SEMI-AUTOMATED KNOWLEDGE CAPTURE AND REPRESENTATION FOR THE DEVELOPMENT OF KNOWLEDGE BASED SYSTEMS

Young A, West G, Brown B, Stephen B, Michie C, and McArthur S
Institute for Energy and Environment
Department of Electronic and Electrical
University of Strathclyde
204 George Street Glasgow, G1 1XW, UK
andrew.young.101@strath.ac.uk

Duncan A
The Alan Turing Institute
British Library
96 Euston Rd, London NW1 2DB. U.K.

ABSTRACT

Plant fault detection and diagnosis is an increasing operational necessity, especially in the nuclear sector where safety is of the utmost importance. Currently, operators have to manually inspect data acquired across multiple assets using predefined diagnostic processes, placing a high time burden on the analyst. Data-driven approaches to solving this problem can produce accurate results approaching what the analysts can achieve but in a fraction of the time. However, the majority of these techniques are black box in nature and therefore lack the explicability, often required for critical assets in the nuclear industry. 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. However, the knowledge elicitation process places a significant time cost associated with the development of knowledge-based systems.

In this paper, an approach is proposed for the development of knowledge-based systems that allow for accurate knowledge capture and formalisation that forgo formal knowledge elicitation sessions. By firstly producing a symbolic representation of the time-series data, abstracting similar trends to produce a list of potential rules, it was found that there was a significant time saving using this approach without equivalent loss of accuracy. A knowledge-based system developed in this way would allow for accurate and transparent fault diagnosis in any discipline, without placing a huge time burden on domain experts.

Key Words: Fault detection, knowledge based systems, explicable artificial intelligence, domain expertise, condition monitoring.

1. INTRODUCTION

Adoption of data driven fault diagnostics offers the possibility of harnessing condition monitoring system investment to obtain greater plant performance without the associated manual effort of data interpretation. However, the ”black box” nature of some of these approaches limits their practical applicability, particularly in the nuclear section, due to the lack of direct explicability that some of these techniques provide, without otherwise introducing new data-interpretation methods that may be unfamiliar/unintuitive to the user. Knowledge based systems present an alternative opportunity in their approach to this problem by producing, codifying and exploiting the knowledge behind diagnoses as opposed to only producing the diagnostic results. The main drawback of a knowledge-based system approach is the amount of time required to produce the system both in terms of the knowledge engineering but also the domain expert in transferring their knowledge. For the acceptance of this approach, any system must produce results quicker and more accurately than the domain experts currently doing the task manually, but also not impose a significant overhead for the engineers in the design and development of the system.

In this paper, an approach is proposed that can interrogate historical data, propose new diagnostic rules to the domain expert to be labelled and then automatically input them into a knowledge base. This approach embeds the expert in this process and allows them to make the final decision on what knowledge is relevant for inclusion to the knowledge base. It also reduces the time burden placed on the domain experts through the removal of the need for formal knowledge elicitation sessions. The rules that are proposed to the user by this approach are determined by clustering repeating patterns in the data and proposing only rules that best represent a baseline knowledge base. It was found that using this approach, on data gathered from a boiler feedpump in a nuclear power station, it was possible to rapidly produce and deploy a knowledge base that can accurately detect several anomalies in previously unseen data. Due to the sensitive nature of this data, a case study is presented at the end of this paper demonstrating the technique on a publicly available dataset.

2. BACKGROUND

2.1. Expert Systems

The most commonly used knowledge-based approach for fault diagnosis is a rule-based expert system or case-based reasoning [1]. The focus of this paper will be on one of these a rule-based expert system, this consists of five main components (Fig. 1).

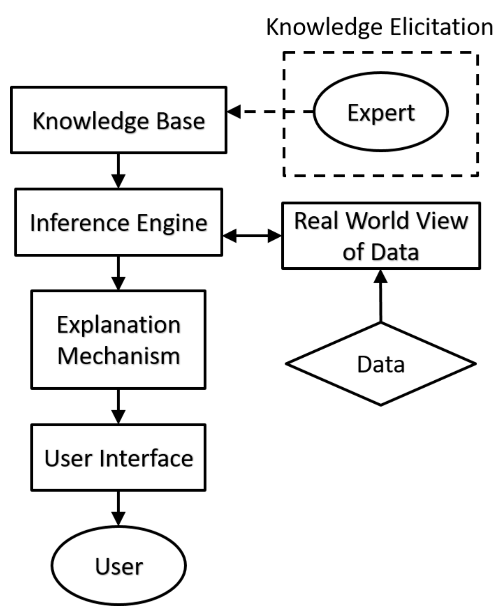


Figure 1. Typical rule-based expert system architecture

2.1.1. Knowledge Base

The knowledge base is one, if not the most import component of an expert system. This contains all the formalised domain-specific knowledge that is elicitated from the domain experts, through a process called knowledge elicitation. There have been many ways to store this knowledge in an expert system, these include **if-then** rules [2], semantic nets [3], predicate logic [4], and petri nets [5]. In a rule-based expert system, the knowledge is typically stored as if-then rules.

