition of tailored distance measures over MTS to the identification of patterns in the form of dictionaries or shapelets (Baydogan & Runger, 2016), (Mikalsen, Bianchi, Soguero-Ruiz, & Jenssen, 2018), (Bianchi, Livi, Mikalsen, Kampffmeyer, & Jenssen, 2019).

In this paper, we focus on classifiers based on recurrent neural networks (RNNs) (Bianchi, Maiorino, Kampffmeyer, Rizzi, & Jenssen, 2017), which the first process sequentially the MTS with a dynamic model, and then exploit the sequence of the model states generated over time to perform classification (Bianchi, Scardapane, Løkse, & Jenssen, 2018a). Among the possible RNN architectures, we used models

from the family of Reservoir computing (RC), which is an established paradigm for modelling nonlinear temporal sequences (Lukoševičius & Jaeger, 2009). In ML tasks, echo state networks (ESNs) are the most common RC models, wherein the input sequence is projected to a high-dimensional space through the use of a (fixed) nonlinear recurrent reservoir (Lukoševičius & Jaeger, 2009). The reservoir states can be used to extract informative dynamical features useful to separate the classes (see Fig. 3).

Learning is performed by applying traditional classification models designed for vectorial data to the representations generated from high-dimensional reservoir space. The lack of flexibility in the recurrent part is balanced by a range of advantages, including faster training compared to other recurrent neural networks (RNNs).In tasks requiring a limited amount of temporal memory, ESNs achieve state-of-the-art results in many real-world scenarios constrained by time budgets, low-power hardware and limited data (Scardapane & Wang, 2017).

The implementation of our classifier is based on the unified Reservoir Computing framework for multivariate time series classification (Bianchi, Scardapane, Løkse, & Jenssen, 2018b). The unified framework consists of four modules: i) a reservoir module, ii) a dimensionality reduction module, iii) a representation module, and iv) a readout module. The reservoir module is responsible to extract a rich pool of dynamical features and can be implemented as a unidirectional or bidirectional reservoir.

The advantage of bidirectional architectures is that they can extract from the input sequence features that account for dependencies very far in time. For our data, we found out that a standard reservoir is sufficient to achieve good performance and the bidirectional architecture is not needed.

The dimensionality reduction module projects the sequence of reservoir activation on a lower-dimensional subspace, using unsupervised criteria. Since the reservoir is characterized by a large number of neurons, dimensionality reduction yields a more compact representation, which can provide a regularization to the model that enhances its generalization capability and simplifies its training (Belkin & Niyogi, 2003). In the context of RC, commonly used algorithms for reducing the dimensionality of the reservoir are PCA and kernel PCA, which project data on the first D eigenvectors of a co-variance matrix (Løkse, Bianchi, & Jenssen, 2017). In this work, we apply an extension of PCA for data represented as tensors.

The representation module generates vectorial representations from the sequence of reservoir states. Popular choices are the last reservoir state, the average of all reservoir states or more advanced representations, such as the parameters of a linear model trained to predict the next state of the reservoir (Bianchi et al., 2018b). In this work, we chose this last representation since it yields the best results.
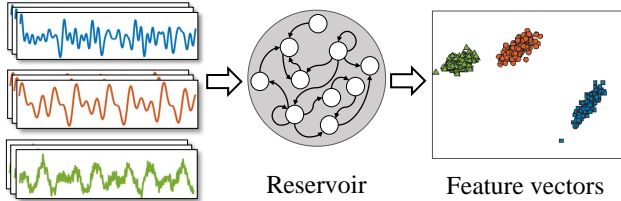
Figure 3. The reservoir generates high dimensional states containing a rich pool of dynamical features of the underlying system observed from the time series. The reservoir states can be combined to extract vectorial representations of the MTS, which can be separated using traditional classification models for vectorial data.

