Knowledge-Based Systems 150 (2018) 1–13

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

RDR-based knowledge based system to the failure detection in industrial cyber physical systems

Dohyeong Kim[a], Soyeon Caren Han[b], Yingru Lin[c], Byeong Ho Kang[b], Sungyoung Lee[a,*]

[a] Department of Computer Science and Engineering, Kyung Hee University, 1732, Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do 17104, Republic of Korea
[b] School of Information Technologies, University of Sydney, Sydney, NSW, Australia
[c] School of Engineering and ICT, University of Tasmania, Private Bag 87, Hobart, Tas 7001, Australia

A R T I C L E I N F O

Article history:
Received 2 October 2017
Revised 18 December 2017
Accepted 4 February 2018
Available online 24 February 2018

MSC:
00-01
99-00

Keywords:
Alarm network
Sensor data mining
Knowledge-based system
Failure detection
Knowledge engineering and cyber physical system

A B S T R A C T

Cyber Physical System(CPS) allows to collect different sensor and alarm data from large number of facilities in industrial plants. Failure and faulty diagnosis is one of the most complicated and dynamic problems in the industrial plant management since most of failures are extremely ambiguous which needs to be solved based on an expert's experience. This makes the solutions very subjective and requires too much time, efforts and monetary investment. In this paper, we are proposing new failure detection approach with machine learning and human expertise by using alarm data. As the first step of developing this new method, we collected several types of alarm data that detected functional failure in Hyundai Steel factory. We analyzed and processed the alarm data with 35 domain experts. Based on the data, we propose a knowledge based system which is Ripple Down Rule-based. This system acquires knowledge by machine learning which is maintained by human experts. The evaluation results showed that the proposed failure detection framework can reduce the time of human expertise acquisition and the cost of solving over-generalization and over-fitting problems by using machine learning techniques.

1. Introduction

In the process and manufacturing industries, there has been a large push to produce higher quality products, to reduce product rejection rates, and to satisfy increasingly stringent safety and environment regulations. To meet the highest standards, most of modern industrial plants contain large number of facilities interacting with thousands of sensors and control, and those detected sensor data can be managed by Cyber Physical System(CPS) While theses facilities can compensate for many types of disturbances, there are changes in the process which the controllers cannot handle adequately. These changes are called as faults or failures. A single failure in a facility can produce inconsistent outcomes, which can harm the core part of the industrial plant that may cause a critical industrial disaster. Therefore, it is crucial to find and apply the best solution for maintaining facilities and preventing industrial disasters [1]. Failure and fault diagnosis is a key application that improves efficiency and productivity.

The early-stage solution was the regular manual maintenance by human workers but this approach cannot be a perfect solution to prevent most industrial disasters [2]. Firstly, because regular maintenance is not effective for all facilities and secondly, because it is very expensive and time consuming.

The recent trend of industrial plant failure detection applications focuses on two main factors, alarms and human expertise. The CPS collects the status of different types of facilities from the sensors, which are attached, on each facility. For example, Fig. 1 shows the partial architecture of CPS in Hyundai Steel plant. If there is any specific symptom detected by sensors the alarm will be ringed. The collected alarm data is sent to human expert in real time. The human experts have experience of several types of industrial disasters which gave them sufficient knowledge in diagnosing and treating failures. Applying facility sensor network, alarm data and human expertise seems to be a good combination in handling failure but this approach also has two key issues.

Firstly, poor facility alarm and sensor network management may produce alarm flooding, which is the phenomenon of presenting more alarms in a given period of time than a human

* Corresponding author.
E-mail addresses: dhkim@oslab.khu.ac.kr (D. Kim), Caren.Han@sydney.edu.au (S.C. Han), Yingru.Lin@utas.edu.au (Y. Lin), byeong.kang@utas.edu.au (B.H. Kang), sylee@oslab.khu.ac.kr (S. Lee).