2.1.2. Real World View of Data

This component is where all the data relating to the asset under analysis is stored. This can take the form of time-series data, facts relating to the data or asset, or even data relating to intermediate processing steps. For a rule-based expert system, the data stored here is compared with the **if...** conditions stored in the knowledge base.

2.1.3. Inference Engine

The inference engine performs the processing of the data stored in the real world view of data with the knowledge, or rules, stored in the knowledge base. By systematically comparing the data or facts relating to the current asset state, the inference engine will be able to infer if any of the diagnostic rules have been met. This is done primarily through a forward or backward chaining approach. Depending on the structure of the knowledge stored in the knowledge base, it is usually more efficient to choose one of these approaches over the other [6].

2.1.4. Explanation Mechanism

Based on the diagnostic conclusion produced from the inference engine, the explanation mechanism provides explanations and justifications for any decision made [7]. This includes all intermediate steps and facts to produce the conclusion. For any diagnostic system to be accepted by the end-users, an adequate explanation mechanism is crucial.

2.1.5. User Interface

The final component in a typical expert system is the user interface, this provides communication between the end-user and the system. The reasoning and justifications produced from the explanation mechanism for the diagnostic conclusion are displayed in human-readable text [7]. Inputs at this stage can also include additional facts relating to the asset, e.g. unmonitored variables, to further define the diagnostic conclusions. This information can also be passed to external supervisory control and data acquisition (SCADA) systems.

2.1.6. Knowledge Elicitation

While not a component of the operational expert system the knowledge elicitation process is a crucial step in the development of an expert system. This is the process where all the domain-specific knowledge is captured from experts. The two main parties in the knowledge elicitation process are the expert and the knowledge engineer. The expert is the condition monitoring engineer that performs the day to day fault diagnosis manually. From doing this they have acquired years of experience having been through formal training, and having seen multiple instances of the different type of faults that can occur on the asset. The knowledge engineer is the person developing the expert system and is not a domain expert. It is their responsibility to lead the knowledge elicitation session and they can ask the correct questions to elicit all the relevant domain-specific knowledge from the expert.

CommonKADS [8] is a widely used and accepted methodology for performing this knowledge elicitation. It is used to systematically break down all the potential issues arising from the knowledge capture and formalisation phase. This splits up the process into three levels and six main modelling activities. At the context level this is split up into a organisation, task and agent model, and at a concept level split up into a knowledge and communication model. Finally, at the artefact level a design model is produced. Splitting up the process this way allows to better inform the knowledge engineer and to make it easier to develop the structured interview.

Due to the complex nature of most industrial processes these knowledge elicitation sessions can be extremely time-consuming for both parties and the issue surrounding this has been labelled the ”knowledge elicitation bottleneck” [9]. There has been a significant amount of research undertaken to address this issue by proposing alternative knowledge elicitation approaches [10] and [11], however, this is still not considered a solved problem.

3. METHODOLOGY

As previously discussed, the development of knowledge based systems suffers from the knowledge elicitation bottleneck. The approach proposed in the rest of this paper attempts to address the initial knowledge capture and formalization that would form the knowledge base in a typical expert system architecture (Fig. 1). The output from this will be a set of proposed rules to be entered into the knowledge base of a knowledge based system, this must be achieved without the need for extensive user input and must remain true to the original knowledge based system ideology in that the output must be fully explicable.

This is achieved by discretizing each of the condition monitoring datastream available and produce a symbolic representation, with parameters, for each discrete time segment. While this process decreases the length of the data in the time domain by segmenting the data into discrete time intervals, it does, however, increase the dimensionality of the data. This increase in the dimensionality of the data presents a problem both for visualization and for processing, therefore, principal component analysis is used to reduce the dimensionality of the data. Density-based spatial clustering of applications with noise is then used to cluster similar symbolic representations which relate to recurring patterns in the data. These need to be processed manually by the domain expert into specific fault categories, or into irrelevant features of the data. After this selection has been performed the selected rules can be committed to the knowledge base to be used for processing future faults. The full process for the approach is shown in Fig. 2.