The readout module classifies the representations and is either implemented as a linear readout, or a support vector machine (SVM) classifier, or a multi-layer perception (MLP). The MLP is a universal function approximator that can learn complex representations of the input by stacking multiple layers of neurons configured with a non-linear activation, e.g., rectified linear unit (ReLU). Deep MLPs are known for their capability of disentangling factors of variations from high-dimensional spaces (Goodfellow, Lee, Le, Saxe, & Ng, 2009), and therefore can be more powerful and expressive in their instantaneous mappings from the representation to the output space than linear readouts (Bianchi, Livi, Alippi, & Jenssen, 2017). When paired with a recurrent architecture like the Reservoir module, the number of layers in the MLP determines the "feed-forward" depth in the RNN (Pascanu, Gulcehre, Cho, & Bengio, 2014). In this work, we tried all three readouts: linear, SVM, and MLP.

2.4. Dataset for activity detection

Our dataset consists of 5 time series containing the values of 5 sensors monitored for 15 days. The time series are

- *DDM torque*;
- *Drilling RPM*;
- *Hook POS WOUT COMP*;
- *Hook POS W COMP*;
- *Hook load*.

The time series have different length because the values for each sensor are sampled at a different time resolution. Each entry in a time series indicates when the value measured by the sensor changes. Therefore, given two consecutive measures $x(t)$ and $x(t + \delta)$, all the values assumed in the time window δ are the same. This makes possible to interpolate the time series to have the same length. In particular, the shorter time series are stretched to match the length of the longest time series, by filling the new extra value by carrying forward the last value observed. There are three different activities associated with specific windows in the time series. The activities we want to identify are:

- Drilling and connection (ID: 1)
- Tripping out of the hole (ID: 5)
- Condition and/or Circulate mud (ID: 7)

Examples of drilling and connection activity are depicted as vertical blue bands in Fig. 4, which shows time series for 6 hours.

To build our dataset, we extracted chunks from the total time series of 15 days of measurements and assigned a class to each segment. Each segment might have a different length. For simplicity, we resized each segment by means of cubic interpolation to assume the same length of 500 time steps. Examples of the activity *Drilling and connection (ID: 1)* are depicted in Fig. 5. Examples of the activity *Tripping out of hole (ID: 5)* are depicted in Fig. 6. Examples of the activity *Condition and/or Circulate mud (ID: 7)* are depicted in Fig. 7.

After extracting the activity segments, we ended up by having a total of 342 MTS of length 500, associated with one of the three IDs (1,5,7) indicating the corresponding class. We also added 100 additional MTS of length 500 randomly sampled from the remaining parts of the time series and we assigned them with the ID=0, denoting the *other activity* class. The reason for including data associated to other activities is because we want to assess the capability of our classification model to discriminate between the 3 classes of interest and the rest.

3. RESULTS

To translate the processes described in Figure 2 to a hierarchical Petri-Nets system, we divided the system into the following sub-systems: Overall System Monitor, IBOP Monitoring System, IBOP Analysis, and IBOP Maintenance. The HCPN model for the overall system is shown in Figure 8.

The sub-system for the The "IBOP Monitoring System" model receives several inputs, i.e. sensor 1, sensor 2 and sensor 3 (see Figure 9), from actual sensors installed on the rig providing measurements for drilling RPMs, pressure in the drilling pipe, open/close commands, etc.

The "Overall System Monitor" sub-process, (shown in Figure 8), registers the inputs from sensors and updates conditional operations based on ongoing activities, manufacturing specifications of pressure levels' limits, history of the valve operation (e.g. number of valves' opening and closing, type of drilling sequence – drilling, trip in, trip out).

The outcome of the "Overall System Monitor" module would be either "Normal Operation" or "Abnormal Operation" leading to either initiating the "IBOP Analysis" sub-process or "IBOP Maintenance" sub-process (see Figure 10 and Figure 11).