https://doi.org/10.1016/j.knosys.2018.02.009
0950-7051/

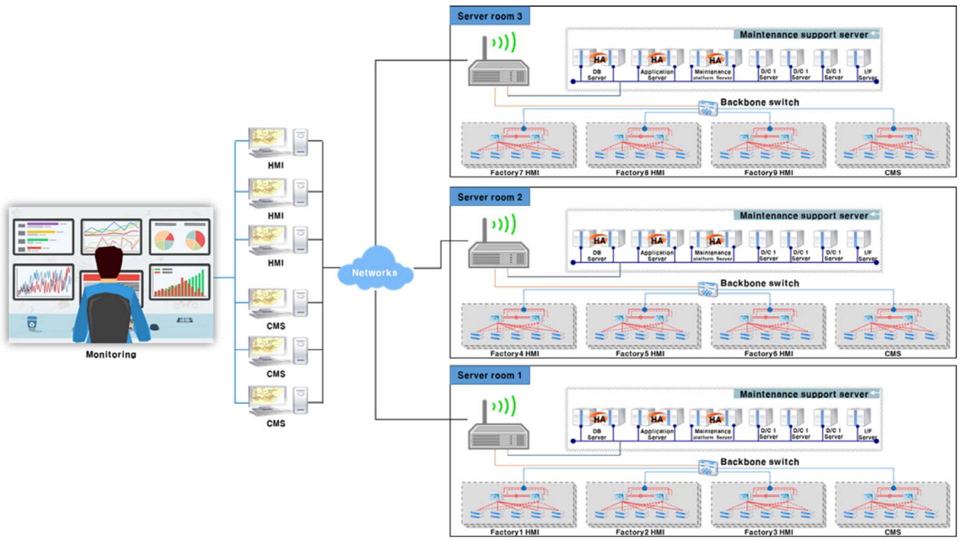

Fig. 1. A partial architecture of cyber physical system in Hyundai Steel plant.

operator can effectively respond. The amount of the collected alarm is too enormous to be properly checked and handled by human experts. Owing to this, some severe failures can be misled or skipped, which may cause a critical industrial disaster. Alarm flooding has been identified as the root cause of significant plant incidents such, as Texaco Pembroke [3] and Three Mile Island Nuclear plant [4]. Several machine learning-based(rule-based or model-based) outlier detection algorithm [5] was proposed in order to reduce human expert effort but it still requires further maintenance with algorithm and human experts.

Secondly, diagnostic and treatment activities are too depended on human experts. There are only limited numbers of human experts who have sufficient experiences in the certain industrial plant. There are two major issues, including availability and lopsided experiential knowledge. Human experts are not always available with every single situation. It can not be always expected any proper treatment if human experts are not available. Additionally, human experts may have lopsided experiential knowledge. Different human experts can diagnose and treat a failure differently. Moreover, some failure cannot be diagnosed or treated since the expert have never experienced before [6].

In order to solve those issues, knowledge based systems are introduced [7] with data mining and human knowledge engineering. The aim of knowledge based system is reasoning and using a knowledge base to solve a complex problems, such as prediction, detection, or recommendation. Most knowledge based systems are constructed by using two different approaches, machine learning technique and human expertise.

For the first solution, machine learning has been applied in order to manage knowledge for detecting failures. Machine learning techniques enable the system to acquire the knowledge from existing alarm data with no help of a domain expert. The techniques are very fast in finding the important pattern and knowledge from the provided data so it reduced the time and cost. However, machine learning has some drawbacks, such as over-generalization and over-fitting [8]. Another solution for failure detection knowledge based system was conducted with human experts. Human domain experts have enough experience so they can save knowledge in order to solve complex problems in a specific domain. However, knowledge acquisition from a human expert is normally in a slow pace. Even if the knowledge was acquired, the acquired expertise tends to be lopsided and would not cover the whole concept of knowledge in the domain since experts acquire domain knowledge based on their past experience [9].

To address this concern, the paper proposes a new industrial plant failure detection approach that is able to leverage the benefits of machine learning and human expertise by using alarm data. In order to achieve this, we firstly collected various types of alarm data that detects a functional failure in Hyundai Steel factory over a one-year period (from September, 2015 to July, 2016). Based on this data, we recruit 35 domain experts in Hyundai Co. and ask them to select the feature and label the class for the training dataset. The training dataset acquires failure detection knowledge from machine learning and human experts by using Rippledown Rules (RDR) based knowledge based system. The proposed approach generates knowledge through machine learning known as InductRDR and enables the maintenance of knowledge to be ascertained through human experts.