Figure 2. Proposed automated knowledge extraction methodology

3.1. Signal to Symbol Transformation

The first step in the approach is to perform signal to symbol transformation (SST) [12] on the raw time-series data for each datastream. This segments the data into discrete time intervals, each time interval is then assigned a symbol based on several calculations. For this application, the symbols selected are shown in Fig. 3. These are the most basic version of all the trends present within each datastream which is common throughout. The subtle differences will be for example a tolerance on how much variation a signal can have before not being considered stable, the period a rise occurs over of the slope, or how noisy a signal has to be to be considered fluctuating. Each of these variations were assigned two parameters (x and y). As a signal can fluctuate while rising or falling this potentially presents a situation where both symbols need to be considered. To that end, for each time segment, four associated parameters are produced that can accurately represent the data, see Table I. The output from this stage for each time interval is a list of parameters relating to the symbolic representation of length $4N$, where N is the number of datastreams available.

Table I. 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

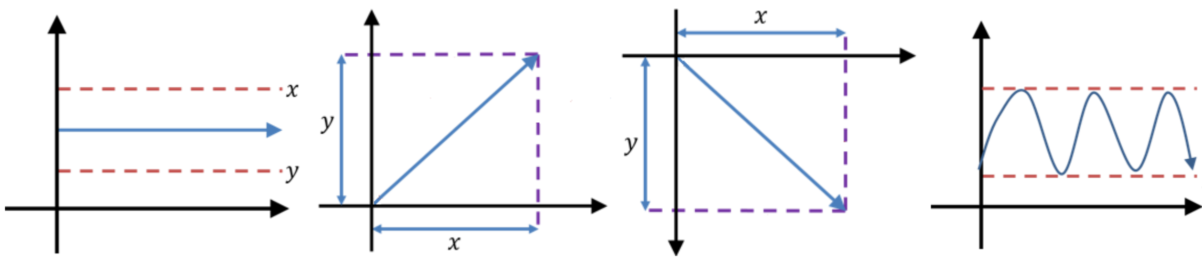


Figure 3. Definition of parameters based on corresponding symbols.

3.2. Principal Component Analysis

Using a signal to symbol transformation algorithm allows for the reduction in the length of the data in the time domain. However, due to the four parameters required to accurately represent the symbols produced this increases the data from 1-D time-series data to 4-D data for each datastream. Because of the dimensionality increase, this makes the data extremely difficult to visualise, hence, a dimensionality reduction technique must be implemented. This will allow for the end-user to visualise the data, but also allows for the selection of clusters within the data that will form the basis for any rules produced by this approach. Principal Component Analysis (PCA) [13] is a well known and used dimensionality reduction technique to increase the interpretability of multivariate data while minimizing information loss.

Using the first two principal components, which will cover a large proportion of the variation within the data, will allow for the visualisation of the symbolic representation for any number of datastreams on a standard 2-D plot.

3.3. Density-based spatial clustering of applications with noise

The next stage in the approach is to automatically cluster the symbolic representations into similar examples of the same type of fault or feature within the data. As the dimensionality of the data is reduced to two dimensions it is possible to use Density-based spatial clustering of applications with noise (DBSCAN) [14] to label the cluster within the data and then visually displays these to the end-user for selection.

3.4. Generation of Rules

Each cluster within the data represents similar symbolic representations, by tracing these back to the original time segments it is possible to produce a diagnostic rule that will detect the data points within this cluster. For each time segment in the cluster, there will exist $4N$ parameters, where N is the number of datastreams. Removal of trends that are not similar across all time segments produces a subset of parameters that directly relates to a diagnostic rule for this feature or fault within the original dataset.

4. CASE STUDY: TENNESSEE EASTMAN PROCESS DATASET

This approach was tested on the publicly available Tennessee Eastman Process (TEP) Dataset [15]. This is a real industrial process that was modelled computationally in 1993 by Downs and Vogel [16]. The dataset has been used extensively in the fault detection and diagnosis research area. In [17] a case-based reasoning approach is used for fault diagnosis, while in [18] support vector machines were proposed for fault diagnosis in chemical plants. Fig. 4 shows the schematic of the TEP containing five major units the condenser, compressor, reactor, vapour/liquid separator and the product stripper.