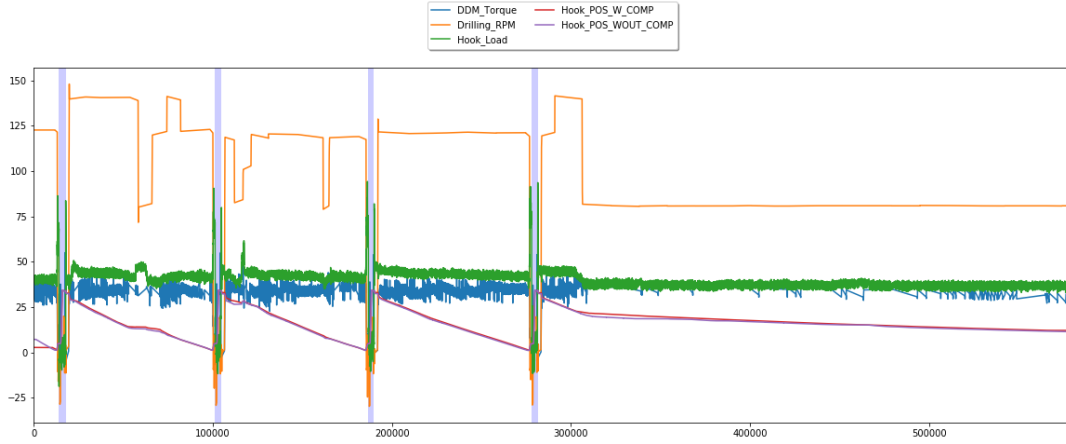The "IBOP Analysis" sub-process keeps track of valve status

Figure 4. Time series for 6 hours of activity. The blue bands indicate *drilling and connection* activities.
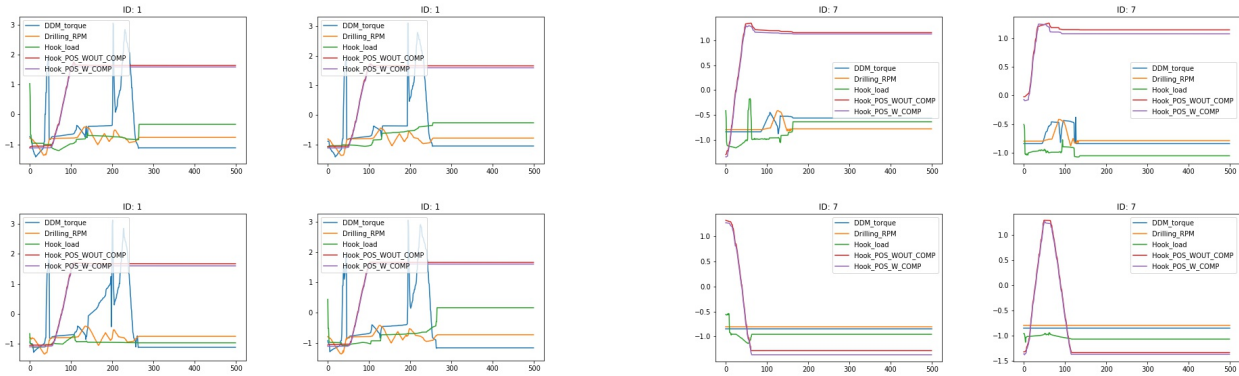

Figure 5. Examples of time series segments of activity *Drilling and connection*.

Figure 7. Examples of time series segments of activity *Condition and/or Circulate mud*.
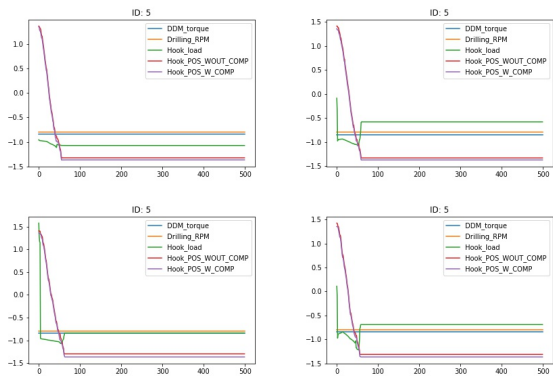

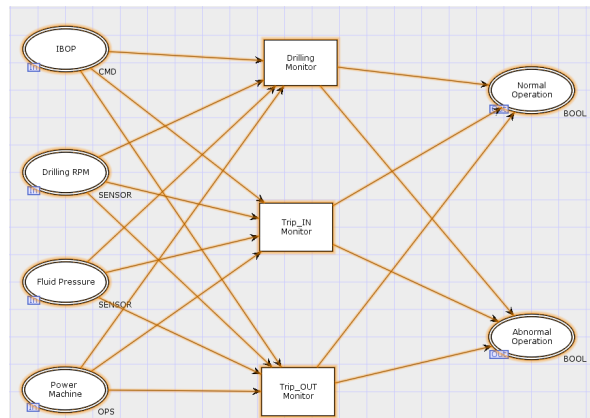Figure 6. Examples of time series segments of activity *Tripping out of hole*.

