This is a peer-reviewed, accepted author manuscript of the following research article:
Young, A., West, G., Brown, B., Stephen, B., Duncan, A., Michie, C., & McArthur, S. (2022). Capturing symbolic expert knowledge for the development of industrial fault detection systems: manual and automated approaches.
International Journal of Condition Monitoring and Diagnostic Management, 25(2), 67-75.

COMADEM
International Journal of Condition Monitoring and Diagnostic Engineering Management

Capturing Symbolic Expert Knowledge for the Development of Industrial Fault Detection Systems: Manual and Automated Approaches

Andrew Young[a]*, Graeme West[a], Blair Brown[a], Bruce Stephen[a], Andrew Duncan[b], Craig Michie[a], and Stephen McArthur[a]

[a]*University of Strathclyde, 204 George Street, Glasgow, G1 1XW, UK*
[b]*The Alan Turing Institute, British Library, 96 Euston Rd, London, NW1 2DB, UK*

* *Corresponding author. Tel.: +44 141 444 7241; email: andrew.young.101@strath.ac.uk*

ABSTRACT

In critical infrastructure, such as nuclear power generation, constituent assets are continually monitored to ensure reliable service delivery through pre-empting operational abnormalities. Currently, engineers analyse this condition monitoring data manually using a predefined diagnostic process, however, rules used by the engineers to perform this analysis are often subjective and therefore it can be difficult to implement these in a rule-based diagnostic system. Knowledge elicitation is a crucial component in the transfer of the engineer's expert knowledge into a format suitable to be encoded into a knowledge-based system. Existing methods to perform this are extremely time-consuming, therefore a significant amount of research has been undertaken in an attempt to reduce this. This paper presents an approach to reduce the time associated with the knowledge elicitation process for the development of industrial fault diagnostic systems. Symbolic representation of the engineer's knowledge is used to create a common language that can easily be communicated with the domain experts but also be formalised as the rules for a rule-based diagnostic system. Additionally, an automated approach is proposed to capture and formalise the domain expert knowledge without the need for formal knowledge elicitation sessions. Two case studies are then presented using both the manual and automated approaches. The results show that using the manual approach it is possible to quickly develop a system that can accurately detect various types of faults, and also there is a significant time saving using the automated approach without an equivalent loss in accuracy.

Keywords: *Condition monitoring; nuclear power plants; expert systems; knowledge-based systems; automation.*
Article history: *Received ; Published .*

1. Introduction

Fault detection and diagnostics is an active research area, especially in the nuclear industry for rotating machinery [1][9][14][19]. The two most commonly used approaches that can be adopted for the development of systems for fault detection or diagnostics. These are either data-driven approaches, e.g. machine learning, or knowledge-based approached, e.g. expert systems. While both of these techniques have similar aims and can provide similar results they differ quite significantly in their implementation.

The basis for data-driven approaches is centred around statistical models of the problem data. The individual parameters of the model are learned through a process called training where a large volume of data is input into the model and the model attempts to produce the correct output for the majority of cases. It should also be noted that related to many data-driven approaches a balanced dataset is required, i.e. there is an adequate number of samples for both normal data and fault states. Due to the nature of these models and the lack of explainability for many data-driven approaches; for critical assets (especially in the nuclear industry) that can present an issue. This is because supporting evidence is often required when making decisions on these assets as there is a significant amount of cost involved in the repair, replacement or downtime of these assets, another consideration is that "black box" techniques cannot currently support the safety-case oriented practices of the nuclear industry for example.

Knowledge-based approaches are the second technique that can be used to solve this problem, they attempt to solve (or support the resolution of) complex problems where a significant amount of human expertise or expert knowledge is required. This knowledge is acquired from the engineers or domain experts through a process called knowledge elicitation, this is then formalised into a format that is compatible with the technique, e.g. as the rules for a rule-based expert system. The one main advantage of this type of approach over data-driven approaches is the ability to not only solve a problem but also to explain and justify the reasoning behind why a decision was made. However, this comes with the disadvantage that a significant time cost is associated with the capturing of the knowledge, then formalising this into a knowledge-based system. Because of this disadvantage, there has been a significant amount of research undertaken across numerous fields to streamline the knowledge elicitation process [12][17], as this is the most time-consuming part of the development of an expert system.

The next section of this paper provides background information into rule-based expert systems, a type of knowledge-based approach. Section three proposes a new methodology for knowledge elicitation through the use of symbolic representation of the expert's knowledge and the parametrization of this knowledge. Section four proposes an automated approach that can analyse historical data, then propose new diagnostic rules using the framework described in section three. While section five presents a case study of the manual methodology applied to boiler feed pumps from an advanced gas-cooled reactor in the UK. Section six presents a case study comparing the manual and automated approaches in respect to the time taken to develop the system but also the accuracy. Finally, the conclusions and future work are presented in section seven.

2. Rule-Based Expert Systems

Knowledge-based systems can be used for a variety of applications to provide not only accurate decisions but also the explanation and reasoning behind these decisions. One example of these is rule-based expert systems [8], which stores the knowledge captured from the domain expert as a set of rules. These rules are formalised in a way that mimics the domain experts reasoning process and are mainly applied to knowledge or time-intensive problems.