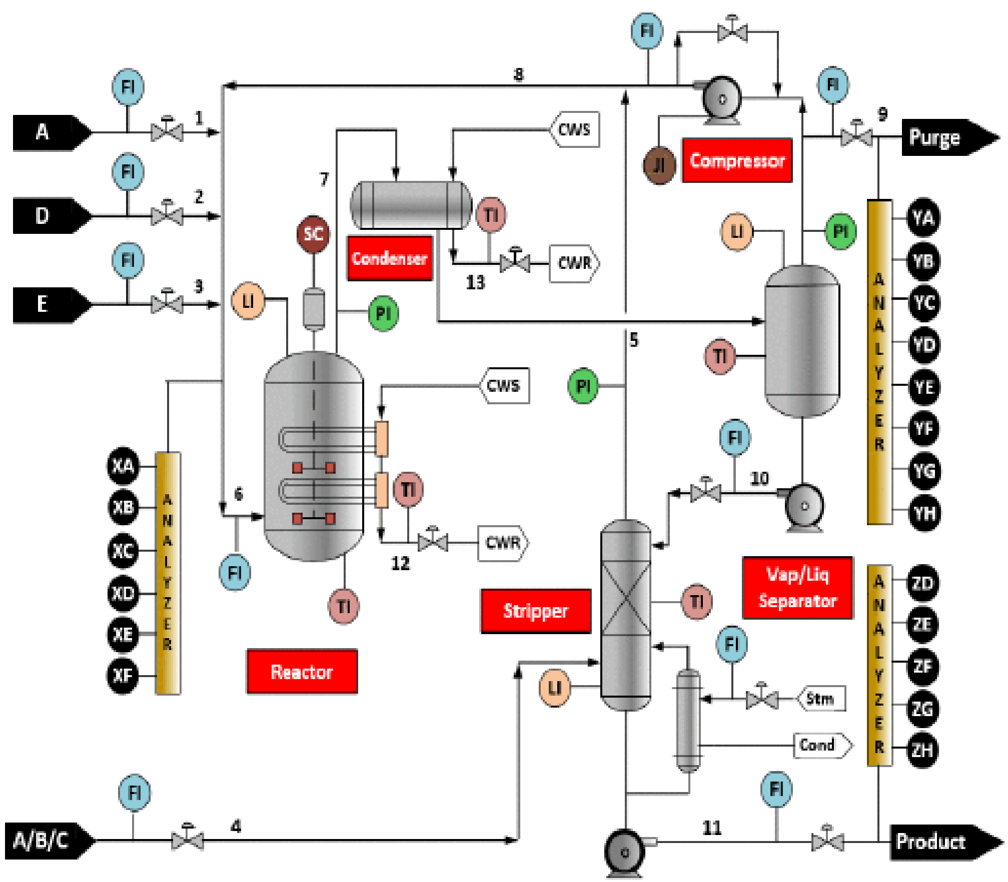


Figure 4. Schematic diagram for the Tennessee Eastman Process [19]