The contribution of this research can be summarised as follows:

- The paper proposes an innovative approach to data-aided industrial failure diagnosis by using machine learning for the knowledge acquisition phase of a knowledge based system and human expertise for the knowledge maintenance phase.
- For failure detection in CPS applied large industrial plants, many studies have been conducted with using simple outlier detection, basic data-based machine-learning techniques, or human experts monitoring. The proposed approach produces the following benefits: (1) machine-learning generated knowledge base that removes the knowledge bottleneck and (2) the human expertise maintenance that enables for incremental learning and solves over-generalisation and over-fitting issue.

This paper is organized as follows: In the next section, we provide the related works and discussion, followed by failure detection knowledge-based system in section “Data Collection”. In section “Failure Detection Framework”, we describe the experiments of failure detection and proved the novelty of the proposed methodology. Finally, we conclude the paper in section “Evaluation”.

2. Related works

2.1. Failure detection

Sensor networks and alarm has been used in order to detect and predict functional failure in the large industrial plant. Traditional Alarm analysis system aim to find root causes of a failure in the industrial process by analysing the real-time alarm [10]. The most popular approach in failure detection is the simple outlier detection from alarm and sensor data. The detected outliers, unusual patterns in the alarm or sensor data, do not always end up with the severe issue or failure, as well as to require human expertise in order to define the type of failure. Hence, it is almost impossible for human expert to diagnose the failure and provide the appropriate solution in a short period of time [1,8,11].

Foong et al. [2] aims to prioritize the alarms during alarm floods which would ease the burden of operators with meaningless or false alarms by using fuzzy logic and 125 fuzzy rules. To facilitate in rule construction, five linguistic values are used to determine the ranges of the criticality for each parameter which are lowlow, low, normal, high and highhigh. These ranges of values are gathered from oil refinery engineers or experts. For the output, four different categories of alarm prioritization are used which are (1) normal, (2) low, (3) high and (4) emergency. Nan et al. [12] proposes a knowledge-based fault diagnosis method, which uses the valuable knowledge from the experts and operators, as well as real time data from a variety of sensors. The Methods used were Fuzzy logic and five output functions. Abele et al. [13] developed an alarm system that performs Root Cause Analysis (RCA) upon an alarm model constructed with Bayesian networks. In the paper, methods are presented to construct Bayesian networks for RCA(Root Cause Analysis) with a knowledge-based and a machine learning approach.

Aizpurua et al. [14] aims to build a rule based expert system used to find the Alarm Root Cause. The system finds the root cause of avalanches of alarms and their effects and reduce their number through grouping or clustering techniques, complying with the Engineering Equipment and Material Users' Association (EEMUA) 191 standards. Zhao et al. [15] proposed a power system alarm processing and fault diagnosis expert system (AFDES). In the proposed expert system, Backus–Naur Form (BNF) is used to design a kind of expert rule frame which operator can write and add the rules with his own defining language to rule-base. Ebersbach and Peng [16] developed a first artificially intelligent system for fault diagnosis and machine condition monitoring using integrated analysis of vibration, oil and wear debris analysis technique. It designed and implemented an expert system for analyse vibration data with similar accuracy as an expert maintenance engineer in an automated software package allowing high analysis throughput, and hence suitable for commercial condition monitoring laboratories or on-site use. Safavian and Landgrebe [17] aims to reduce the number of alerts presented to the operator. It used a rule-based method. 6 knowledge bases are built, and the rules describe typical interrelations between alarm messages which have a common cause. The concept has been implemented in a software prototype which manages the alarm log, plant model and interrelation rules and presents the grouped alarms in an interactive alarm display. The alarms are not deleted from the alarm logs. Rather, it is the same alarm log but structured hierarchically. The result is a compact alarm display with fewer alarm messages visible on the top level but a higher information density. The application of the approach on two case studies resulted in a successful reduction of alarms. Folmer and Vogel-Heuser [18] presented an overview of an algorithm for the automatic alarm data analyzer (AADA). It is able to find possible and significant reasons for alarm floods by identifying the most frequent alarms and those causal alarms consolidating alarm-sequences. 12.000.000 alarms are used as a dataset. For the experiment, the alarm logs have been available from four different industrial process control and manufacturing plants as case studies, e.g. purification plant (continuous process), hydraulic fiber press (discrete and continuous (hybrid) process) and incineration plant (continuous). They found that this data with AADA can reduce alarm floods and operator workload by improving and redesigning AMS (alarm management).