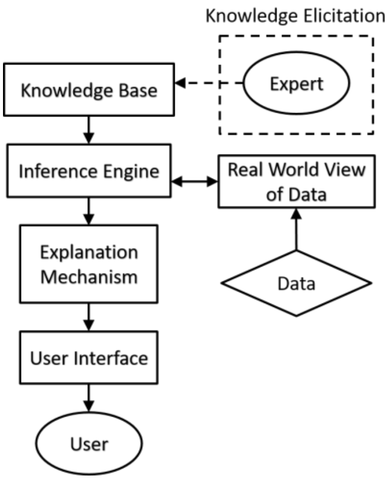

Figure 1: Typical rule-based expert system architecture

A typical rule-based expert system contains five main components, see Figure 1. The first of these is the **Knowledge Base** and which contains all the domain-specific captured knowledge from the experts. Figure 2 shows an example of how the rules are expressed in an expert system as a set of IF-THEN rules. This can be considered as a fixed set of data, therefore it remains the same throughout the decision making process. The **Real World View of Data** is the next component, this contains all the data, and facts relating to the asset under analysis. This can be considered the current state of the machine and hence is fluid and constantly changing. The facts relating to the asset are then compared with the IF condition in the knowledge-based to determine intermediate facts, which can then be stored in the Real World View of Data or a diagnostic conclusion. The third component is the **Inference Engine**, which performs the analysis by comparing the rules in the knowledge base to the facts stored

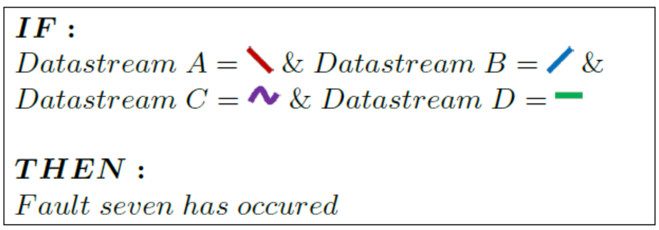

Figure 2: Structure of rules stored in knowledge base

in the real world view of data. The **Explanation Mechanism** provides justifications and an explanation as to why the inference engine has decided on a conclusion. This component is crucial for the system to be accepted by the user or by industry. Finally, a **User Interface** allows for communication between the user and the system, whether this is for the input of new facts relating to the data or the output of the diagnostic conclusions, this information can also be passed to external programs or systems.

3. Manual Symbolic Capture of Knowledge

For many industrial applications fault diagnosis involves the engineers following a predefined diagnosis process. Therefore, the expert knowledge has already been acquired to some extent, although this is not always complete enough to be formalised into a set of rules for a rule-based expert system. There is often a significant amount of subjectivity involved when the engineers assess the problem, due to esoteric experiences with the asset, rules of thumb, or different formal training. However, at a high level, they are often looking for standard data trends such as increases or decreases in specific data, or an increased noise or fluctuation. There is often no prescribed quantitative information relating to these trends that they analyse, such as how much increase or decrease relates to a specific rise or fall, or how much increase in fluctuation relates to a signal moving from stable to fluctuating, as these values will change based upon multiple factors, such as the type of machine, the age of the machine and the operational profile of the machine. Therefore, before the knowledge can be formalised into a rule-based expert system this additional knowledge must be acquired from the domain experts through the knowledge elicitation process. There are several different approaches for performing this knowledge elicitation [3], some of these include: structured and unstructured interviews: observation through active participation or focused observation; and task or decision analysis. For complex problems this is an extremely time-consuming task, this bottleneck in the development of a knowledge-based system has long been recognised and has hence been called the "knowledge elicitation bottleneck" [5].

The rest of this section focuses on a new methodology to streamline this knowledge elicitation process by simplifying the knowledge into a set of symbols, or common language, that can be easily communicated between domain experts and data engineer. The proposed methodology is a three-stage process that involves a minimum of two structured interviews.

3.1. Definition of Symbols

These symbols were selected as low-level predicates that could be used to broadly describe a time series at any instant. The trends that were selected are shown in Figure 3, these symbols are a stable symbol that relates to normal behaviour, a rise and fall symbol for an increase or decrease over a specific time with a specified limit, and a fluctuating symbol for an increase in noise present in the signal.

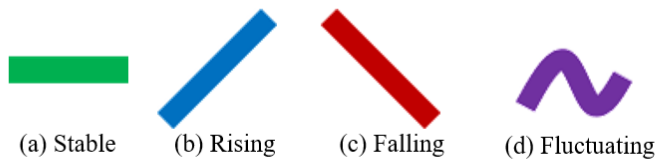

Figure 3: Four selected symbols/trends

These symbols were selected as they are the most basic trends that can be present in time-series data, and any complex trend can be constructed from these primitives. This allows for the domain experts to easily communicate the diagnostic process they follow using a common language.

3.2. Definition of Rule Base

The next stage of the process is to set up a structured interview with the domain experts to agree on a definition of the rule base. This requires a definition of the individual faults that are being analysed, the specific datastreams necessary to determine those faults, and the associated trends for each of those datastreams. For each fault, a table can be produced that contains all the information discussed above, the example format of this table is shown in Table 1. Additionally, any comments that the engineers can provide at this stage will also prove to be extremely useful, this develops a rationale behind each piece of knowledge and for example could be: the physical reasoning behind the associated trends: or clarification on a subset of faults where a full data set or other operational influence is unavailable to fully diagnose a specific problem.

Table 1: Example format for individual fault diagnostic rules

Datastream	Trend	Comments
Datastream A		
Datastream B		
Datastream C		
Datastream D		
Datastream E		

Following the meeting, each of the tables for the individual faults are combined to produce an overall rule-base for the asset being analysed, an example of this is shown in Table 2. Regarding system development, it is now possible to construct a prototype rule-based expert system using placeholder values for the quantitative parameters relating the each of the individual trends, which will be set in the next stage.