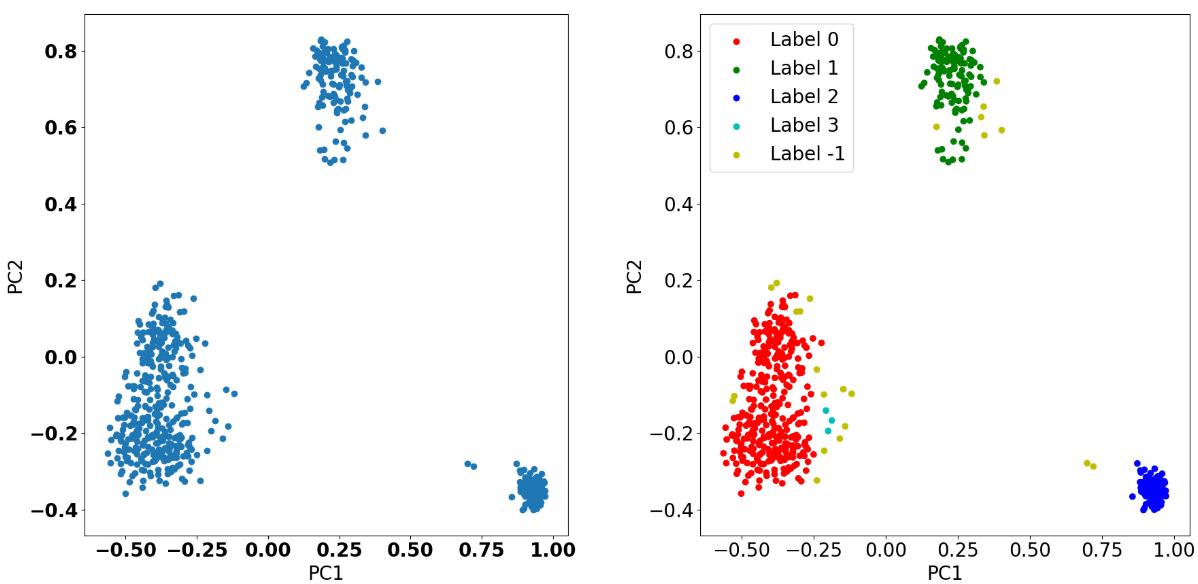
4.1. Dataset Generation

The TEP simulation dataset contains two main training datasets, *TEP_FaultFree_Training* and *TEP_Faulty_Training*. Within the fault-free dataset, there exists 500 simulations each with 500 samples that are all examples of normal operation. The faulty dataset contains 500 simulations of 20 specific faults each with 500 samples. The sampling rate of the data was 3 minutes. Both datasets were then combined injecting the faulty data at set intervals into the fault-free dataset to produce a dataset that includes over 40 years of condition monitoring data. This data was split up into 20% for training and 80% for testing.

4.2. Implementation

4.2.1. Automated Approach

Using the discussed signal to symbol transformation algorithm, a symbolic representation of the training dataset was produced. For each time segment selected the four parameters highlighted in Table I were produced. This simplified dataset allows for easy detection of trends within the data for a domain expert, however, without extensive knowledge elicitation sessions it would be very difficult for someone unfamiliar with the asset to draw these same conclusions. To simplify the 204 dimension symbolic representation (52 datastreams $\times$ 4 parameters), PCA was used to reduce the dimensionality of the data. Fig. 5a shows a plot of the first two principal components having performed this operation. From this, it can be ascertained that there are three main clusters within the data, the assumption being that the largest cluster related to normal behaviour, or less distinct faults and the two smaller clusters represent two fault cases. To automate the process of labelling these clusters DBSCAN was used, the results are shown in Fig. 5b. The only manual input from the user is required at this stage to select the labels that will represent the proposed rules for the diagnostic system. Label one and two were selected as the two potential rules to be inserted into the knowledge base of the expert system. For each of the two clusters, the symbolic representation for



(a) (b)

Figure 5. Plot of first two principal components of the signal to symbol transformation output. (a) Unlabelled, (b) Labelled using DBSCAN

each datastream was compared. Any uncorrelated representation for specific datastreams were removed and considered as irrelevant states, and for the remaining datastreams, a rule was produced that would classify each point within that cluster. All the extracted knowledge is shown in Table II. This entire process took less than 30 seconds to produce the two rules and implement them in the knowledge base.

Table II. 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	Datastream	Fault A	Fault B
xmeas_3	U, 0.02	D, 0.03	**xmeas_23**		D, 0.28
xmeas_4	U, 0.01		**xmeas_25**		U, 0.32
xmeas_7	F, 39.00	U, 0.10	**xmeas_28**	D, 0.63	D, 0.09
xmeas_10	U, 0.98	D, 0.41	**xmeas_29**		D, 0.49
xmeas_11	U, 0.01	D, 0.09	**xmeas_30**	F, 18.00	D, 0.01
xmeas_13		U, 0.10	**xmeas_31**		U, 0.56
xmeas_16	F, 32.00	U, 0.10	**xmeas_34**	D, 0.61	D, 0.13
xmeas_18	D, 0.02	U, 0.02	**xmeas_35**	U, 0.01	D, 0.35
xmeas_19	D, 0.46	U, 0.97	**xmeas_36**	U, 0.02	D, 0.43
xmeas_20		D, 0.09	**xmeas_38**		U, 0.19
xmeas_22	U, 0.02	D, 0.04	**xmeas_39**	D, 0.30	

4.2.2. Manual Approach

For comparison, a more traditional manual knowledge capture approach was performed. This was achieved by manually assessing 10 examples of each datastream for the corresponding faults. Through manual interpretation, it was assessed what each of the 4 parameters should be for only the similar trends in the dataset. The results are shown in Table III. Due to the manual nature of this approach, to collate the data for two rules and implement this it took approximately 8 hours to complete.

