Evolving Systems (2018) 9:119–133
DOI 10.1007/s12530-017-9206-8

ORIGINAL PAPER

Evolving ANN-based sensors for a context-aware cyber physical system of an offshore gas turbine

Farzan Majdani[1] · **Andrei Petrovski**[1] · **Daniel Doolan**[2]

Received: 27 December 2016 / Accepted: 17 October 2017 / Published online: 27 October 2017

Abstract An adaptive multi-tiered framework, that can be utilised for designing a context-aware cyber physical system to carry out smart data acquisition and processing, while minimising the amount of necessary human intervention is proposed and applied. The proposed framework is applied within the domain of offshore asset integrity assurance. The suggested approach segregates processing of the input stream into three distinct phases of Processing, Prediction and Anomaly detection. The Processing phase minimises the data volume and processing cost by analysing only inputs from easily obtainable sources using context identification techniques for finding anomalies in the acquired data. During the Prediction phase, future values of each of the gas turbine's sensors are estimated using a linear regression model. The final step of the process— Anomaly Detection—classifies the significant discrepancies between the observed and predicted values to identify potential anomalies in the operation of the cyber physical system under monitoring and control. The evolving component of the framework is based on an Artificial Neural Network with error backpropagation. Adaptability is achieved through the combined use of machine learning and computational intelligence techniques. The proposed framework has the generality to be applied across a wide range of problem domains requiring processing, analysis and interpretation of data obtained from heterogeneous resources.

Keywords Smart condition monitoring · Context awareness · Cyber physical system · Asset integrity · Artificial neural network

1 Introduction

There exists a growing demand for smart condition monitoring in engineering applications often achieved through evolution of the sensors used. This is especially true when some constraints are present that cannot be satisfied by human intervention with regard to decision making speed in life threatening situations (e.g. automatic collision systems, exploring hazardous environments, processing large volumes of data). Because computer-assisted instrumentation is capable of processing large amounts of heterogeneous data much faster and is not subject to the same level of fatigue as humans, the use of machine-based condition monitoring in many practical situations is preferable.

Cyber physical systems (CPSes) integrate information processing, computation, sensing and networking, which allows physical entities to operate various processes in dynamic environments (Lee 2008). Many of these intelligent CPSes carry out smart data acquisition and processing that minimise the amount of necessary human intervention. In particular, a considerable research interest lies in the area of managing huge volumes of alerts that may or may not correspond to incidents taken place within CPSes (Pierazzi et al. 2016).

The integration of multiple data sources into a unified system leads to data heterogeneity, often resulting

✉ Farzan Majdani
f.majdani-shabestari@rgu.ac.uk

Andrei Petrovski
a.petrovski@rgu.ac.uk

Daniel Doolan
Daniel.Doolan@bcu.ac.uk

[1] School of Computing Science and Digital Media, Robert Gordon University, Aberdeen, UK

[2] School of Computing and Digital Technology, Birmingham City University, Birmingham, UK

into difficulty, or even infeasibility, of human processing, especially in real-time environments. For example, in real-time automated process control, information about a possible failure is more useful before the failure takes place so that prevention and damage control can be carried out in order to either completely avoid the failure, or at least alleviate its consequences.

Computational Intelligence (CI) techniques have been successfully applied to problems involving the automation of anomaly detection in the process of condition monitoring (Khan et al. 2014). These techniques however require training data to provide reliable and reasonably accurate specification of the context in which a CPS operates. The context enables the system to highlight potential anomalies in the data so that intelligent and autonomous control of the underlying process can be carried out.

Anomalies are defined as incidences or occurrences, under a given circumstances or a set of assumptions, that are different from the expected outcome (for instance when generator rotor speed of the gas turbine goes below 3000 rpm). By their nature, these incidences are rare and often not known in advance. This makes it difficult for the Computational Intelligence techniques to form an appropriate training dataset. Moreover, dynamic problem environments can further aggravate the lack of training data by occurrence of intermittent anomalies.

Computational Intelligence techniques that are used to tackle dynamic problems should therefore be able to adapt to situational/contextual changes. A multi-tiered framework for CPSes with heterogeneous input sources is proposed in the paper that can deal with unseen anomalies in a real-time dynamic problem environment. The goal is to develop a framework that is as generic, adaptive and autonomous as possible. In order to achieve this goal both machine learning and computational intelligence techniques are applied within the framework, together with the online learning capability that allows for adaptive problem solving.

The application of the CI techniques to provide evolving functionality of the intelligent sensors deployed within cyber physical systems is the first novel contribution of the presented work. The second contribution is the implementation of the generic framework to make the CPSes context-aware by processing a large amount of heterogeneous data. Finally, the application of these novel approaches to developing evolving sensory systems for optimising the operation of an offshore gas turbine constitutes another original contribution of the paper that demonstrates practical benefits of the suggested methodology.

2 Cyber physical systems

Rapid advances in miniaturisation, speed, power and mobility have led to the pervasive use of networking and information technologies across all economic sectors. These technologies are increasingly combined with elements of the physical worlds (e.g. machines, devices) to create smart or intelligent systems that offer increased effectiveness, productivity, safety and speed (Lee 2008). Cyber physical systems (CPS) are a new type of system that integrates computation with physical processes. They are similar to embedded systems but focus more on controlling the physical entities rather than processes embedded computers monitor and control, usually with feedback loops, where physical processes affect computations and vice versa. Components of cyber physical system (e.g. controllers, sensors and actuators) transmit the information to cyber space through sensing a real world environment; also they reflect policy of cyber space back to the real world (Park et al. 2012).

Rather than dealing with standalone devices, cyber physical systems are designed as a network of interacting elements with physical inputs and outputs, similar to the concepts found in robotics and sensor networks. The main challenge in developing a CPS is to create an interactive interface between the physical and cyber worlds; the role of this interface is to acquire the context information from the physical world and to implement context-aware computing in the cyber world (Lun and Cheng 2011). Figure 1 illustrates a conceptual framework for building context-aware cyber physical systems (Rattadilok et al. 2013), adapted from a widely used modern sensor system reference model standardised by the CENSIS Innovation Centre for Sensor and Imaging Systems (www.sensorsystems.org.uk). The component parts and function of this reference model need to be delineated by function and interface in order to effectively develop effective instrumentation system in particular and cyber physical system in general.

Each layer of the framework is dedicated to a certain context processing task, ranging from low-level context acquisition up to high level context application using either existing or acquired knowledge. In particular, the context acquisition layer corresponds to the exploration of the available sensory data, including their visual representation, identification of the appropriate sampling periods, and data transformation (for example, differencing) for further analysis. The context processing layer deals with pre-processing of measured signals (e.g. identification of outliers, signal validation, etc.) and with detection of their salient features (e.g. the presence of outliers). The main function of the second layer is to make necessary preparations for building data-driven models with good generalisation capabilities. Of particular interest to the authors are the models based on computational intelligence techniques artificial neural networks, support vector

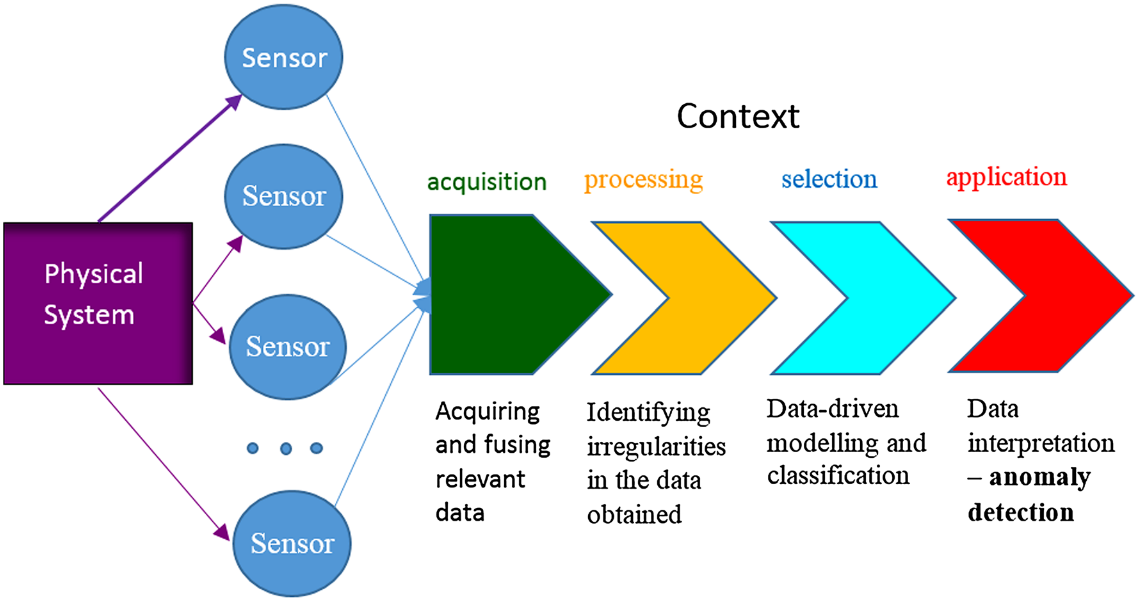

Context
Physical System
Sensor
Sensor
Sensor
Sensor
acquisition
processing
selection
application
Acquiring and fusing relevant data
Identifying irregularities in the data obtained
Data-driven modelling and classification
Data interpretation – anomaly detection