Table 2: Example format for asset-specific rule base

Cause	Datastream A	Datastream B	Datastream C
Fault 1			
Fault 2			
Fault 3			
Fault 4			
Fault 5			

3.3. Definition of Parameters

Having defined the necessary symbols to accurately interpret related data streams; agreed with the domain experts the individual faults and the associated trends used to assess these faults: the next stage is for all this information to be tabulated and parametrised. Subsequently, a second structured interview is arranged to determine the individual magnitudes for each specific trend associated with each specific rule. The previously mentioned symbols that are now shown in Figure 3 are regarded as the most basic trends that are present in the data. The expert knowledge that is required to qualify the diagnostic rules shown in Table 2 is the subtle differences in the trends in Figure 3. For each symbol various parameters must be assigned to them that accurately describe the possible variations in the symbols for different rules, this is shown in more detail in Figure 4 and Table 4. This information and the corresponding parameters can be tabulated and presented to domain experts in a structured interview knowledge elicitation session. An example of this structure is shown in Table 3. The parametrisation of the knowledge allows for efficient and accurate capture of the domain-specific knowledge by focusing the domain experts on a simplified version of the problem. This also facilitates the ease of formalising this knowledge into the rules for a rule-based expert system, without the need to listen to hours of audio recording or to interpret the engineer's answers to specific questions.

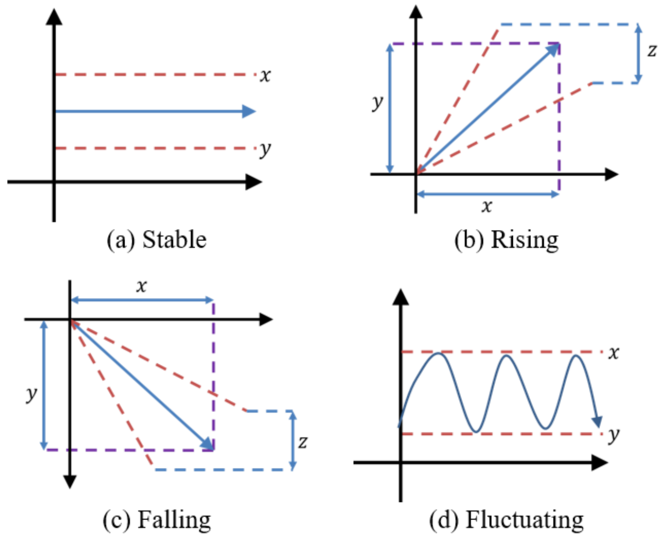

Figure 4: Definition of parameters for subtle difference in symbols/trends

Table 3: Example structure for rule specific table to be completed during knowledge elicitation session

Datastream	Parameters
Datastream A – Rising	$x =$,$y =$,$z =$
Datastream B – Falling	$x =$,$y =$,$z =$
Datastream C – Fluctuating	$x =$,$y =$
Datastream D – Stable	$x =$,$y =$

3.4. Implementation

After gathering all the expert knowledge from the knowledge elicitation meetings, the methodology proposed to evaluate the diagnostic rules on time series data is to first segment the data into specific time regions, see Figure 5. Splitting up the time

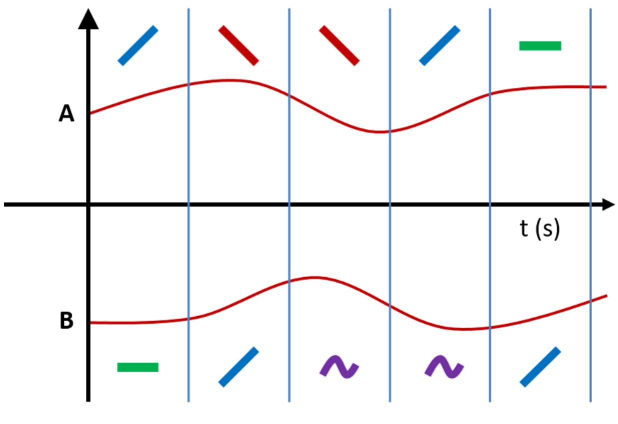

Figure 5: Example of signal to symbol transformation for two time series data sources

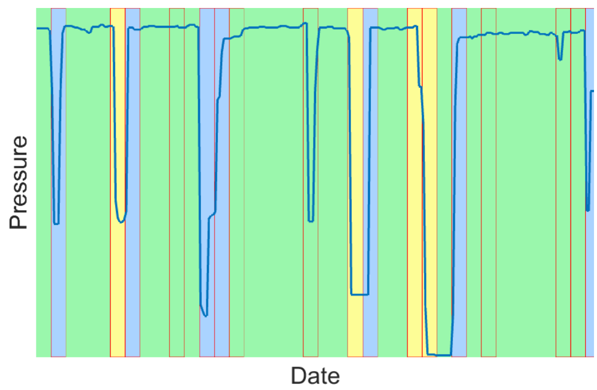

Figure 6: Example of signal to symbol transformation for pressure datastream. (Green - Stable, Blue - Rising, Yellow - Falling, and Red border - Fluctuating)

series data into various timesteps, or segments, based on the information provided by the expert, each data stream/channel timestep is assigned a symbol that is either rising, falling, fluctuating or stable.