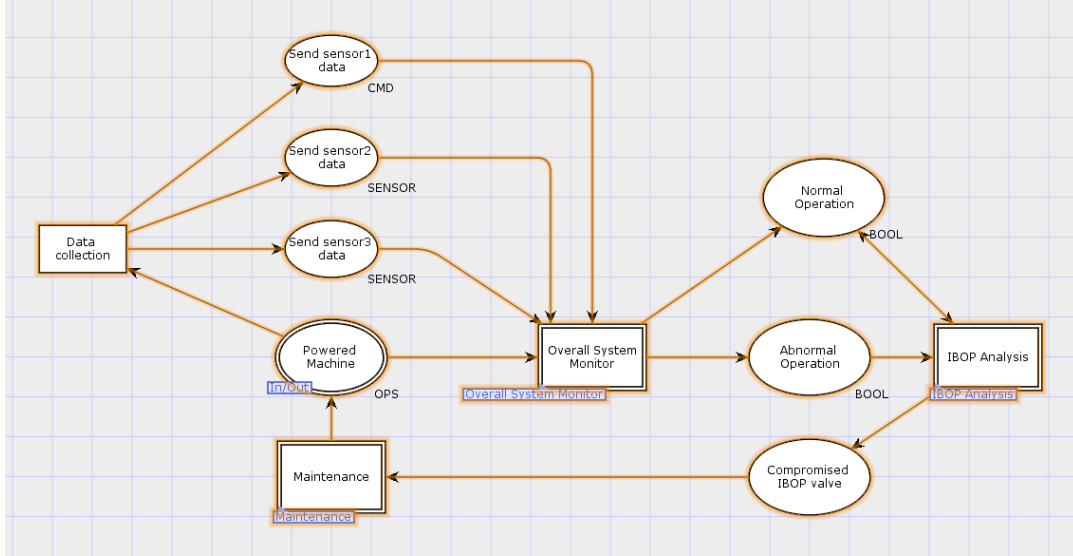Figure 8. HCPN model of "Overall System Monitor" process

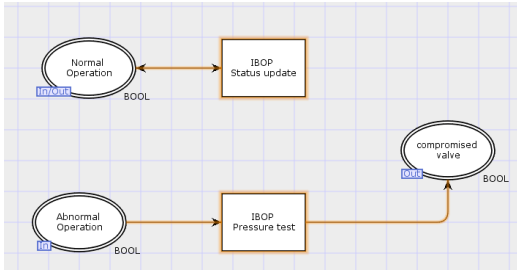Figure 9. HCPN model of "IBOP Monitoring System" process

Figure 10. HCPN model for "IBOP Analysis" sub-process for operational condition check and update
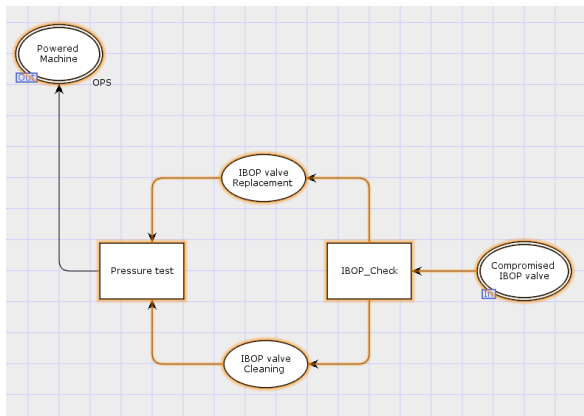

Figure 11. HCPN model for "IBOP Maintenance" sub-process

and initiates the maintenance routine in case of detecting abnormal operational conditions (e.g. valve closed and pressure fluctuation).