Fig. 1 Framework for designing context-aware CPS

machines, etc., built and tuned with the help of genetic algorithms, particle swarm optimization and artificial immune systems. The remaining layers of the proposed framework operate at a higher abstraction level. The third context selection layer is responsible for building, evaluating, and correcting (if necessary) the data-driven models based on empirical data supplied by the lower layers. The final context acquisition layer purports to examine the outputs of the models built at the previous layer in order to obtain or refine knowledge about the principles or rules that govern the dynamics of the processes under investigation (Petrovski et al. 2013).

Of a particular interest in the context of the present work is the data acquisition and processing layers that in context-aware CPSes are often implemented on the basis of intelligent and evolving sensors. Figure 2 illustrates a possible structure of evolving CPS sensors, wherein the adaptation or evolution of the sensors is done through building a data-driven process model (typically implemented in the context selection layer of the framework) and its tuning using machine learning techniques (Rattadilok et al. 2013). Thus, referring back to Fig. 1, the context processing and selection layers of the CPS framework are merged together to form evolving sensors within the CPS under investigation.

Cyber physical systems may consist of many interconnected parts that must instantaneously exchange, parse and act upon heterogeneous data in a coordinated way. This creates two major challenges when designing cyber physical systems: the amount of data available from various data sources that should be processed at any given time and the choice of process controls in response to the information obtained.

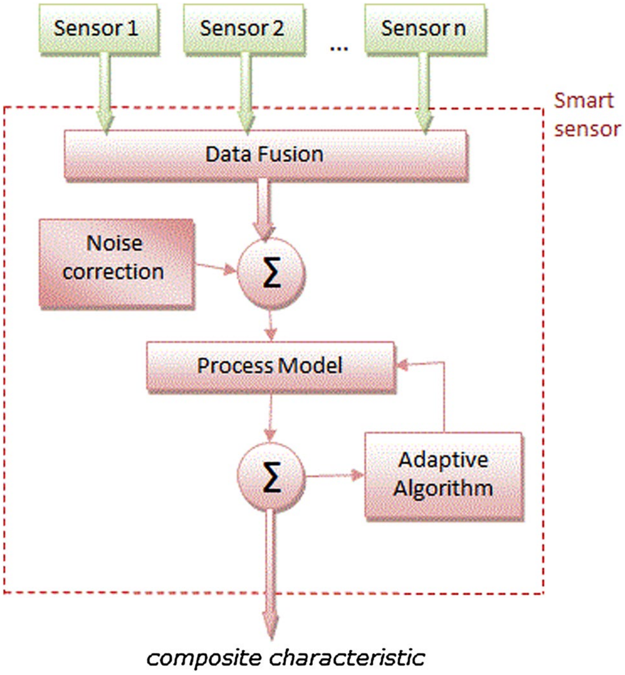
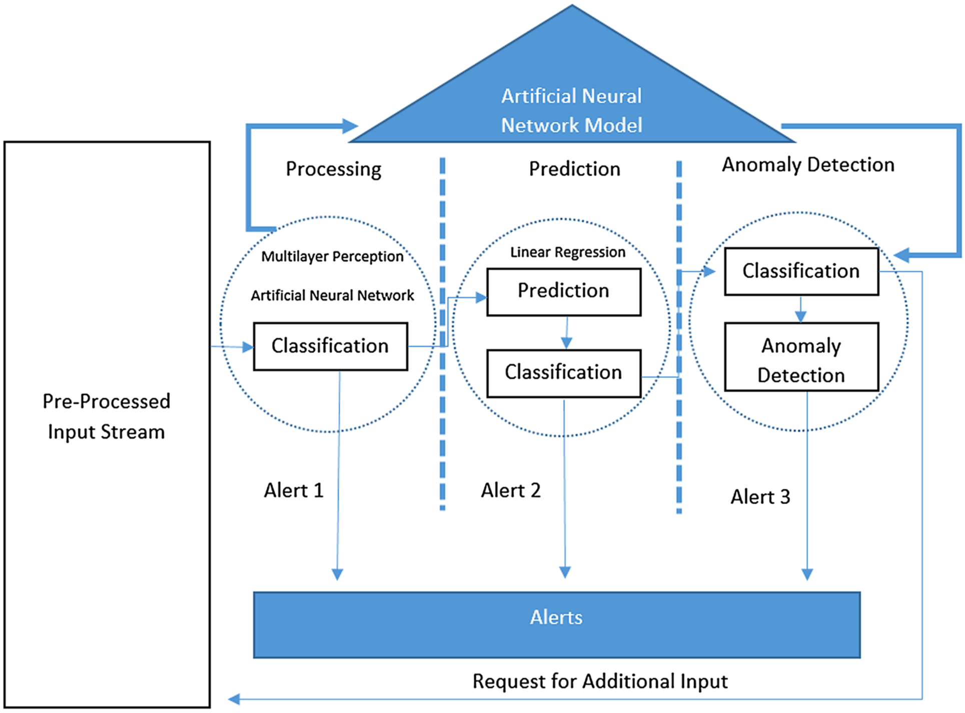


Fig. 2 Structure of an evolving sensor

An optimal balance needs to be attained between data availability and its quality in order to effectively control the underlying physical processes. Figure 3 illustrates a systematic approach to handling the challenges related to context processing, which has been successfully applied by the authors to various real world applications (Petrovski et al. 2013; Rattadilok et al. 2013).

Fig. 3 Systematic approach to context processing



As can be seen from Fig. 3, the suggested approach segregates processing of the input stream into three distinct phases. The Processing phase minimises the volume of data and the data processing cost by analysing only input streams from easy to obtain data sources using context selection techniques for finding anomalies in the acquired data. If any anomalies are detected at this stage, Alert 1 gets activated. This phase of the process is used to analyse real-time data and is a safe guard process on scenarios where the frameworks prediction fails to highlight an occurrence of unexpected changes in the environment.

In the Prediction phase, future values of each sensor in the CPS under investigation (gas turbine's sensor in our case) are estimated, using a linear regression model. Moreover, a new parameter is added which gets populated with the "predicted status" value for each data instance, indicating with Alert 2 whether any of the future predicted value of the sensors goes beyond the set threshold.

The final step of the process—Anomaly Detection—classifies the meaning and implications of overall predicted future values so that anomalies being present in the underlying operation process of the cyber physical system are shown. If any anomalies are detected at this stage, Alert 3 is triggered.

Such an approach allows for the acquisition of data and/or activation of the necessary physical entities on an ad-hoc basis, depending on the outcome at each phase. Moreover, the accuracy attained at the specified phases can be enhanced by incorporating additional data sensors or additional environmental factors. Computational intelligence techniques and expert systems have been successfully applied to tackling many anomaly detection problems, where anomalies are known a priori (El-Abbasy 2014). More interesting, however is to detect previously unseen anomalies, especially for real-time control of the cyber physical systems, which is the focus of the approach suggested in the current paper.

Statistical analysis and clustering are examples of techniques that are commonly used when the characteristics of anomalies are unknown (Chandola et al. 2009). Figure 4 illustrates a more detailed process for the systematic approach where machine learning and computational intelligence techniques are combined to tackle the unknown anomalies and learn from the experience when similar anomalies occur again. In Fig. 4, a circle labelled "b" represents a belief function of the output from both the statistical analysis and computational intelligence nodes, such that

$$f(t) = \sum_{i=1}^{n} w_i \mu_i(\overline{X}) + \sum_{j=1}^{m} w_j \eta_j(\overline{X}) \tag{1}$$

The weights (w_i and w_j) of this belief function are adaptively adjusted depending on how much knowledge related to the problem context has been obtained. The contribution of the CI nodes increases with collection of more normal and abnormal data points that can be used for training. This allows the system to run autonomously if required, and any potential anomalies are flagged for closer inspection at the anomaly classification phase.