Table 4: Description of parameters for quantifying the subtle difference in the trends

Trend	Parameter	Description
Stable	x	The upper limit in variation for a signal to be considered stable
	y	The lower limit in variation for a signal to be considered stable
Rising	x	The period for the rise to occur over
	y	The minimum change in the measurement
	z	Two values relating to the spread of the x and y parameters
Falling	x	The period for the fall to occur over
	y	The minimum change in the measurement
	z	Two values relating to the spread of the x and y parameters
Fluctuating	x	The upper limit for the transition between stable to fluctuating
	y	The lower limit for the transition between stable to fluctuating

Algorithm 1: **Signal to symbol transformation. Where x, y and z are defined in Figure 4**

if *50% of data (< x*mean(data) or > y*mean(data))* **then**
  Result: Stable
else if *50% of data (> x*mean(data) or < y*mean(data))* **then**
  Result: Fluctuating
else
  Calculate average of first and last 10% of data for x period of time;
  if *First < y*Last* **then**
    Result: Rising
  else if *First > y*Last* **then**
    Result: Falling

The assigning of the symbols is performed using a technique based on Signal to Symbol transformation [11] which has been successfully used for rotating plant in the nuclear industry previously [4]. For this application, the symbols are assessed by first calculating the average of the first and last 10% of the timestep, a comparison is then performed to determine which of the following four categories best describes the timestep. The categories are defined as: **Stable** less than 50% of the data is out with the thresholds set by x and y, **Fluctuating** more than 50% of the data is out with the thresholds set by x and y, **Rising** the mean value for the first 10% of the data is greater than y times the mean value of the last 10%, and **Falling** the mean value for the last 10% of the data is greater than y times the mean value of the first 10%, where x, y and z are defined in Figure 4. Algorithm 1 shows the pseudocode for this calculation and an example is shown in Figure 6 for a generic pressure datastream.

Having formalised the rules and implemented the Signal to Symbol Transformation as described above, it is possible to detect faults in near-real-time across multiple data sources. As new timesteps are input into the system each datastream can be assigned a symbol. When all datastreams have been assigned a symbol the expert system can then compare the symbols with the rule base to determine if any fault has occurred. If a positive correlation occurs this timestep is marked with the corresponding fault type. Over time it is possible to build up a history of any faults that have occurred historically in the asset, an example of this over a small period is shown in Table 5.

Table 5: Example processing of 4 datastreams for 5 timesteps

Datastream	T1	T2	T3	T4
A	/	\	/	~
B	\	/	\	\
C	~	–	–	/
D	–	–	–	~
Fault	N/A	7	6	N/A

4. Automated Symbolic Capture of Knowledge

In the previous section, a new symbolic based knowledge elicitation methodology was proposed to capture, formalise, and implement a knowledge-based diagnostic system. While an improvement in the time taken to perform and knowledge elicitation session this approach still requires significant input from a domain expert to capture the expert knowledge. The rest of this section attempts to alleviate this issue and proposes an automated knowledge capture approach. This captured knowledge would form the knowledge base in a typical expert system architecture (Figure 1).

The proposed methodology is shown in Figure 7 and is a three-stage process. Initially, all the condition monitoring datastreams are segmented into individual time segments and each of these segments are assigned a symbol using a signal to symbol transformation. For each symbol key metrics are calculated that aid in distinguishing between the same symbol. Producing a symbolic representation of that data reduces the length of the data in the time domain but increases the

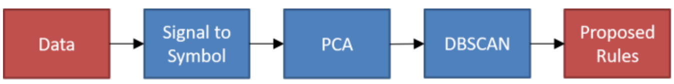

Figure 7: Proposed automated knowledge extraction methodology

dimensionality of the data by the number of parameters calculated for each symbol. This increased dimensionality presents not only a problem from a processing standpoint but also an explainability perspective. Therefore, to overcome this a dimensionality reduction technique, principal component analysis (PCA), was implemented so the data could be visualised and more easily processed. To automatically process and cluster the data density-based spatial clustering of applications with noise (DBSCAN) was used to group repeating patterns that relate to features, or potential faults, within the data. From these clusters, it is possible to automatically generate a rule and hence populate a knowledge base that can be implemented into an expert system architecture.

4.1. Signal to Symbol Transformation

The first stage of the process is to produce a symbolic representation of all the data using a signal to symbol transformation (SST) [4]. This works by segmenting the data at discrete time intervals then for each time interval one of four symbols is assigned. In the previous section, the four symbols were defined in Figure 3 and the various parameters that require to be calculated are defined in Figure 4. An algorithm was developed to complete this process automatically where the output for each time segment is four associated parameters, see Table 6, which accurately represent the data in that time segment.

Table 6: Description of signal to symbol transformation parameters

Parameter	Options
1	Stable, Rising, or Falling
2	Positive float for Rising, Positive float for Falling, or 0 for Stable
3	Fluctuating or N/A
4	Positive integer, or N/A

4.2. Principal Component Analysis

While producing this symbolic representation of the data reduces the overall length of the data with respect to time, it increases the dimensionality four-fold due to the four symbols produced for each time segment. Processing and visualisation of this processed data becomes complicated due to the increased dimensionality; therefore, a dimensionality reduction technique must be implemented. Doing this allows for the domain experts to visualise the symbolic representation of their data, but also allows for the data to be more easily clustered to produce the rules for the diagnostic system. Principal Component Analysis (PCA) [15] is a well-known and used dimensionality reduction technique to increase the interpretability of multivariate data while minimizing information loss. By selecting the first two principal components of the data, which covers a large percentage of the variation within the data, it is possible to visualise the data on a standard 2D plot.