4.3. Comparison

Using the same 80% of the data not used in the training of the automated approach both knowledge bases were implemented into an expert system to detect two potential faults. Through analysis of the knowledge produced it was found that Fault A related to Fault 2, and Fault B related to Fault 6 in the ground truth data. Therefore, the ground truth for all other faults was set to normal behaviour, so that any false positives could be detected.

Fig. 6 shows the results for the knowledge base produced using the automatic approach. No false positives were detected for either class and accuracy of 92.89% and 92.11% for Fault 2 and Fault 6 respectively was achieved. However, this related to 27 (7.11%) for Fault 2 and 30 (7.80%) for Fault 6 faults being classified as No Fault. While this is a satisfactory outcome for many applications due to the critical nature of the main application field this would present a problem and a better situation would be to have no false negatives but also have more false positives.

Table III. 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	Datastream	Fault A	Fault B
xmeas_1			**xmeas_23**		D, 0.28
xmeas_3	U, 0.02	D, 0.03	**xmeas_24**		D, 0.01
xmeas_4	U, 0.01		**xmeas_25**		U, 0.32
xmeas_7		U, 0.10	**xmeas_28**	D, 0.62	D, 0.08
xmeas_10	U, 0.98	D, 0.34	**xmeas_29**		D, 0.49
xmeas_11		D, 0.08	**xmeas_30**		D, 0.01
xmeas_13		U, 0.10	**xmeas_31**		U, 0.56
xmeas_16	F, 32.00	U, 0.10	**xmeas_34**	D, 0.60	D, 0.12
xmeas_18	D, 0.02	U, 0.02	**xmeas_35**	U, 0.01	D, 0.35
xmeas_19	D, 0.40	U, 0.96	**xmeas_36**	U, 0.02	D, 0.42
xmeas_20		D, 0.09	**xmeas_38**		U, 0.19
xmeas_22	U, 0.02	D, 0.04	**xmeas_39**	D, 0.25	

The results for the manual approach are shown in Fig. 7 as expected the results for the manual approach provide a better classification accuracy than the automated approach. Like the automated approach, no false positives were detected for either class and accuracy of 99.74% and 100% for Fault 2 and Fault 6 respectively was achieved. The manual approach was able to classify all faults correctly except for one instance of Fault 2.

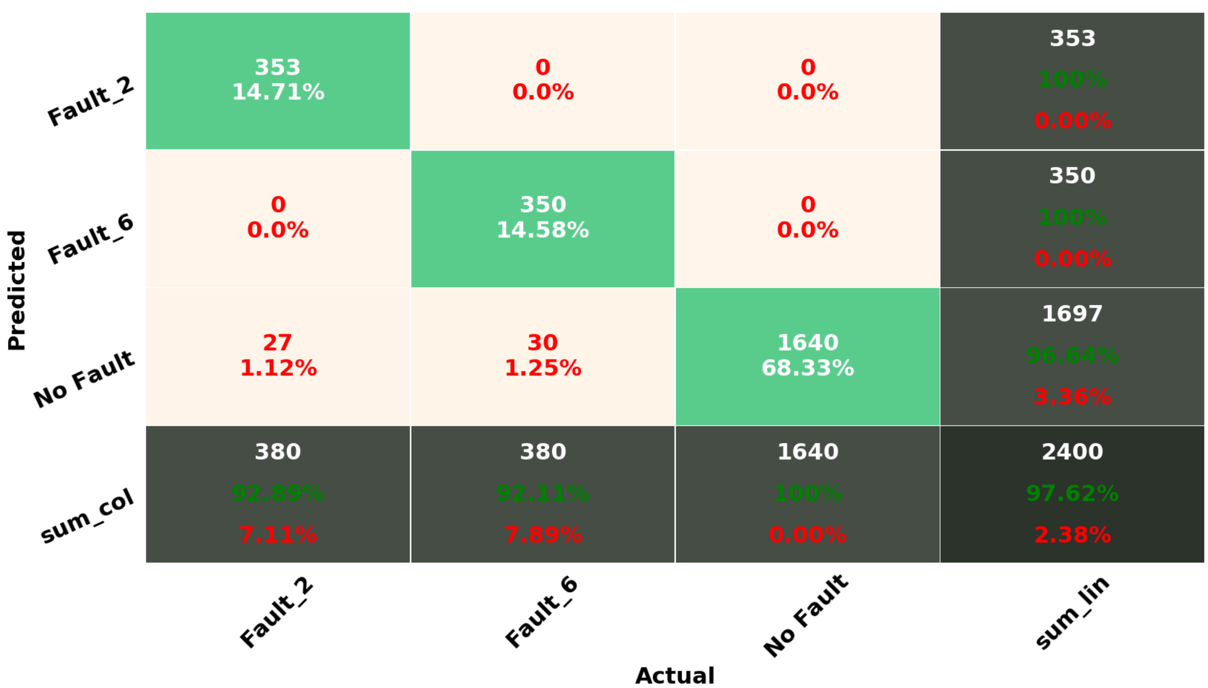


Figure 6. Confusion matrix for knowledge automatically extracted using 20% of the data.