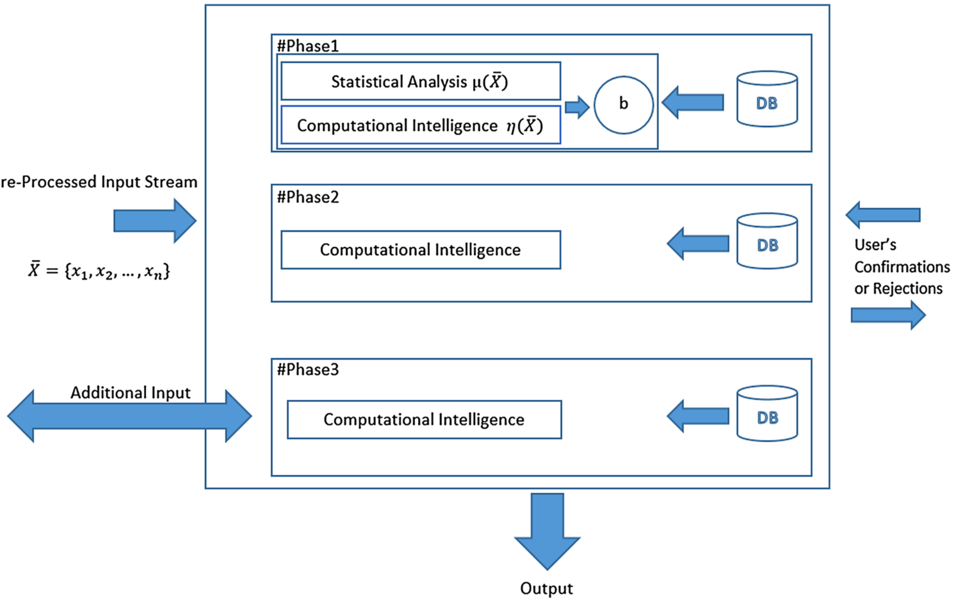

#Phase1
Statistical Analysis $\mu(\bar{X})$
Computational Intelligence $\eta(\bar{X})$
b
DB
Pre-Processed Input Stream
$\bar{X} = \{x_1, x_2, \dots, x_n\}$
#Phase2
Computational Intelligence
DB
User's Confirmations or Rejections
#Phase3
Additional Input
Computational Intelligence
DB
Output

Fig. 4 Context processing in a CPS

With the use of parallelisation and/or distributed systems, multiple machine learning, CI techniques and various belief functions can be evaluated simultaneously with their parameters being adaptively chosen. Anomaly identification using a combination of such techniques, as described in Fig. 4, has been successfully applied to a traffic surveillance application (Rattadilok et al. 2013), a smart home environment and automotive process control (Petrovski et al. 2013), and in some other applications (Duhaney et al. 2011).

3 Experimental results

It is a common practice that most of the sensory data on a platform are stored in a historian system (e.g. the PI system), which act as a repository of sensory information gathered from one or multiple installation. For this study we used historical sensor data of a gas turbine from an offshore installation in the North Sea. This data in real-time is transmitted offshore via satellite Internet. The integration of smart sensors with networked computing clearly indicated the appropriateness of considering the gas turbine under investigation as an example of a cyber-physical system, since it utilises the computing–networking combination.

3.1 Data monitoring flow

Data from most of the Turbine's sensors goes straight to the connected High Frequency Machine Monitoring System (HFMMS). This is due to the high volume of data generated every fraction of a seconds, which makes it almost impossible for any other system to handle such data volumes. These sensor values are then passed into Conditional Monitoring System (CMS) to prevent any possible system failure, without resorting to the ground truth values rarely available for real CPSes, in particular used in the oil and gas industry. On CMS there are varieties of formulae and thresholds to measure and assure safety conditions and efficiency of the turbine. These sensors' data, although not very important as part of the CMS, nevertheless is used for controlling different divisions of the Turbine and is passed straight into HMI. In addition to this, FMMS is also connected to HMI, enabling the SCADA software on the HMI to read all sensor values from Turbine as well as being able to write some values into some of actuators. On the HMI there is also another software called OPC Server, which is capable of writing data into OPC client that then writes data into the historian. The proposed Cyber Physical System then reads data from historian, as shown in the Fig. 5 illustrating the entire process.

3.2 Data cleaning process and challenges

One of the challenges in exporting data from a historian system, such as PI, is the necessity of interpolating values that are calculated by the PI system during the export process and are not real data. Another challenge is that some sensors can have an assigned text when the value goes below (or beyond) the admissible range and that text get written to the PI system instead of an actual number. For example, for some of the sensors during the reboot process the word, **Configuration** get stored in PI instead of a value;another example of this is **I/O timeout**, which gets written into PI when a connection to a sensor is temporarily lost. Unfortunately in such scenarios, where the expected value is a number rather than text, the entire instance needs to be removed since it is problematic for many machine learning algorithms

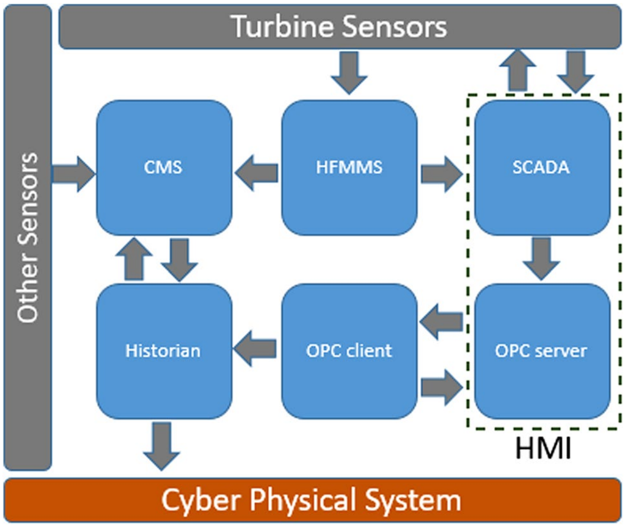


Fig. 5 Data monitoring flow

to combine textual and numeric data in the same input processing approach.

The issues, which highlighted above and any other similar issues, make the use of approaches such as time-series very hard since after deleting the records with missing values the expected time frame changes. As a result, to be able to work with the data we need to use bigger window size, which might not be ideal and fail to capture vital information. Another challenging issue with the output dataset from the PI system is selecting the right sensors for the study. The reason for that is because not only the oil and gas installations are packed with many sensors, but in the majority of cases having redundant sensors is very common—having many sensors can make selection of the right data input channels very difficult. However as well as being a challenge, these redundant sensors might work to an advantage where by comparing the value of the main and redundant sensors, it becomes possible to validate sensor inputs before writing them into the data historian.

Notwithstanding, care should be taken while doing input validation, because sometimes the value **Doubtful** gets written to the PI system, indicating a potentially broken sensor. In addition to this, it might also happen from time to time that a sensor due to various reasons temporarily goes offline, which in those scenarios results in an **Out of Service** message written to PI. Therefore removing instances from the dataset due to various reasons as illustrated above, makes it difficult to have a solid dataset. By trial and error, from the initial interval of every 1 s we have increased the dataset into the interval of 5 s to populate the sensor dataset used for further investigations.

Another vital challenge in the area of data cleaning is the attribute selection. In something like gas turbine the whole system is packed with 800 + sensors and to be able to run any study on that we needed to reduce number of attributes to around only 20 sensors.

For this study, initially 432 sensors have been identified, which assumed to have direct impact on the performance of the turbine engine. However these factors are mainly selected based on those sensors, which are used as part of calculation in CMS to monitor real-time malfunctions. To identify the significance of each of these contributing sensors, we used a factorial Design subcategory of Design of Experiment using the Minitab statistical software. A factorial design aims at carrying out experiments to identify the effect of several factors simultaneously. To conduct this experiment, instead of varying one element at the time, all the factors change concurrently. The most common approaches for conducting the studies is to run either a Fractional or a Full Factorial Design. One of the known approach to Full Factorial Design is 2-level Full Factorial, when the experimenter assigns only two value of maximum and minimum to each factor. Therefore the number of runs necessary for a 2-Level Full Factorial Design is 2^k where k is the number of factors.