4.3. Density-based spatial clustering of applications with noise

Following this, the next stage takes the dimensionality reduced symbolic representation and clusters the repeating patterns within this data using the Density-based spatial clustering of applications with noise (DBSCAN) [7] algorithm. Each of these clusters will relate to instances in the data where there is a matching pattern across all datastreams that has occurred throughout the data. These clusters could relate to faults that occur on the asset but also may highlight repeating features that occur during normal operational behaviour. By highlighting these clusters on a 2D plot that can be easily displayed to the end-user, the expert can have the final say on what is included in the knowledge base of the expert system.

5. Case Study: AGR Boiler Feed Pumps – Manual Approach

Following the proposed methodology for capturing expert knowledge manually, a case study was performed for data gathered from boiler feed pumps of an advanced gas-cooled (AGR) reactor in the UK. This case study was selected as these assets are critical to the continued operation and electrical generation of an AGR power station, therefore, it is imperative that the pumps are monitored for any abnormal behaviour that may contribute to accelerated plant degradation or to tripping the plant which would result in reduced or zero power generation. The diagnostics rules for the asset were supplied by the domain experts at the beginning of the project. This determined each data stream necessary to diagnose a given predefined list of faults, These rules were represented by a set of trends, i.e. stable, fluctuating, rising or falling. The data contained 37 faults and the associated trends for 10 specific data streams covering pressure, temperature, speed, vibration, and flow. The additional data required to formalise this knowledge into the rules for a rule-based expert system were acquired through knowledge elicitation meetings following the proposed methodology.

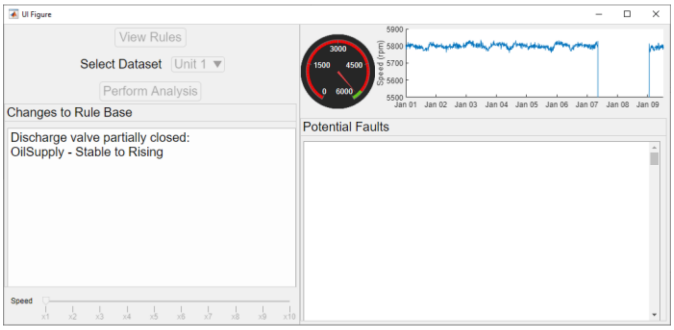

Figure 8: Main GUI for automated boiler feed pump diagnostics

Having captured and formalised the domain experts knowledge it was possible to develop a prototype demonstrator for quickly and accurately identifying faults in the boiler feed pump data in real-time. To implement the knowledge base, all the knowledge that was acquired from the knowledge elicitation meetings is stored in a Microsoft Excel spreadsheet. It was stored in this format so that any engineers using the system would easily be able to view all the captured knowledge and therefore provides greater acceptance of the system and also that the captured knowledge is correct. If the analyst wishes to amend a specific rule or add a new fault type, this can be done by editing the spreadsheet directly. Any updates that are made to the rule-based are automatically detected by the system, and displayed to the analyst in the "Changes to Rule Base" panel in formatted text, see Figure 8. Currently, any amendments made to this file will only be saved for the same session, however, the functionality to load the rule base from historical sessions can be added in the future. This functionality will also require for validation of any new, or amended rules using historical data to ensure that the quality of the knowledge base is maintained. When the analyst is satisfied with the knowledge stored in the rule base the analyst can begin to perform the analysis. The average analysis for the current rule base (37 faults and 10 datastreams) takes less than

0.5 seconds to complete one timestep of the analysis. When potential faults are detected, they are displayed to the analyst in the "Potential Faults" panel, see Figure 8. The date of any fault detected is displayed and the analyst can select this to open up a new window (Figure 9). This presents the analyst with a drop-down menu that contains all faults detected and the associated data streams used in the analysis. The analyst can then view each of these datastreams to display the data covering the time in question on the axis to the right of this window and confirm that the correct fault has been identified.

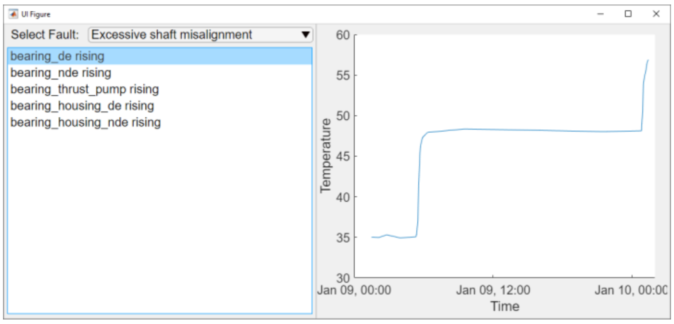

Figure 9: Fault justification window for boiler feed pump diagnostics

This methodology has allowed for the rapid development of a rule-based expert system for fault detection in boiler feed pumps. Due to the novel approach adopted for the knowledge elicitation process, it was possible to minimise the amount of time required from the domain experts but still accurately elicit all the knowledge necessary to develop the system.