For the activity detection module, we report the results obtained by our RC classifier when using different types of readout, namely: a linear classifier, an SVM with radial basis function kernel and a Multilayer Perceptron. The randomized reservoir is configured with the following hyperparameters: number of internal units $R = 800$; spectral radius $\rho = 0.99$; non-zero connections percentage $\beta = 0.25$; input scaling $\omega = 0.15$; noise level $\xi = 0.001$. When the classification is performed with a ridge regression readout, we set the regularization value $\lambda = 1.0$. The MLP is configured with 2 layers of 10 processing units with ReLU activation, the dropout probability is $p_{\mathrm{drop}} = 0.1$; the ℓ_2 regularization parameter is $\lambda = 0.0001$; gradient descent is performed with the Adam algorithm (Kingma & Ba, 2014) and we train the models for 1000 epochs. Finally, the SVM hyperparameters, are the smoothness of the decision hyperplane, $c = 1.0$, and bandwidth of the rbf kernel, $\gamma = 0.1$. We used 267 MTS (60% of the whole dataset) for training the model. We used 178 MTS (40% of the whole dataset) for testing the performance of the trained model on new, unseen data.

Tab. 1 reports results in terms of accuracy and F1-score, which is defined as

$$F1 = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

where *precision* is defined as $\frac{TP}{TP+FP}$ and *recall* is $\frac{TP}{TP+FN}$ (TP = True Positives, FP = False Positives, FN = False Negatives). We used F1 score because the number of samples in each one of the 4 classes (activity ID 0, activity ID 1, activ-
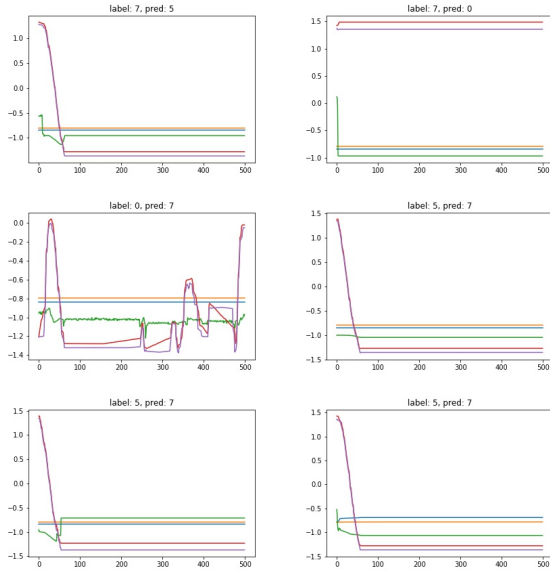
6

Figure 12. MTS wrongly classified by our model when using the MLP readout. On the top of each figure "label" indicates the value of the activity that is given, while "pred" is the value predicted by the RC classifier.

ity ID 5, and activity ID 7) is slightly unbalanced. The results

Table 1. Activity classification results. We report the mean accuracy and F1 score obtained on the test set.

Classifier	Accuracy	F1 score
Linear	89.3	88.2
SVM	90.4	89.8
MLP	96,1	96.1

show that the linear classifier achieves the lowest results. This is expected since a linear classifier can only separate data with hyperplanes and it cannot learn more complicated functions to divide the classes. The SVM classifier achieves slightly better results, thanks to the non-linearity of the RBF kernel. However, SVM depends on critical hyper-parameters, such the bandwidth, that are difficult to optimize. Too small values can overfit the data, while large values produce too rough decision boundaries. As expected, the MLP obtained higher performance, thanks to the capability of learning a complex function that can disentangle factors of variation in the data.

When using the MLP, only 7 out of 178 MTS in the test set are classified incorrectly. We show the 7 MTS wrongly classified in Fig. 12. We can see that most of the classification errors stem from a disagreement about class 5 and 7. By looking at the training data, the two classes contain MTS with very similar patterns and it is expected that a few examples can be confused with each other. Overall, we obtain very high accuracy and F1 score.