Since Minitab only allows total of 15 factors for each experiment, similar sensors have been grouped into a total of 29 groups and a separate set of experiments have been ran on each sensor group. Having a scenario where 15 elements match the expected pattern is very rare, therefore percentage thresholds have been used as part of the filtering process. Depending on the expected minimum or maximum value for each sensor, as part of the full factorial scenarios, thresholds have been added in accordance with the following formulae:

$$f(x) = n_{max} - ((n_{max} - n_{min}) \times \tau) \tag{2}$$

$$f(x) = n_{min} + ((n_{max} - n_{min}) \times \tau) \tag{3}$$

If an instance of the dataset satisfies the scenario, the record with the performance rate of the engine gets stored into a file for further analysis. This means for each scenario multiple instances satisfy the requirement. After going through all the elements, a new cut down version of the dataset gets formed. Then once more application goes through all the scenarios, one by one, and if the scenario expect more minimum values than maximum values the least sensor value of all the instances get selected and vice versa for the maximum value. Moreover, if the expected minimum and maximum are equal, then the average performance value of the instances get selected. This process lead to a single performance value for each scenario, which will then gets feed into Minitab. Generated p-values using Minitab then help to identify statistically significant factors. Since each p-value is a probability, it ranges from 0 to 1, and measures the probability of obtaining the observed values due to randomness only, therefore the lower the p-value of a parameter is the more significant this parameter appears. If the p-value of a factor is less than 0.05, this means that a factor is statistically significant.

This approach has been used on three-month worth of data from a PI historian, which led to a selection of total 25 sensors from different parts of the gas turbine out of initial 432 sensors (see Fig. 6). Within this period system experienced eight failures, which are indicated by blue arrows in Fig. 7. The sample data for the 3-month period includes around 217,000 instances. Sensors used from the turbine are listed in Table 1.

In addition to all the sensors we also had a turbine status, which has each of the instances of the dataset labelled as either False, True or I/O timed out. False indicates the turbine failure state, True indicates the engine is running, and I/O Timed out indicates when the engine is getting restarted or communication between the PI historian and offshore is temporarily lost. The importance of having the I/O Timeout state is to prevent the system from sending an alarm when the system is actually in a state of reboot.

Table 1 Gas turbine sensors

Sensor description	Unit	Count
Power turbine rotor speed	rpm	2
Gas generator rotor speed	rpm	2
Power turbine exhaust temperature	F	6
None drive end direction	mm/s	1
Drive end vibration × direct	um P-P	1
Turbine inlet pressure	psia	1
Compressor inlet total pressure	psia	1
Ambient temperature	F	1
Axial compressor inlet temperature	F	2
Mineral oil tank temperature	F	1
Synthetic oil tank temperature	F	1
OB bearing temperature	C	1
IB bearing temperature	C	1
IB thrust bearing temperature	C	1
OB thrust bearing temperature	C	1
Generator active power	Mwatt	1
Grid voltage	V	1

3.3 Processing

The Processing phase of the proposed context-aware CPS implements a computational intelligence technique [an artificial neural network (ANN)] to classify the input stream. The ANN was chosen as many studies have shown that it is the most effective classification model to predict the condition of offshore oil and gas pipelines on varieties of factors, including corrosion (El-Abbasy 2014; Schlechtingen and Santos 2011). Also these studies highlighted the effective use of Bayesian and decision tree approaches in condition-based maintenance of offshore wind turbines (Nielsen and Srensen 2011). Random Forest Tree is another algorithm, which is widely used in the field of predictive maintenance in the oil and gas industry to forecast a remote environment condition, where visual inspection is not sufficient (Topouzelis and Psyllos 2012). Moreover, additional algorithms used to detect anomalies on offshore turbines includes k-Nearest Neighbour (kNN), Support Vector Machine (SVM), Logistic Regression and C4.5 decision tree (Duhaney et al. 2011). Based on these studies,seven algorithms have been compared to identify the best performing one. These algorithms are listed in Table 3. Moreover Table 2 lists the most significant hyperpatameters used for each algorithm.

3.3.1 *Evolving process*

In the processing phase when the input stream is analysed and classified, it gets appended to the training dataset.The whole framework is wrapped by a Linux bash file and gets

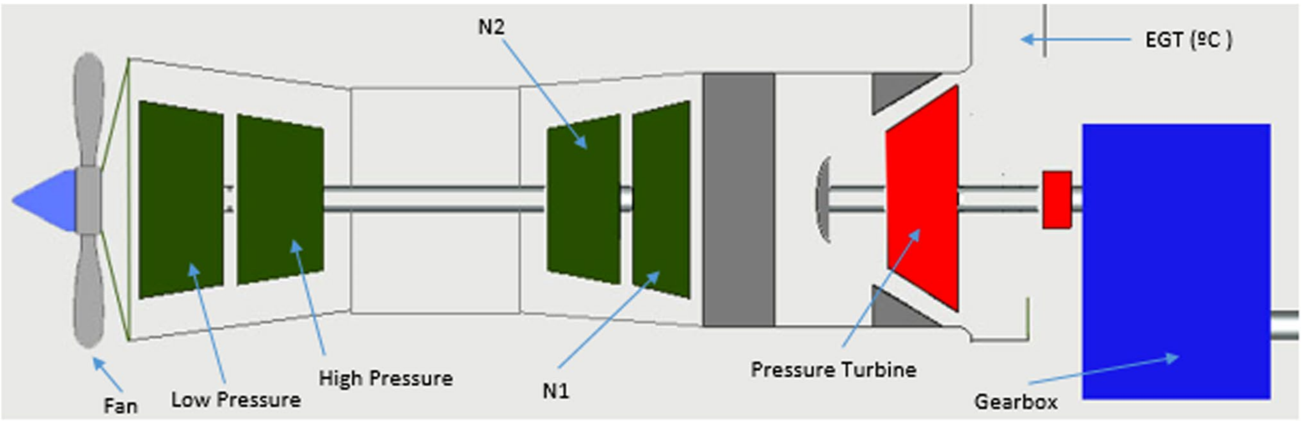


Fig. 6 Gas turbine process design

Fig. 7 Turbine's fail scenarios

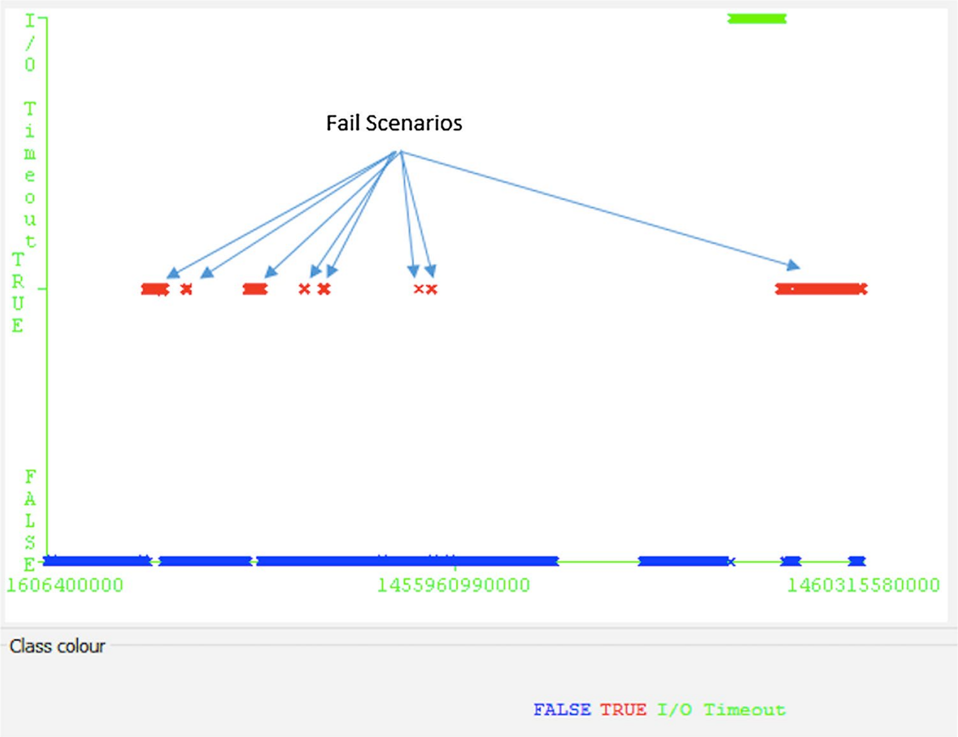


Table 2 Comparison of algorithm performance

Algorithm	Hyperparameters
Multi-layer perceptron (MLP) neural network	Iteration: 5000 Hidden layers: 4 Neurons per layer: 24
C4.5 decision tree (%)	Confidence factor: 0.25 Number of folds: 3 Minimum number of objects: 2 Number of leaves: 30 Size of the tree: 59
Decision tree random forest	Minimum number of records per node: 10 Number of threads: 4 Quality measure: Gini Index Number of leaves: 29 Size of the tree: 58
k-nearest neighbour (%)	Number of neighbours to consider (k): 3
Support vector machine (SVM)	Overlapping penalty: 1.7 Kernel:polynomial Power: 1.3 Bias: 0.7 Gamma: 0.3
Logistic regression (%)	–
Nave Bayes (%)	Default probability: 0.004 Maximum number of unique nominal values per attribute: 20