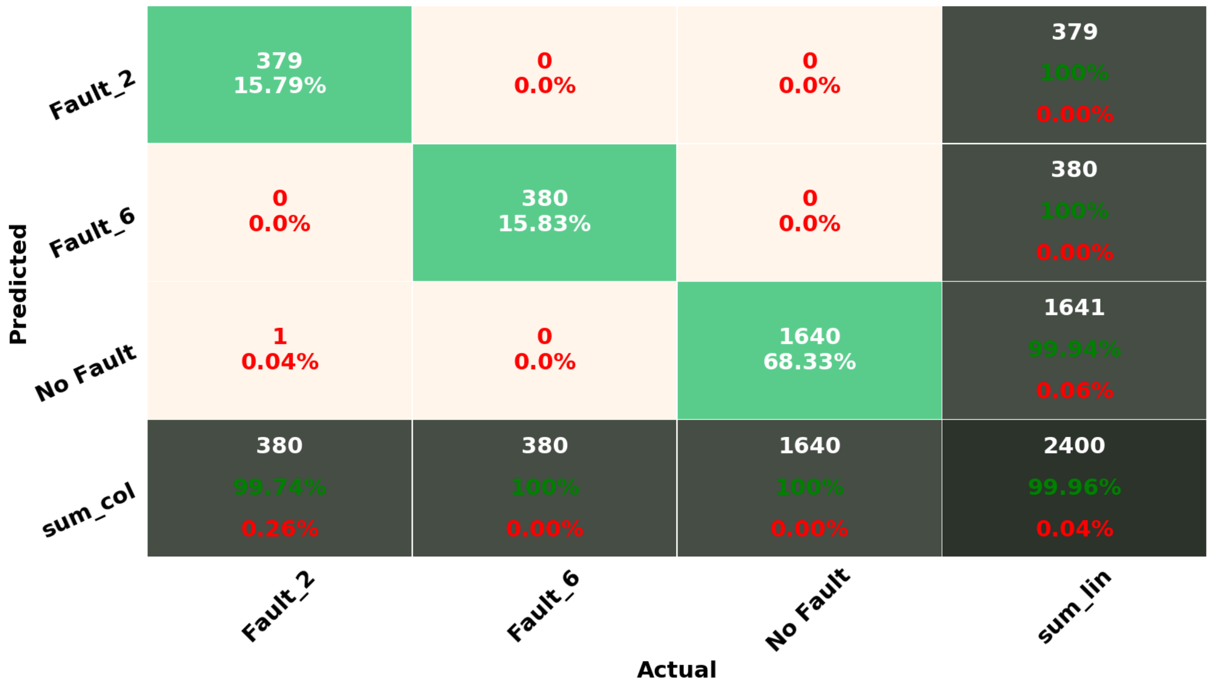


Figure 7. Confusion matrix for knowledge extracted using the manual approach. Five samples per datastream.

While the manual approach performed better than the automated approach at detecting the two specified faults, this did however come with a significant time impact with a 960% increase in time, see Table IV, over the manual approach. For a small two fault problem, the time to implement the manual approach may be acceptable, but for more complex systems or systems involving significantly more faults or datastreams the proposed approach would significantly reduce the time taken to produce an initial system.

Table IV. 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%)	$\approx$ 8 hours

5. CONCLUSIONS

This paper proposed an approach for the automated extraction of knowledge from multiple time series datastreams for the development of a rule based fault diagnostic expert system. From the results, it was shown that it is possible to quickly and accurately produce a knowledge base automatically using a combination of signal to symbol transformation to produce a symbolic representation of the data and clustering algorithms. For the case study presented while the results for this automated approach was above 90% accuracy for both of the selected faults, this still fell short of the accuracy produced from the manual approach. However, the significant decrease in implementation time and also the removal of formal knowledge elicitation sessions that have historically been the main hurdle for expert system deployment, these both outweigh the small drop in performance achieved by the automated approach. Crucially, as all the knowledge used to produce any decision is stored within the knowledge base this approach also produces a fully explicable output for the reason behind why any decision has been made.