4. CONCLUSION

Operation of the offshore machinery is a balance between economy and safety, and for critical components like the IBOP safety always comes first. At present, the recommended practice is replacement before each critical operation, at a high cost. The overarching goal of the research work is to gather reliable information about the present condition as well as previous operations in order to give reliable decision support on the replacement of the component. Also, gaining exact information of the usage pattern based on activity detection and load corrected usage factors will provide valuable feedback to the operators and managers in the field and the personnel responsible for the maintenance and operation procedure, as the cause of unnecessary wear and tear can now be traced back to the sub-optimal operation of the device.

In this paper, we presented a preliminary integration of two methods for automating IBOP condition monitoring process namely: 1) Discrete Event Systems (DES), Petri-nets method to simulate the IBOP monitoring system, 2) Artificial Intelligence (AI)- reservoir computing RNN for activity detection soft sensing module. The Petri-Nets model receives a combination sensor data directly from the rig system and the activity detection module to keep track of drilling activities and generate the necessary actions based on ongoing activities in relation to the IBOP condition. The proposed system helps to capture different processes related to monitoring the IBOP condition and increase operational safety and system reliability. The accuracy of the system predictions and output is greatly dependent on the quality and accuracy of the signals coming from the rig, especially in real-time operations.

The system was designed using historical data rather realtime data stream. Future steps would be to validate the system using a live data stream to validate the performance for both the Petri-Net and the automated activity detection models.

5. ACKNOWLEDGMENT

REFERENCES

Baydogan, M. G., & Runger, G. (2016). Time series representation and similarity based on local autopatterns. *Data Mining and Knowledge Discovery*, *30*(2), 476–509.

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, *15*(6), 1373–1396.

Bianchi, F. M., Livi, L., Alippi, C., & Jenssen, R. (2017). Multiplex visibility graphs to investigate recurrent neural network dynamics. *Scientific reports*, *7*, 44037.

Bianchi, F. M., Livi, L., Mikalsen, K. Ø., Kampffmeyer, M., & Jenssen, R. (2019). Learning representations of multivariate time series with missing data. *Pattern Recognition*, *96*, 106973. doi: https://doi.org/10.1016/j.patcog.2019.106973

Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Jenssen, R. (2017). *Recurrent neural networks for short-term load forecasting: An overview and comparative analysis*. Springer.

Bianchi, F. M., Scardapane, S., Løkse, S., & Jenssen, R. (2018a). Bidirectional deep-readout echo state networks. In *European symposium on artificial neural networks*.

Bianchi, F. M., Scardapane, S., Løkse, S., & Jenssen, R. (2018b). Reservoir computing approaches for representation and classification of multivariate time series. *arXiv preprint arXiv:1803.07870*.

Bruno, G., & Marchetto, G. (1986). Process-translatable petri nets for the rapid prototyping of process control systems. *IEEE Transactions on software Engineering*(2), 346–357.

Cai, B., Liu, Y., Liu, Z., Tian, X., Zhang, Y., & Liu, J. (2012). Performance evaluation of subsea blowout preventer systems with common-cause failures. *Journal of Petroleum Science and Engineering*, *90*, 18–25.

Fowler, J., Roche, J., et al. (1993). System safety analysis of well control equipment. In *Offshore technology conference*.

Goodfellow, I., Lee, H., Le, Q. V., Saxe, A., & Ng, A. Y. (2009). Measuring invariances in deep networks. In *Advances in neural information processing systems* (pp. 646–654).

Harlow, W. F., Brantley, B. C., & Harlow, R. M. (2011). Bp initial image repair strategies after the deepwater horizon spill. *Public Relations Review*, *37*(1), 80–83.

Holand, P., & Rausand, M. (1987). Reliability of subsea bop systems. *Reliability engineering*, *19*(4), 263–275.

Holloway, L. E., Krogh, B. H., & Giua, A. (1997). A survey of petri net methods for controlled discrete event systems. *Discrete Event Dynamic Systems*, *7*(2), 151–190.