6. Case Study: Tennessee Eastman Process – Automated Approach

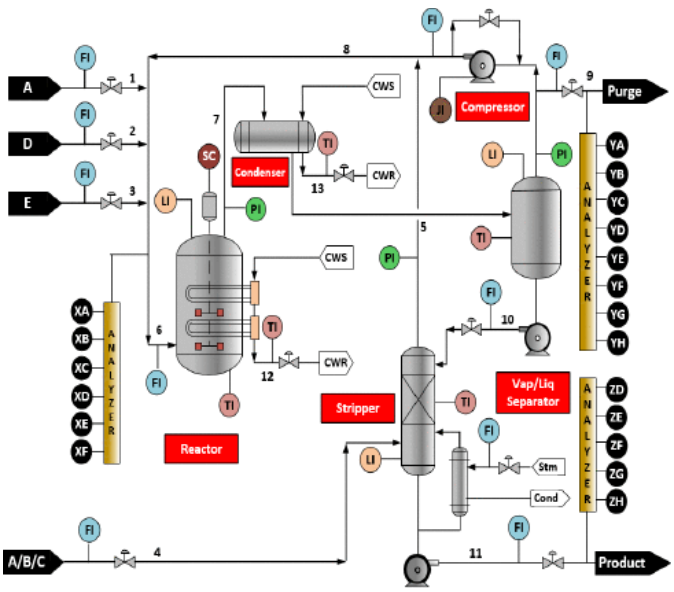

Figure 10: Schematic diagram for the Tennessee Eastman Process [13]

The second case study compares the time taken to develop and accuracy of a system developed using the methodology proposed to capture expert knowledge manually in Section 3 against the automated approach proposed in Section 4 and is discussed in more detail in [20]. This comparison is achieved using the publicly available Tennessee Eastman Process (TEP) Dataset [1]. This is a real industrial process that was modelled computationally in 1993 by Downs and Vogel [6]. The dataset has been widely utilised in studies on fault identification and diagnosis. For fault detection, [21] employs a case-based reasoning technique, whereas [18] proposes support vector machines for fault diagnosis in chemical plants. Figure 10 depicts the TEP's schematic, which includes five primary units: the condenser, compressor, reactor, vapour/liquid separator, and product stripper.

TEP_FaultFree_Training and *TEP_Faulty_Training* are the two major training datasets in the TEP simulation dataset. There are 500 simulations with 500 samples each in the fault-free dataset, all of which are examples of normal operation. The faulty dataset is made up of 500 simulations of 20 different defects, each with 500 samples. The data was sampled every 3 minutes. Both datasets were then integrated, with the defective data being injected at predetermined intervals into the fault-free dataset to form a dataset that includes nearly 40 years of condition monitoring data. This data was divided into two parts: 20% for training and 80% for testing.

A symbolic representation of the training dataset was created using the mentioned signal to symbol transformation approach. The four parameters shown in Figure 4 were generated for each time segment. For a domain expert, this simplified dataset makes detecting trends in the data simpler; but, without intensive knowledge elicitation sessions, it would be exceedingly difficult for someone unfamiliar with the asset to reach the same conclusions. PCA was used to decrease the dimensionality of the data to simplify the 204-dimension symbolic representation (52 datastreams x 4 parameters). Figure 11 shows a plot of the first two principal components having performed this operation. It may be concluded from this that the data is divided into three primary clusters, with the premise that the larger cluster represents normal behaviour or fewer distinguishable defects, while the two smaller clusters indicate two fault situations. DBSCAN was used to automate the process of labelling these clusters, with the results displayed in Figure 11. At this point, the user's only manual input is to choose the labels that would reflect the diagnostic system's recommended rules.

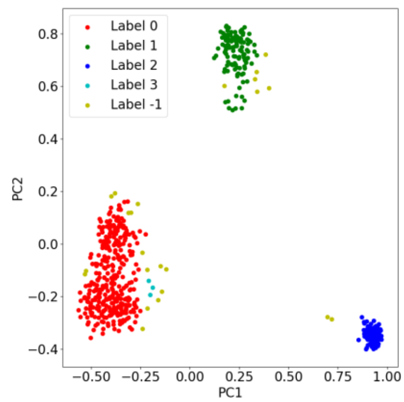

Figure 11: Plot of first two principal components of the signal to symbol transformation output. Labelled using DBSCAN

The two potential rules to be incorporated into the expert system's knowledge base were labelled one and two. Each datastream's symbolic representation was compared for each of the two clusters. Any uncorrelated representations for specific datastreams were deleted and considered irrelevant states, and a rule was created to categorise each point within that cluster for the remaining data streams. Table 7 displays all of the knowledge that was retrieved. The entire process of creating the two rules and inserting them into the knowledge base took less than 30 seconds.

Table 7: Automatically extracted knowledge formulated into rules. U is a rising trend, D is a falling trend, and F is a fluctuating trend.

Datastream	Fault A	Fault B
xmeas_3	U, 0.02	D, 0.03
xmeas_4	U, 0.01	
xmeas_7	F, 39	U, 0.10
xmeas_10	U, 0.98	D, 0.41
xmeas_11	U, 0.01	D, 0.09
xmeas_13		U, 0.10
xmeas_16	F, 32	U, 0.10
xmeas_18	D, 0.02	U, 0.02
xmeas_19	D, 0.46	U, 0.97
xmeas_20		D, 0.09
xmeas_22	U, 0.02	D, 0.04
xmeas_23		D, 0.28
xmeas_25		U, 0.32
xmeas_28	D, 0.63	D, 0.09
xmeas_29		D, 0.49
xmeas_30	F, 18	D, 0.01
xmeas_31		U, 0.56
xmeas_34	D, 0.61	D, 0.13
xmeas_35	U, 0.01	D, 0.35
xmeas_36	U, 0.02	D, 0.43
xmeas_38		U, 0.19
xmeas_39	D, 0.30	

The approach proposed in Section 3 was used for comparison. This was accomplished by personally inspecting 10 instances of each datastream for defects. A manual interpretation was used to determine what each of the four parameters should be for just the related patterns in the dataset. Table 8 displays the results. Due to the manual nature of this technique, it took approximately 8 hours to collect the data for two rules and apply them.