Table 3 Comparison of algorithm performance

Algorithm	Accuracy (%)	Error (%)
Multi-layer perceptron (MLP) neural network	100	0
C4.5 decision tree (%)	94.74	5.26
Decision tree random forest	94.73	5.27
k-nearest neighbour (%)	94.07	5.93
Support vector machine (SVM)	87.21	12.79
Logistic regression (%)	46.5	53.5
Nave Bayes (%)	40.45	59.55

executed using a timer every 5 min. To prevent downtime while the framework cycle gets restarted with the updated training dataset there are two parallel virtual machines called primary and secondary. The two machines run side by side. The secondary virtual machine runs the cycle with 2.5 min lag which provide enough time for the primary virtual machine restart the cycle with the updated training dataset before itself restart the cycle. the process create a continues monitoring system which every 5 min evolves and retrain the model with the updated dataset without downtime.

As it is illustrated in Table 3, Multi-Layer Perceptron (MLP) Neural Network generates the best result amongst other algorithms. All the results listed are the best results for each of the algorithms considered, obtained by adjusting their hyper-parameters to achieve the best performance using the Auto-Weka package for comparing CI techniques. Therefore to implement the Processing phase of the suggested framework, a Multilayer Perceptron is used, which is a feedforward Artificial Neural Network (ANN). Funahashi

(1989), Hornik et al. (1989) and Qin et al. (2016) have all shown that only one hidden layer can effectively generate highly accurate results and to improve the processing time. Therefore initially an ANN Multilayer Perceptron with Backpropagation of error with one hidden layer has been used. However, in addition to that the chosen algorithm has been been trained with 1, 2, 3 and 4 hidden layers and tenfold cross validation. The experiments had been carried out up until four hidden layers, which eventually generated an excellent result. Table 4 lists the results obtained from the experiments with 1–4 hidden layers.

Although by using only one hidden layer we have managed to classify 92.77% of the instances correctly, by increasing the number of hidden layers to 4, all test instances could be correctly classified.

Figure 8 illustrated an artificial neural network design. The input layer corresponds to the 25 input sensors of the gas turbine. The middle layers are used to form the relations between the neurons, their number being determined at runtime. The output neurons are the three classifications which indicates the status of the turbine.

Table 4 ANN multilayer perceptron optimisation

Layers count	One	Two	Three	Four
Correctly classified (%)	92.77	92.77	94.95	100
Incorrectly classified (%)	7.23	7.23	5.05	0
Kappa statistic	0.60	0.60	0.74	1
Mean absolute error	0.09	0.09	0.062	0
Root mean squared error	0.21	0.21	0.17	0
Relative absolute error (%)	57.32	57.79	39.10	0.34
Root relative squared error (%)	74.89	74.97	62.71	0.77
Coverage of cases (0.95 level) (%)	100	100	100	100
Mean rel. region size (0.95 level)	4.65	64.65	55.25	33.33

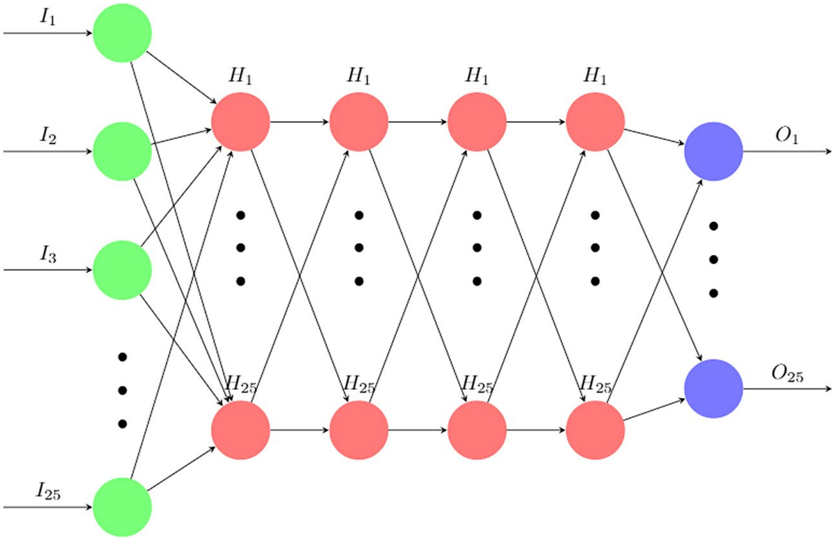
3.4 Prediction

The second stage of the proposed model is the Prediction Phase. The purpose of this phase is to predict the future values for the next 24 h of all 25 sensors. During this phase three-month historical data has been used to train a linear regression model for each sensors. In addition to that, the thresholds for each of the sensors, provided from currently installed Conditional Monitoring System have been used to set threshold alarms. After training the models the developed anomaly detection framework was put into practice for each sensors times series with the lag period of 24 h for each sensor to predict the next 24-h datasets. Therefore, if any of the predicted values for each of the sensors fall below or beyond the allowed threshold interval, then Alarm 2 gets activated. Figure 9 illustrates the predicted results for all the 25 sensors chosen.

3.5 Anomaly detection

Since the combination of all the sensors together reflects the status of the turbine, after predicting future sensor values, all the predicted values get merged into a single test dataset. A Multi-Layer Perceptron (MLP) Neural Network model, which has been selected as the best performing algorithm as part of the Processing phase, was used again for labelling the status of the turbine for each of the instances. After predicting the status of the turbine for all instances of the dataset, the developed framework iterates through all labels and, if any of the instances are labeled as failed, Alarm 3 gets triggered. The system then picks the timestamp of the predicted time and deducts it from the current time to provide the estimated hours left until the system failure. In the final step of the Anomaly Detection phase, the total remaining hours gets included into an automatically generated email and is