Ahmed et al. [19] proposed an alarm system framework with various types of alarm data management system, including data filtering system, alarm delay, and alarm deadline settings. Izadi et al. [20] described and evaluated the most efficient alarm filtering system. They presented the alarm filtering approach that calculates the similarity in the alarm and sensor data sequence, and clusters them in each group. The research conducted by Zhu et al. [21] was focused on finding an alarm flooding management system. They proposed a dynamic alarm management approach by applying the Bayesian Network technique.

The above researches aimed at analysing the characteristics of alarm data, and managing the size of an alarm. This would limit the participation of domain experts, and most of the researches are not evaluated to check whether the detected alarm or sensor data pattern affects the real failure.

2.1.1. State-of-the-art on knowledge engineering techniques for failure diagnosis

Several machine learning and data mining techniques were applied for industrial failure diagnosis. Yin et al. [22] introduced machine learning-based online fault diagnosis by using incremental support vector data description (ISVDD) and extreme learning machine with incremental output structure (IOELM). An online fault diagnosis approach combining ISVDD and IOELM could detect new failure mode and recognise fault based on learning knowledge of the diagnosis system.

Extreme learning machine(ELM)-based real-time fault diagnostic system for gas turbine generator systems was proposed [23], and compared with the most successful machine learning algorithm, including support vector machine. The evaluation result is 98.22% accuracy in 2.7 ms. The proposed ELM fault diagnostic framework is generic, it could be applied to the other applications of condition monitoring in which the fault identification time is critical.

Wind turbine failure diagnosis system applied a binary tree SVM and a self-organising feature map neural network [24]. Fuzzy logic, support vector machine (SVM) and artificial neural networks were employed for continuous monitoring and fault diagnosis for monoblock centrifugal pump [25]. Feature extraction using wavelets and SVM algorithm for classification are successful approaches for practical applications in industrial fault diagnosis.

Li and Zhao [26] proposed gravitational search algorithm (GSA) to identify and diagnose new fault samples by calculating the weighted kernel distance between them and the fault cluster centers. The proposed method has been applied in unknown fault diagnosis, and evaluation results have shown the effectiveness of the proposed method in achieving expected diagnosis accuracy for both known and unknown faults of rotatory bearing. The application of evolving fuzzy modeling to fault-tolerant control was proposed in two steps: fault detection by applying model-based approaches and fault accommodation by using fuzzy models [27].

Despite machine-learning based framework studies in failure diagnosis, it was not successfully adopted in the real industrial application. This is because machine learning is too dependent on the quality of the training data. If the size or range of the training data was insufficient and unrepresentative of the industrial domain.

The following sections will describe two major approaches, human expert and machine learning techniques, in acquiring and managing knowledge.

2.1.2. Single classification ripple down rules (SCRDR)

In the knowledge engineering field, Ripple Down Rules(RDR) [9] is regarded as one of the best knowledge acquisition method for expert systems. RDR is able to reduce the knowledge acquisition bottleneck issue [28] and also enables

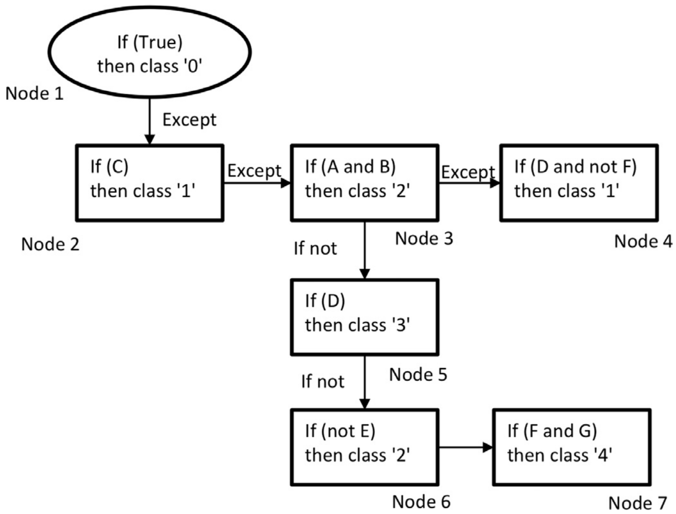

Fig. 2. An example of SCRDR knowledge tree.