Future work will look into techniques of bringing the domain expert back into the loop through an active reinforcement learning type approach to iteratively improve the automatically extracted knowledge to achieve results closer to the performance from the manual approach. More complex clustering algorithms or dimensionality reduction techniques will be used to extract additional rules that were not extracted using the existing approach. In addition to this, exploration of more complex signal to symbol transformation will be used to define additional symbols and parameters within that data.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] I. BECERRA-FERNANDEZ and R. SABHERWAL, *Knowledge management: Systems and processes*, Routledge (2014).

[2] A. F. ALDAOUR and S. S. ABU-NASER, "An Expert System for Diagnosing Tobacco Diseases Using CLIPS," *International Journal of Academic Engineering Research (IJAER)*, **3**, *3*, 12 (2019).

[3] L.-Y. SHUE, C.-W. CHEN, and W. SHIUE, "The development of an ontology-based expert system for corporate financial rating," *Expert Systems with Applications*, **36**, *2, Part 1*, 2130 (2009).

[4] M. Gufar and U. Qamar, "A rule based expert system for syncope prediction," *Proc. 2015 SAI Intelligent Systems Conference (IntelliSys)*, p. 559–564, 2015.

[5] S. J. H. Yang, J. J. P. Tsai, and Chyun-Chyi Chen, "Fuzzy rule base systems verification using high-level Petri nets," *IEEE Transactions on Knowledge and Data Engineering*, **15**, *2*, 457 (2003).

[6] A. AL-AJLAN, "The comparison between forward and backward chaining," *International Journal of Machine Learning and Computing*, **5**, *2*, 106 (2015).

[7] A. VICKERY and H. BROOKS, "Expert systems and their applications in LIS," *Online Review* (1987).

[8] G. SCHREIBER, B. WIELINGA, R. DE HOOG, H. AKKERMANS, and W. VAN DE VELDE, "CommonKADS: A comprehensive methodology for KBS development," *IEEE expert*, **9**, *6*, 28 (1994).

[9] N. SHADBOLT and A. BURTON, "Experiments in Knowledge Elicitation," (1990).

[10] G. D'ANGELO and F. PALMIERI, "Knowledge elicitation based on genetic programming for non destructive testing of critical aerospace systems," *Future Generation Computer Systems*, **102**, 633 (2020).

[11] S. BERNARDI, U. GENTILE, R. NARDONE, and S. MARRONE, "Advancements in knowledge elicitation for computer-based critical systems," 2020.

[12] J. J. A. Costello, G. M. West, S. D. J. McArthur, and G. Campbell, "Self-Tuning Routine Alarm Analysis of Vibration Signals in Steam Turbine Generators," *IEEE Transactions on Reliability*, **61**, *3*, 731 (2012).

[13] S. WOLD, K. ESBENSEN, and P. GELADI, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, **2**, *1-3*, 37 (1987).

[14] M. ESTER et al., “A density-based algorithm for discovering clusters in large spatial databases with noise.,” *Proc. Kdd*, p. 226–231, 1996.

[15] S. AVERKIEV, “Tennessee Eastman Process Simulation Dataset,” 2020, https://www.kaggle.com/averkij/tennessee-eastman-process-simulation-dataset/metadata.

[16] J. DOWNS and E. VOGEL, “A plant-wide industrial process control problem,” *Computers & Chemical Engineering*, **17**, *3*, 245 (1993), Industrial challenge problems in process control.

[17] H. ZHAO, J. LIU, W. DONG, X. SUN, and Y. JI, “An improved case-based reasoning method and its application on fault diagnosis of Tennessee Eastman process,” *Neurocomputing*, **249**, 266 (2017).

[18] I. YéLAMOS, G. ESCUDERO, M. GRAELLS, and L. PUIGJANER, “Performance assessment of a novel fault diagnosis system based on support vector machines,” *Computers & Chemical Engineering*, **33**, *1*, 244 (2009).

[19] A. RAGAB, M. EL-KOUJOK, M. AMAZOUZ, and S. YACOUT, “Fault detection and diagnosis in the Tennessee Eastman Process using interpretable knowledge discovery,” *Proc. 2017 Annual Reliability and Maintainability Symposium (RAMS)*, p. 1–7, 2017.