Fig. 8 ANN multilayer perceptron proposed model



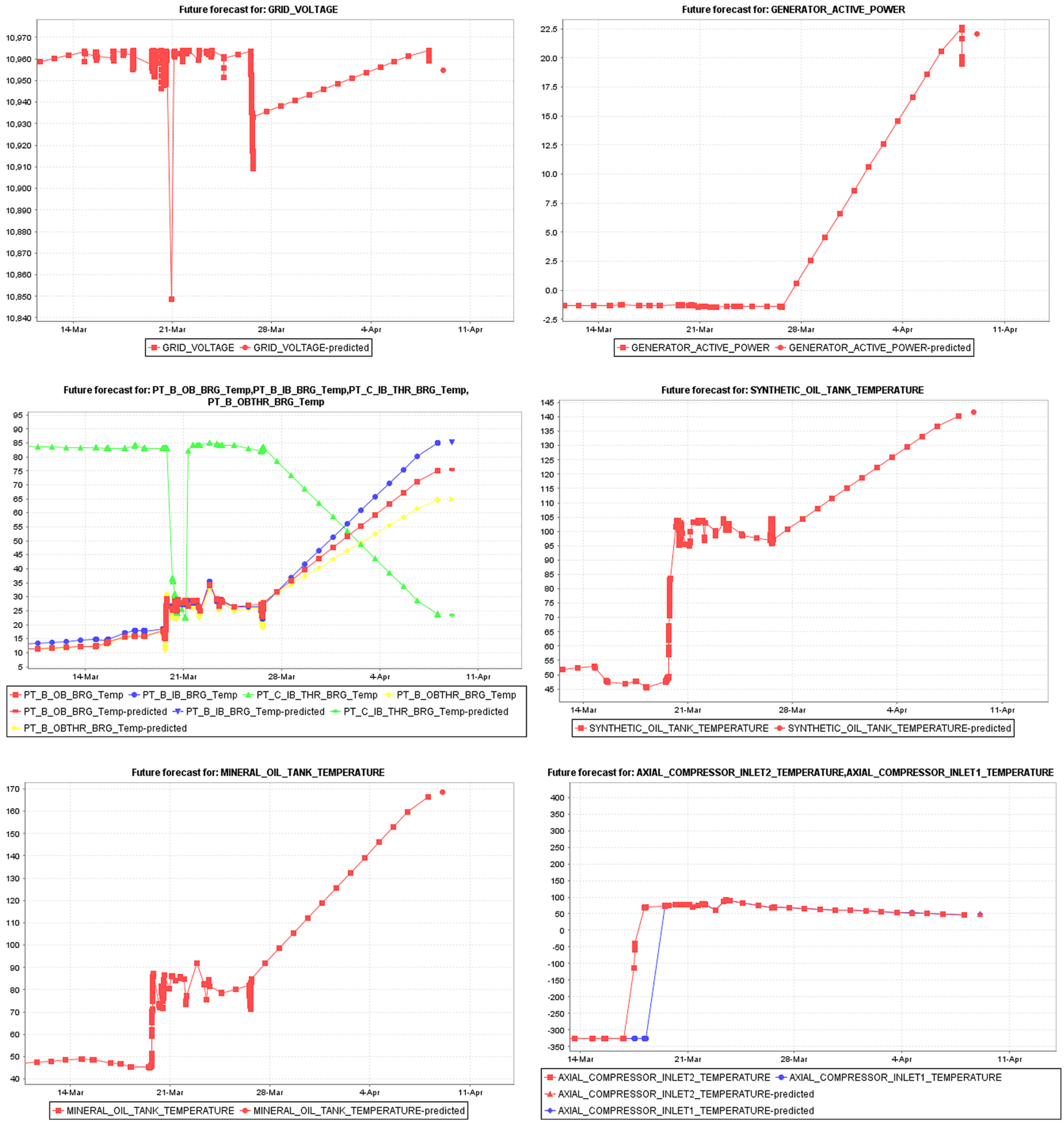


Fig. 9 Predicted sensor values

sent out to the preset list of email addresses, as well as playing an audio alarm on the PC.

3.6 Overall automated process

Initially Weka (www.cs.waikato.ac.nz/ml/weka/) was used to run each of the phases separately. However, in the final stage of the process we have actually formed the proposed framework using Knime (www.knime.org/). Knime is an open source data analytics, reporting and integration platform. Although there are other alternatives, such as Weka's KnowledgeFlow and Microsoft Azure's Machine Learning, Knime was chosen since it has the capability of importing most of Weka's features through the addition of a plugin. Also being able to run java snippets and write the developed model into disk to free up space on memory, it

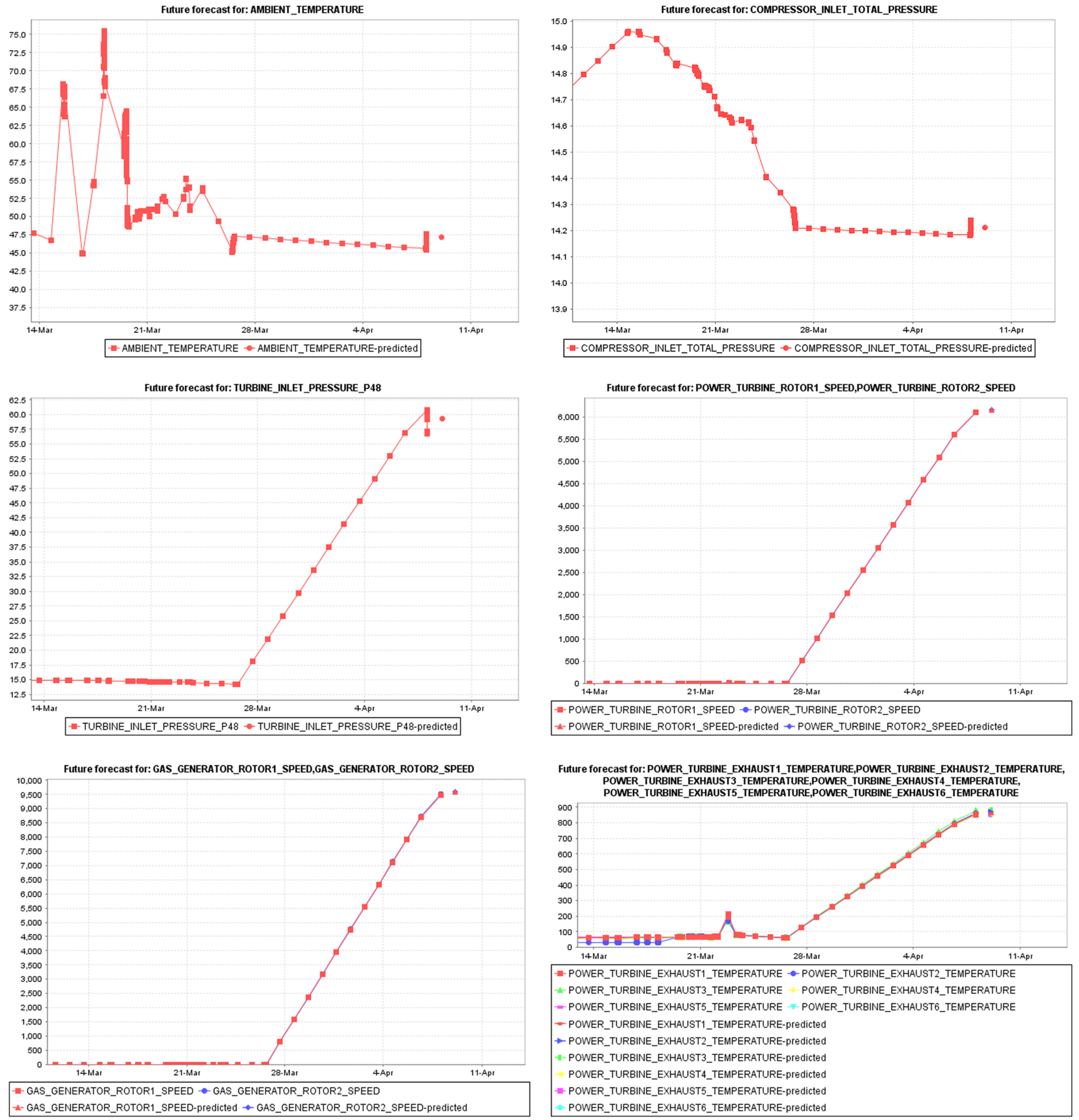


Fig. 9 (continued)

was considered to be a preferred option in comparison to Azure's Machine Learning.

The dataset was divided into two sets of training and test data as illustrated in Fig. 10. Two-month worth of data was used for training, which included eight cases of turbine failure, with the remainder set aside for testing. The training dataset has been used to form an Artificial Neural Network Multilayer Perceptron (MLP) using backpropagation of error (Pal and Mitra 1992).

After training the model, it was tested against the developed ANN MLP to classify the status of the engine. This implementation covered the Processing phase of the proposed Cyber Physical System. This was followed by introducing times series lag and a linear regression model to