Table 8: Manually extracted knowledge formulated into rules. U is a rising trend, D is a falling trend, and F is a fluctuating trend.

Datastream	Fault A	Fault B
xmeas_3	U, 0.02	D, 0.03
xmeas_4	U, 0.01	
xmeas_7		U, 0.01
xmeas_10	U, 0.98	D, 0.34
xmeas_11		D, 0.08
xmeas_13		U, 0.10
xmeas_16	F, 32	U, 0.10
xmeas_18	D, 0.02	U, 0.02
xmeas_19	D, 0.40	U, 0.96
xmeas_20		D, 0.09
xmeas_22	U, 0.02	D, 0.04
xmeas_23		D, 0.28
xmeas_24		D, 0.01
xmeas_25		U, 0.32
xmeas_28	D, 0.62	D, 0.08
xmeas_29		D, 0.49
xmeas_30		D, 0.01
xmeas_31		U, 0.56
xmeas_34	D, 0.60	D, 0.12
xmeas_35	U, 0.01	D, 0.35
xmeas_36	U, 0.02	D, 0.42
xmeas_38		U, 0.19
xmeas_39	D, 0.25	

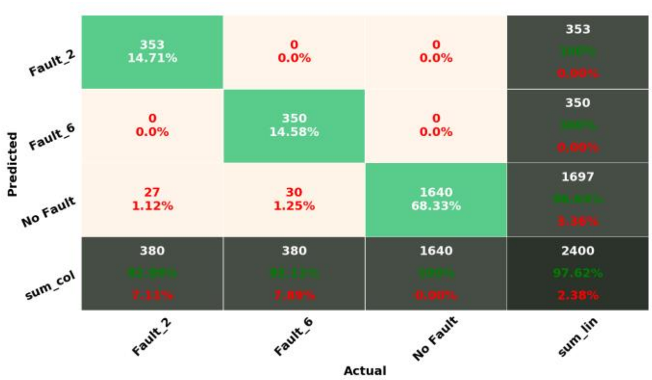

Figure 12: Confusion matrix for knowledge automatically extracted using 20% of the data.

Both knowledge bases were implemented into an expert system to detect the two defects using the same 80% of data not utilised in the automated approach's training. Fault A was discovered to be connected to Fault 2, and Fault B was found to be related to Fault 6 in the ground truth data after an examination of the information provided. As a result, the ground truth for all other faults was adjusted to normal behaviour to identify any false positives.

The results of the knowledge base created using the automated approach are shown in Figure 12. There were no false positives in any class, and the accuracy for Fault 2 and Fault 6 was 92.89% and 92.11%, respectively. However, this resulted in 27 (7.11%) of Fault 2 faults being classed as No Fault and 30 (7.80%) of Fault 6 faults being classified as No Fault. While this is a reasonable result for many applications, the critical nature of the main application field would make this an issue, and a preferable situation would be to have no false negatives but more false positives.

Figure 13 shows the results of the manual strategy. As predicted, the manual approach provides greater classification accuracy than the automated approach. There were no false positives for either class, similar to the automated technique, and accuracy of 99.74% and 100% for Fault 2 and Fault 6 respectively was attained. Except for one case of Fault 2, the

Figure 13: Confusion matrix for knowledge extracted using the manual approach. Five samples per datastream

manual technique properly classified all defects.

While the manual approach detected the two specified defects better than the automated approach, it did so at a large time cost, with a 960% increase in time (see Table 9) over the manual approach. The time it takes to implement the manual technique for a modest two-fault problem may be acceptable, but for more complicated systems or systems with many more faults or datastreams, the suggested approach would greatly reduce the time it takes to construct an initial system.

Table 9: Comparison of results for automated vs manual method.

Method	Fault 2	Fault 6	Time
Automated	353 (92.89%)	350 (92.11%)	< 30 seconds
Manual	379 (99.74%)	380 (100%)	≈ 8 hours

7. Conclusions & Future Work

Essential assets in critical infrastructures, such as nuclear power production, are constantly monitored to guarantee dependable service delivery by anticipating operating anomalies. Engineers now analyse condition monitoring data manually using a specified diagnostic method; however, the rules employed by the engineers to do this analysis are frequently subjective, making it difficult to incorporate them into a rule-based diagnostic system. The transfer of an engineer's expert knowledge into a format appropriate for encoding into a knowledge-based system requires knowledge elicitation. Existing ways of doing so are exceedingly time-consuming, hence a large amount of research has been done to try to cut down on this. The contributions from this paper are twofold; first, an approach to capture domain expert knowledge using symbolic primitives is proposed, and secondly, an automated approach building on this manual symbolic approach making use of data mining algorithms is proposed.

7.1. Manual Symbolic Capture of Knowledge

This paper first proposed a new approach to knowledge elicitation for the development of a knowledge-based fault detection system, specifically a rule-based expert system. The benefits of knowledge-based systems over data-driven approaches are the increased explainability, however, the increased cost in the development time has been highlighted as a disadvantage. The methodology discussed attempts to reduce the burden placed on the domain experts by streamlining the knowledge elicitation process, the most time-consuming part of developing an expert system. Through the use of symbolic representation of knowledge and the parametrisation of these symbols, it was possible to set out a framework to follow for these streamlined knowledge elicitation sessions.