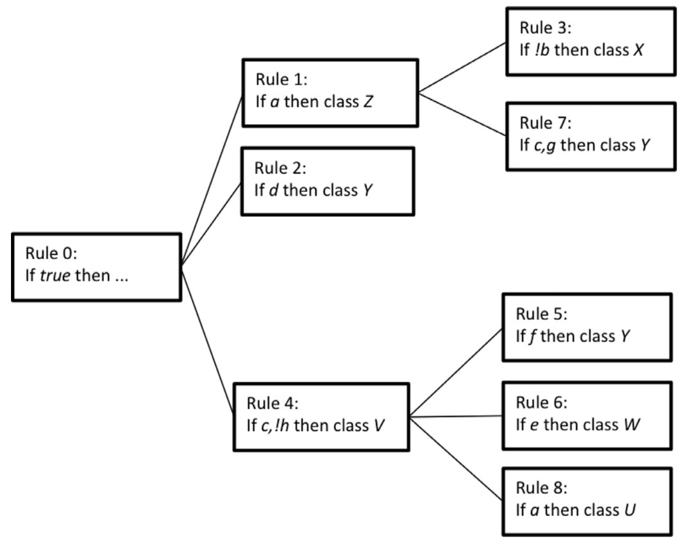

Fig. 3. An example of MCRDR knowledge tree.

resolving the verification process when domain users handle the validation themselves.

SCRDR stands for Single Classification RDR. The example of SCRDR knowledge tree can be found in the Fig. 2. According to [29], the SCRDR structure is a finite binary tree where each node can have two distinct branches, which are called except and if not. Examples are measured from the root node of the SCRDR tree. Each node in the tree is a rule with the form of if α then β (α is the condition and β is the conclusion). If an example satisfies the condition α it is passed to the next node of the except branch. Otherwise, the example is passed to the next node following the if not branch. If an example satisfies α but the node does not have the except branch, β of this node is the conclusion for the example. If an example does not satisfy α but the node does not have the if not branch, β of the last node on the path where the example satisfies its α is the conclusion for the example. In order to ensure that a conclusion is always returned, examples always satisfies the condition of the root node. This node is called the default node and the conclusion is called the default class. For instance, as in Fig. 2, Node 1 is the default node and class ‘0’ is the default class. An example which only satisfies condition A should be passed down through Node 1 and stops at Node 2. Since Node 2 does not have the if not branch, the example is classified as ‘0’ by Node 1. If an example satisfies A, B, C and D, it should be passed down through Node 1, Node 2, Node 3 and stops at Node 4. Since it satisfies the condition of Node 4, it is classified as ‘1’ [30].

When the measure process returns the wrong conclusion for an example, a new node is attached to the last node in the SCRDR tree. If the last node has no except branch, the new node is attached as the except branch, otherwise it is attached as the if not branch. The example which is associated with the new node is called the cornerstone example for that node. The rule generated for the new node entails the features of the cornerstone of the new node but not that of the cornerstone of the last node where the new node is attached. When the measure process returns the wrong conclusion for an example, a new node is attached to the last node in the SCRDR tree. If the last node has no except branch, the new node is attached as the EXCEPT branch, otherwise it is attached as the IF not branch. The example, which is associated with the new node, is called the cornerstone example for that node. The rule generated for the new node entails the features of the cornerstone of the new node but not that of the cornerstone of the last node where the new node is attached [31].

In SCRDR, all rules are constructed in a binary tree. When the system encounters an incorrect classification, a new exception rule is added based on expert judgement. Therefore, SCRDR can incrementally develop a relatively accurate knowledge base, provided the domain is fixed and the experts provide the correct judgements. Since RDR based knowledge base depends on expert judgement, the correctness of the used language expressed by the expert is the key of developing a good knowledge base. According to Pham and Hoffmann [32], it may cost a long time to classify most of the relevant cases correctly, if the target is linear threshold in the numerical input space and an expert is only allowed to use axis-parallel cuts, since it is unsuitable for him to express accurately.