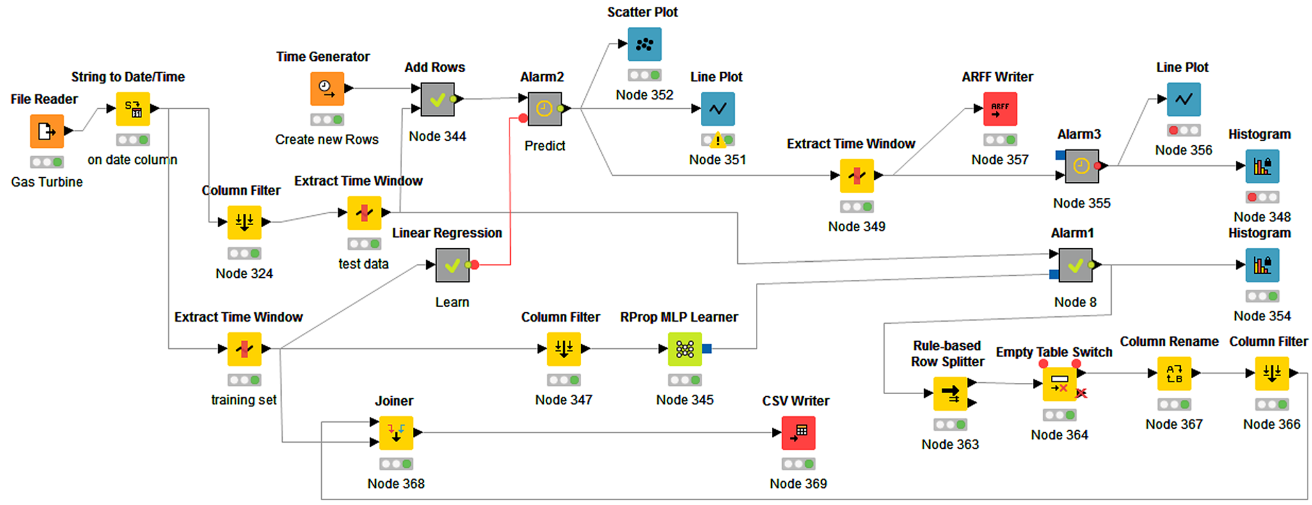


Fig. 10 Overall automated framework of the process

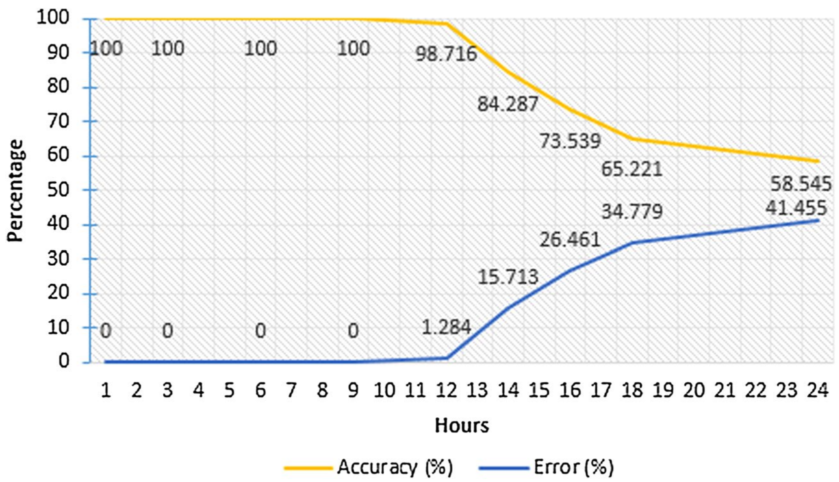
predict the next 24 h on the test dataset. By looking at the eight failure situations, corresponding thresholds were identified for each of the input sensors based on the pre-labeled dataset generated by the CMS. Therefore, if during the prediction stage any of the sensor's value go below or above the set threshold, the second Alarm goes off. However, this alarm is an amber alarm, which does not necessarily imply that the turbine will fail. With 24 h of predicted data for the sensor data gathered in the final stage of the process, all the predicted data is put together as a test dataset and is tested against the model developed in the Processing Phase. If the status of the engine gets classified as False, then the third and last alarm gets activated.

3.7 Evaluation

To test the accuracy and performance of the proposed model, we have divided the available dataset which covers a total of four months into four separate datasets. Then the test for each month was run individually by removing 5-day worth of data from each dataset. This led to developing a model used to predict each of eliminated days on an hourly basis. To achieve this, the operational performance of the turbine for the next 1, 3, 6, 9, 12, 14, 16, 18 and 24 h on each days has been predicted. Then the average performance across these 5 days was estimated, using twenty experiment that have been averaged, as illustrated in Fig. 11.

The average performance shows that within the first 12 h the proposed framework could predict the status of the turbine with nearly 99% accuracy, which is a very high performance. Even for the 15 h period, prediction was around 84.28%, where other studies (Topouzelis and Psyllos 2012) support that predictive accuracy over 84% by having 25 features (sensors) or above is considered to be of high performance. Also, Naseri and Barabady (2016) showed the waiting downtime associated with each item of corrective

Fig. 11 Hourly performance evaluation



maintenance for gas turbine is considered to be about 7210 h. Therefore having performance of even 73% after 16 h can very effectively reduce the downtime by 20%. However, after 18 h the prediction performance shows a sudden decline and, when it gets to prediction of the next 24 h, the result is really poor by being around 58%. Table 5 lists the average value of the results for each prediction.

Figure 12 shows the correlation of the 4 factors of Rotor speed, Exhaust temperature, generated active power and the prediction in processing phase of the framework. As the speed of rotor increases, this results in a rise of exhaust temperature, which leads to higher generated power. The figure illustrates the prediction is clearly matching the scenario where the rotor speed and exhaust temperature is low, system is generating low power or it is in the fail state.

As illustrated in Fig. 13, the prediction phase of the framework where future values of the sensors are predicted and correlation of the values matching what is expected where when Rotor speed is increasing, the predicted values of Exhaust temperature is increasing, so is the expected Generator Active Power.

Table 5 Comparison of real-time status vs. predicted status

Hours	Accuracy (%)	Error (%)
1	100	0
3	100	0
6	100	0
9	100	0
12	98.716	1.284
14	84.287	15.713
16	73.539	26.461
18	65.221	34.779
24	58.545	41.455

To visualise the accuracy of the framework in the Anomaly Detection phase (Fig. 14), the correlation between Rotor Speed, Exhaust Temperature, Generator active power and the framework's prediction are shown. As expected, in the scenario illustrated in the Figure, although all three input values showing an increase in expected correlation, but all the performances are below the expected rate and, as a result, the Turbine state is identified as failure, which is the expected result.

4 Conclusion

An implementation of a context-aware cyber physical system using evolving inferential sensors for condition monitoring to predict the status of a gas turbine on an offshore installation has been successfully developed. In this research, a three-phase approach has been proposed: In the processing phase, historical data of 25 sensors was collected from different areas of turbine to train an evolving component (ANN-based) used as the basis of the prediction model. In the second phase, future value of each physical sensor were predicted for a certain period of time using linear regression. The final phase makes use of the model developed in phase one to label the predicted data in order to detect anomalies prior to their occurrence.

The developed evolving sensor proved to be capable of highly accurate predictions of gas turbine status up to 15 h in advance with the accuracy of about 84.28%. The clear challenge in these sort of problem is dealing with imbalanced data and taking advantage of a time-series algorithm, such as