Using this framework, it was possible to develop a rule-based expert system for boiler feed pumps from an AGR power station in the UK. Having further developed the expert system beyond the knowledge elicitation process it has been possible to implement all 37 faults that occur on the boiler feed pumps for the corresponding 10 datastreams. The resulting system can detect faults in the data in real-time due to the segmentation of timesteps into symbols and the efficient inference engine deployed in CLIPS (C Language Integrated Production System) [16].

7.2. Automated Symbolic Capture of Knowledge

Secondly, a method for automating knowledge extraction from several time series datastreams for the development of a rule-based fault diagnosis expert system was proposed. The results demonstrated that it is feasible to automatically generate a knowledge base utilising a combination of signal to symbol transformation to construct a symbolic representation of the data and clustering methods. While the results for this automated approach were above 90% accuracy for both of the identified defects in the case study given, they still fell short of the accuracy provided by the manual approach. However, the large reduction in implementation time, as well as the elimination of formal knowledge elicitation sessions, which have historically been the major barrier to expert system deployment, offset the small performance decline obtained by the automated technique. Importantly, because all of the knowledge utilised to make any choice is kept in the knowledge base, this technique offers a completely explainable output for why any decision was taken.

7.3. Future Work

Future work will involve the development of a human in the loop system to improve the captured knowledge during the system operation. By initially using the methodology discussed in this paper to set the initial parameters for the knowledge and the formalisation of the rule base it should be possible to develop an active learning system [10] to query the analyst to determine any false positives. These labelled false positives will then be used to amend the current parameters to improve the overall system performance. For the automated approach, to extract more rules that were not captured using the proposed approach, more advanced clustering algorithms or dimensionality reduction techniques would be applied. Additionally, more complex forms of signal to symbol transformations will be investigated to establish new symbols and parameters inside the data to which symbols will need to be mapped. This mapping needs to be identified through data or knowledge-driven methods.

Acknowledgements

This work was funded by the Engineering and Physical Sciences Research Council under grant EP/R004889/1.

8. References

1. Averkiev, S. (2020). Tennessee Eastman Process Simulation Dataset. https://www.kaggle.com/averkij/tennessee-eastman-process-simulation-dataset/metadata.
2. Ayodeji, A., kuo Liu, Y., and Xia, H. (2018). Knowledge base operator support system for nuclear power plant fault diagnosis. Progress in Nuclear Energy, 105:42 – 50.
3. Cooke, N. J. (1994). Varieties of knowledge elicitation techniques. Int. J. Hum.-Comput. Stud., 41(6):801–849.
4. Costello, J. J. A., West, G. M., McArthur, S. D. J., and Campbell, G. (2012). Self-tuning routine alarm analysis of vibration signals in steam turbine generators. IEEE Transactions on Reliability, 61(3):731–740.
5. Cullen, J. and Bryman, A. (1988). The knowledge acquisition bottleneck: time for reassessment? Expert Systems, 5(3):216–225.
6. Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. Computers & chemical engineering, 17(3), 245-255.
7. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).
8. Grosan, C. and Abraham, A. (2011). Rule-Based Expert Systems, pages 149–185. Springer Berlin Heidelberg, Berlin, Heidelberg.
9. Guan, X. and He, J. (2019). Life time extension of turbine rotating components under risk constraints: A state-of-the-art review and case study. International Journal of Fatigue, 129:104799.
10. Holzinger, A. (2016). Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain informatics., 3(2):119–131.
11. Nii, H. P., Feigenbaum, E. A., and Anton, J. J. (1982). Signal to symbol transformation: Hasp/siap case study. AI Magazine, 3(2):23.
12. O'Hagan, A. (2019). Expert knowledge elicitation: Subjective but scientific. The American Statistician, 73(sup1):69–81.
13. Ragab, A., El-koujok, M., Amazouz, M., & Yacout, S. (2017, January). Fault detection and diagnosis in the Tennessee Eastman process using interpretable knowledge discovery. In 2017 Annual Reliability and Maintainability Symposium (RAMS) (pp. 1-7). IEEE.
14. Tang, S., Yuan, S., and Zhu, Y. (2020). Deep learning-based intelligent fault diagnosis methods toward rotating machinery. IEEE Access, 8:9335–9346.
15. Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52.
16. Wygant, R. M. (1989). Clips — a powerful development and delivery expert system tool. Computers & Industrial Engineering, 17(1):546 – 549.
17. Xiao, C., Jin, Y., Liu, J., Zeng, B., and Huang, S. (2018). Optimal expert knowledge elicitation for Bayesian network structure identification. IEEE Transactions on Automation Science and Engineering, 15(3):1163–1177.
18. Yélamos, I., Escudero, G., Graells, M., & Puigjaner, L. (2009). Performance assessment of a novel fault diagnosis system based on support vector machines. Computers & chemical engineering, 33(1), 244-255.
19. Young, A., West, G., Brown, B., Stephen, B., and McArthur, S. (2019). Improved explicability for pump diagnostics in nuclear power plants. 2019 ANS Winter Meeting and Nuclear Technology Expo ; Conference date: 17-11-2019 Through 21-11-2019.
20. Young, A., West, G., Brown, B., Stephen, B., Michie, C., McArthur, S., & Duncan, A. (2021, July). Semi-automated knowledge capture and representation for the development of knowledge based systems. In 12th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies (NPIC&HMIT 2021).
21. Zhao, H., Liu, J., Dong, W., Sun, X., & Ji, Y. (2017). An improved case-based reasoning method and its application on fault diagnosis of Tennessee Eastman process. Neurocomputing, 249, 266-276.