2.1.3. Multiple classification ripple down rules (MCRDR)

Kang el al. [33] introduced Multiple Classification RDR (MCRDR) as an extension of RDR (SCRDR) to improve the limitations of RDR (SCRDR) including reducing the burden of the knowledge acquisition task and preventing knowledge base being ill structured which may result in considerable repetition of knowledge.

Unlike SCRDR, MCRDR evaluates all the rules in the first level of the knowledge base. The rules of the second level are evaluated to refine the rules which are satisfied at the first level. It keeps evaluating the next level in a recursive way until there is no more level to evaluate or none of the rules can be satisfied [34,35]. MCRDR is able to provide multiple conclusions since it constructs rules with multiple paths. Each path is a particular refinement sequence. Knowledge is acquired from the experts when an example is classified incorrectly or needs to be given a new classification. The process can be described in the following three steps. (1) The expert provides correct classifications for the examples of the system, (2) The system decides on the location for the new rules, and (3) New rules are provided to the system by the expert and added to the knowledge base for correction.

The expert selects valid conditions from the current example to acquire the new rule for a given classification. The rule that has been created is then compared with the cornerstone cases of each node. If any cornerstone cases of a node satisfy this new rule the expert needs to select extra conditions for differentiating the current case and the cornerstone cases. For example, in Fig. 3 when a case only satisfies conditions a and c but its correct class should be W, the system may decide the new rule location is on either Rule 3 or Rule 8.

Since cornerstone cases of Rule 3 and Rule 8 are found to satisfy the conditions, experts are required to provide extra rules to cause those cornerstone cases no longer to satisfy the set of conditions. The system then repeats this process until no remaining cornerstone cases satisfy the rule and it may simply add a new classification which is not in the tree. There are three ways of correcting the knowledge base [33]:

1. Add a stopping rule at the end of a path to prevent the wrong classification
2. Add a rule at the end of a path to give the new classification.
3. Add a rule at a higher level to give the new classification.

As can be seen in Fig. 3, the system should add a new rule at the end of the path (Rule 3 or Rule 8) to give a new classification.

MCRDR concerns multiple independent classifications, whereas it maintains the advantages and principles of SCRDR. Like SCRDR, MCRDR is also based on the premise that a justification experts provide is necessary for a correction of knowledge in a particular context. However, the context in MCRDR is maintained in a different way and only consists of rules that have been satisfied by the data. Besides the validation of MCRDR includes differentiating the new example from a range of different examples.

A class in a MCRDR tree is the set of separated rule paths which provides the same conclusion. For example, in Fig. 3, Class Y has three rule paths: Rule (0, 1, 7), Rule (0, 2) and Rule (0, 4, 5). Therefore, a rule path consists of all conditions of all previous rule nodes and conditions of the last node which concludes the class. Han et al. [36] mentions during the process of building MCRDR structures, the relationship between different classes are untouched and invisible from users. However, this information is able to provide inspiration for users to capture the point of rule creation and help users to realise how relationship may change the meaning or importance in the domain. Although MCRDR can work very effectively in many domains, similar implicit information contained within the structure itself is still not being extracted or exploited simply.

2.2. Knowledge management by machine learning

As mentioned in the previous section, knowledge acquisition is traditionally conducted with human domain expert and knowledge engineers. However, there are two major issues in acquiring knowledge from domain experts: first, knowledge acquisition from human experts is normally in a slow pace; secondly, an expert cannot cover whole concept of knowledge in a specific domain. Because of those issues, it is almost impossible to manage the demand of expanding knowledge since a successful knowledge base may require an extremely large number of concepts and rules.

Machine learning techniques received lots of attention since those can learn and acquire concept and knowledge from the existing data with no domain experts help in a short period time [37]. The most common machine learning techniques for knowledge discovery are neural network and decision tree.