Jensen, K. (1991). Coloured petri nets: A high level language for system design and analysis. In *High-level petri nets* (pp. 44–119). Springer.

Karmakar, S., & Dasgupta, R. (2011). A petri net representation of a web-service-based emergency management system in railway station. *World Academy of Science, Engineering and Technology*, *59*, 2284–2290.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kristensen, L. M., Christensen, S., & Jensen, K. (1998). The practitioner's guide to coloured petri nets. *International Journal on software Tools for technology Transfer*, *2*(2), 98–132.

Løkse, S., Bianchi, F. M., & Jenssen, R. (2017, Jun 01). Training echo state networks with regularization through dimensionality reduction. *Cognitive Computation*, *9*(3), 364–378.

Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, *3*(3), 127–149.

Mikalsen, K. Ø., Bianchi, F. M., Soguero-Ruiz, C., & Jenssen, R. (2018). Time series cluster kernel for learning similarities between multivariate time series with missing data. *Pattern Recognition*, *76*, 569 - 581.

Moreira, F. D. L., da Silva, F. P. A., Almeida, J. S., & Ramalho, G. L. B. (2015). A study of a machine condition monitoring system using petri nets. *Journal of Mechanical Engineering and Automation*, *5*(3), 89–93.

Pascanu, R., Gulcehre, C., Cho, K., & Bengio, Y. (2014). How to construct deep recurrent neural networks. In *Proc. 2017 international conference on learning representations (iclr)*.

Peterson, J. L. (1977). Petri nets. *ACM Computing Surveys (CSUR)*, *9*(3), 223–252.

Ratzer, A. V., Wells, L., Lassen, H. M., Laursen, M., Qvortrup, J. F., Stissing, M. S., . . . Jensen, K. (2003). Cpn tools for editing, simulating, and analysing coloured petri nets. In *International conference on application and theory of petri nets* (pp. 450–462).

Sayda, A. F., & Taylor, J. H. (2006). An implementation plan for integrated control and asset management of petroleum production facilities. In *2006 ieee conference on computer aided control system design, 2006 ieee international conference on control applications, 2006 ieee international symposium on intelligent control* (pp. 1212–1219).

Scardapane, S., & Wang, D. (2017). Randomness in neural networks: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(2).

Shanks, E., Dykes, A., Quilici, M., Pruitt, J., et al. (2003). Deepwater bop control systems-a look at reliability issues. In *Offshore technology conference*.

Skogdalen, J. E., Utne, I. B., & Vinnem, J. E. (2011). Developing safety indicators for preventing offshore oil and gas deepwater drilling blowouts. *Safety science*, *49*(8-9), 1187–1199.

BIOGRAPHIES

Nadia S. Noori a senior scientist at NORCE Norwegian Research Center, is an expert in ICT for safety and security deployment. She is a Marie Skłodowska-Curie alumnus and holds a PhD in Electronic and Information Engineering with specialization in Disaster Management from LaSalle URL in Spain. In addition to engineering education, she holds a master's degree in Technology Innovation Management from Carleton University, Ottawa-Canada. Her main focus in the field of automation in industrial applications is cyber-physical

systems security and safety.

Filippo Bianchi is a researcher at NORCE. His research focuses on developing methodologies at the intersection of signal processing, dynamical systems, graph theory, and deep learning. He mainly worked with Recurrent Neural Networks, Reservoir Computing, and Graph Neural Networks to solve inference tasks on time series and graphs. He also applies machine learning to solve computer vision tasks in other fields such as earth observation.

Tor Inge Waag is a Chief Scientist at NORCE Norwegian

Research Center. His background is in Technical Physics from NTNU in Trondheim, where he took his M.Sc and Ph.D in signal processing for laser light scattering. His work has been concentrated on the entire chain from sensor data via signal processing to decision support, mainly for the offshore industry in Norway. He is a member of Society of Petroleum Engineers (SPE) and of the Norwegian Academy of Technological Sciences (NTVA). His recent activity within the SFI Offshore Mechatronics at the University of Agder has been focused on Condition Based Maintenance.