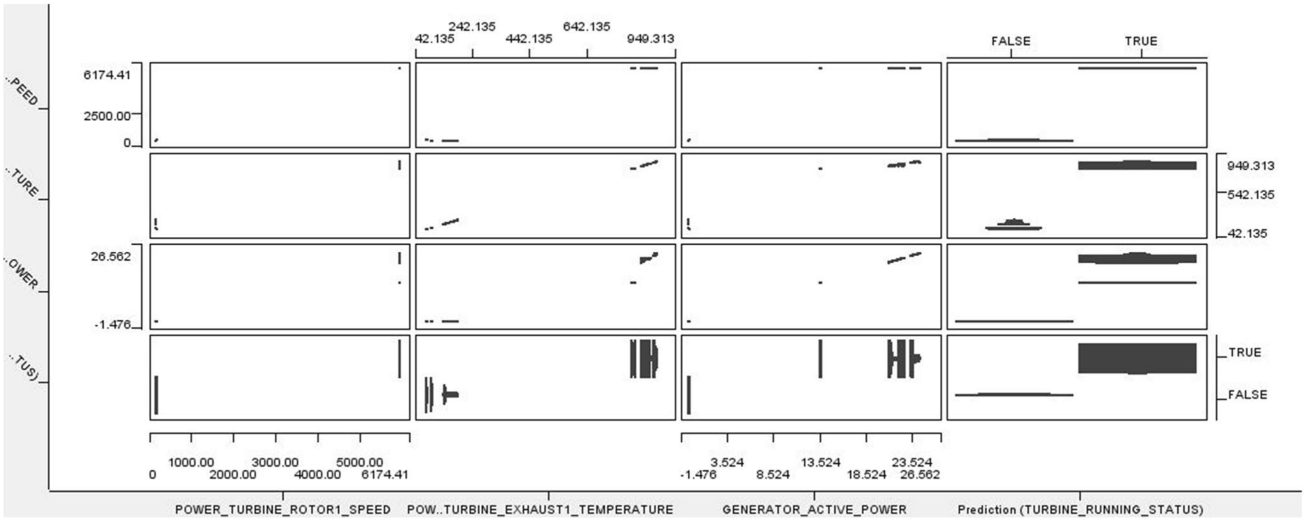


Fig. 12 Processing output

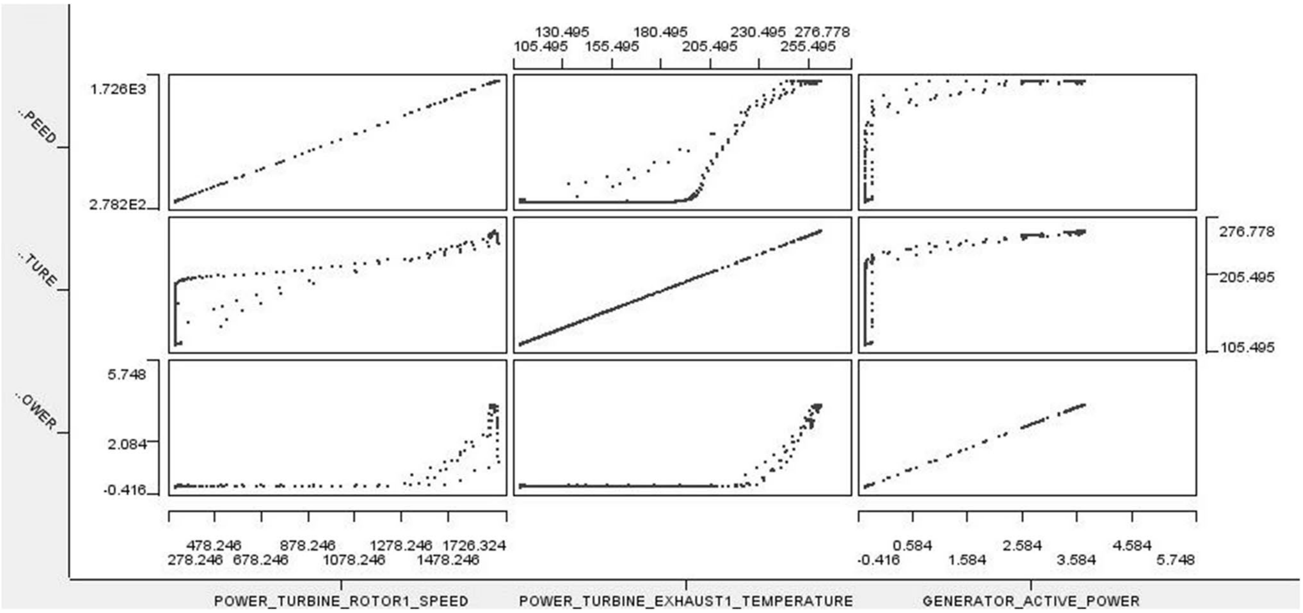


Fig. 13 Prediction output

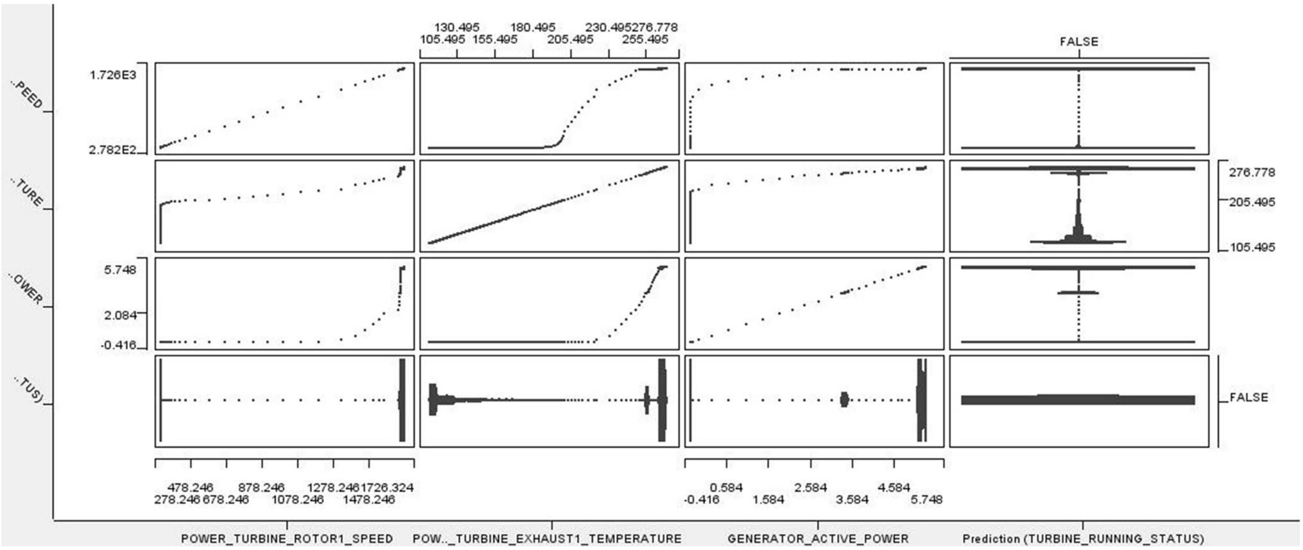


Fig. 14 Anomaly detection output

time series prediction with Feed-Forward Neural Networks, to help improving the length of a predication period. Further research will focus on normalising the imbalanced data using approaches, such as ensemble methods that include bagging and boosting, as well as extending the prediction time frame by assuring high accuracy in anomaly identification through exploring various combinations of computational intelligence techniques with conventional classification approaches.

References

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv (CSUR) 41(3):15

Duhaney J, Khoshgoftaar TM, Sloan JC, Alhalabi B, Beau-Jean PP (2011) A dynamometer for an ocean turbine prototype: reliability through automated monitoring. In: 2011 IEEE 13th international symposium on high-assurance systems engineering (HASE). IEEE Computer Society, Boca Raton, FL, pp 244–251

El-Abbasy MS et al (2014) Artificial neural network models for predicting condition of offshore oil and gas pipelines. Autom Constr 45:50–65

Funahashi K (1989) On the approximate realization of continuous mappings by neural networks. Neural Netw 2(3):183–192

Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366

Khan Z, Shawkat Ali ABM, Riaz Z (eds) (2014) Computational intelligence for decision support in cyber-physical systems. Springer, Berlin

Lee EA (2008) Cyber physical systems: design challenges. Technical report no. UCB/EECS-2008-8. University of California, Berkeley

Lun Y, Cheng L (2011) The research on the model of the context-aware for reliable sensing and explanation in cyber-physical system. Proced Eng 15:1753–1757

Naseri Masoud, Barabady Javad (2016) An expert-based approach to production performance analysis of oil and gas facilities considering time-independent Arctic operating conditions. Int J Syst Assur Eng Manag 7(1):99–113

Nielsen JJ, Srensen JD (2011) On risk-based operation and maintenance of offshore wind turbine components. Reliab Eng Syst Saf 96(1):218–229

Pal SK, Mitra S (1992) Multilayer perceptron, fuzzy sets, and classification. Neural Netw IEEE Trans 3(5):683–697

Park KJ, Zheng R, Liu X (2012) Cyber-physical systems: milestones and research challenges. Comput Commun 36(1):1–7

Petrovski S, Bouchet F, Petrovski A (2013) Data-driven modelling of electromagnetic interferences in motor vehicles using intelligent system approaches. In: Proceedings of IEEE symposium on innovations in intelligent systems and applications (INISTA), FL, pp 1–7

Pierazzi F, Casolari S, Colajanni M, Marchetti M (2016) Exploratory security analytics for anomaly detection. Elsevier Comput Secur 56:28–49

Qin M, Song Y, Akagi F (2016) Application of artificial neural network for the prediction of stock market returns: the case of the Japanese stock market. Chaos Solitons Fractals 85:1–7

Rattadilok P, Petrovski A, Petrovski S (2013) Anomaly monitoring framework based on intelligent data analysis. Intelligent data engineering and automated learning (IDEAL 2013), vol 8206. Lecture notes in computer science, pp 134–141

Schlechtingen M, Santos IF (2011) Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. Mech Syst Signal Process 25(5):1849–1875

Topouzelis K, Psyllos A (2012) Oil spill feature selection and classification using decision tree forest on SAR image data. ISPRS J Photogramm Remote Sens 68:135–143

www.knime.org/. Accessed 25 Nov 2016

www.sensorsystems.org.uk. Accessed 17 Oct 2016

www.cs.waikato.ac.nz/ml/weka/. Accessed 15 Dec 2016