Neural network models the human brain and consists of a number of artificial neurons and connections. Cascading chains of decision units with neurons used to recognize non-linear and complex functions. Knowledge can be acquired based on the input data incrementally so it does not need to be re-programmed. Only training phase is required in order to maintain the knowledge base. Time-series alarm data for failure detection was applied with neural network learning model so the model achieved a highly successful rate even though it has some noisy and outlier data. Tjhai et al. [38] focused on filtering alarms using the combination of neural network and k-means clustering.

Decision Tree algorithm is the typical machine learning approach that builds the knowledge base with the interpretable tree-structured rules. The Nearest Neighbor model is assigned to the most common class among the data samples that are most similar to the newly presented data. The approach has been used with small size of alarm data because of its extremely large distance calculation time. Both decision tree and nearest neighbor algorithm have been used a lot in the sensor failure detection and prediction since it is easy to understand the consequences, which can trace the result. Anuar et al. [39] designed and implemented a failure detection system using a decision tree learning approach, which is fast and easy to interpret.

However, those machine-learning techniques have over-generalization and over-fitting issues if the size or range of data is not sufficient to cover the knowledge in the specific domain.

In order to solve this issue, Gains [31] introduced Ripple Down Rule (RDR) based machine-learning technique, called InductRDR. The purpose of InductRDR is combining the concept of knowledge creation through machine-learning technique and knowledge acquisition from human domain experts. Gains described a sequence of dispersing knowledge partially from the view of a human expert, which consists of the following seven stages: Minimal Rules, Adequate Rules, Critical Cases, Source of Cases, Irrelevant Attributes, Incorrect Decisions, and Irrelevant Attributes & Incorrect Decisions [40].

The first stage is a complete, minimal set of correct decision rules so no data is required for knowledge acquisition since the correct answer is available from the expert. On the contrary, the last stage is a source of data from which the correct answer might be derived with the greatest probability of correct decisions so the expert has provided little. The stages in the middle from top to bottom show a decrease in existing knowledge though human intervention but an increase in new expertise through machine learning [31].

The main use of existing RDR is close to the top stage. Therefore, Induct RDR that derives rules directly from an extension of Cendrowskas Prism algorithm was made to be close to the bottom [41]. This Induct RDR sums standard binomial distribution as the possibility of selecting the correct data at random to measure the correctness of a rule.

In Fig. 4, given a universe of entities E, a target predicate Q and a set of possible test predicates of the form S on entities in E, use them to construct a set of rules from which the target predicate may be inferred given the values of the test predicates. The probability of selecting s and getting c or more correct at random is the sum of the standard binomial distribution.

In supervised learning, there is a risk of over-fitting the noise by memorizing the peculiarities of the training data [42]. Pruning approaches are commonly applied to solve the problem. Although Induct RDR recognizes the importance of pruning, it only removes redundant clauses and compresses the structure to some extent. Reducing over-fitting and improving generalization prediction capability has not been considered [31]. Ripple-Down Rules classifier (Ridor) is an implementation of Induct RDR in Weka. It first creates the default rule. The exceptions are created for the default rule with the lowest (weighted) error rate [43]. Different from the original Induct RDR, Ridor applies information gain to evaluate each rule and it prunes a rule by reducing error pruning.

The original InductRDR [40] has a serious problem in handling the big data since it checks all the rules from all the combinations of conditions. In terms of big data, it would be almost impossible to check all million or billion data record one by one. Hence, the proposed model should consider time-efficiency. Moreover, the original InductRDR has an issue in analysing and handling the numeric data so it tends to produce higher performance with only nominal data type. Therefore, the method we are proposing in this paper should consider the improvement in running time efficiency and numeric data type handling.

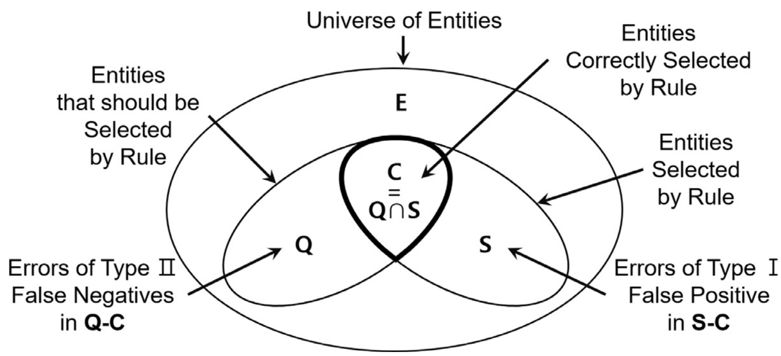

Fig. 4. Problem of empirical induction [31].

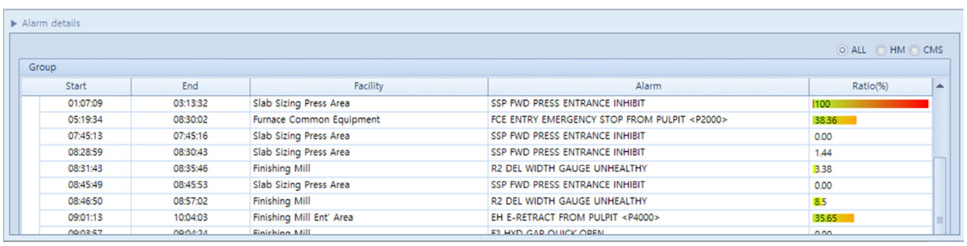

Fig. 5. Alarm data collection interface.

3. Alarm data collection and processing

As mentioned in the introduction section, the paper focuses on detecting failure status by using alarm data particularly on large industrial plants. This section describes how we collected and analyzed a set of alarm data, and classified the facility status based on the data.

3.1. Alarm data collection

We use the alarm data that was collected in the industrial plant from the Hyundai Steel Co., Ltd for a 1-year period (from September, 2015 to July, 2016). In the plant, most alarms are connected with one or more sensors to indicate the facility activities, and an alerting device to detect any failure. We collected over a round half million, 567,748, alarm data from an industrial plant. Fig. 5 shows the alarm data collection interface with the detailed information of some example alarm data that was collected on 25th of July 2016.

As can be seen on the right side of Fig. 5, the alarm data are produced from the two difference sources, HMI (Human Machine Interface) and CMS (Central Management System). We merged alarm data from those two sources and visualized this to the interface as can be seen in the upper-right corner of the Fig. 5. When new alarm is occurred, the alarm integration system collects all detailed information of the specific alarm, including starting time, ending time, facility id, alarm message, and ratio(%). The ratio(%) describes how much capacity that alarm took so it uses 100% of working memory at that period. For example, the first alarm data in the gure indicates the 'forward press entrance inhibits' issue in the Slab Sizing Press (SSP) Area, which occurred from 01:07am to 03:13am. The last column 'Ratio(%)' describes how much capacity that alarm took so it uses 100% of working memory at that period.

3.2. Facility failure status assessment

Next, we focus on assessing the facility status. The collected alarm data is traditionally sent to the human experts, and experts diagnose and treat the failures. However, the aim of this research is proposing an automatic failure detection framework by using machine learning and human expertise. Therefore, it is crucial to have classes/labels for defining the status of facilities based on alarm data. For this task, we asked 35 human experts, who have experienced various types of industrial disasters from the industrial plant of Hyundai steel Co., Ltd since they have sufficient knowledge in diagnosing and treating failures by reading and analyzing the alarm data. Based on the focus group with 35 domain experts, we identified 48 different facility statuses that would be used as class labels. The following are 48 facility statuses identified by domain experts: normal, apc, breakaway, bur, bwd, carbonization, close, collision, corrosion, cradle, cut, damage, defective, division, down, flame, gap, hunting, impact, intrude, leak, nocooling, noenter, nolink, nooff, nooperation, noreversefwd, norupture, nosense, nostop, obstacle, on, open, permeate, plateloadon, position, relaxation, slip, slowincome, speed, stop, transform, trip, up, vibration, wronginputpower, wrongoperation, wrongsense. Based on the given 48 class attributes, we asked 35 experts to classify and label the status of facility by reading and analyzing the provided alarm data. The labelling procedure is as follows. A class label for an alarm data was assigned if 21 out of 35 experts (60% of experts) agreed with the label. In another case, we selected the first and second-rated label, and asked experts to choose one of the labels. For example, let's label the first alarm data with 35 experts. 40% of 35 experts labelled it with 'noenter' class and 35% of experts classified it into 'obstacle' class. In this case, we asked experts to classify the data by picking 'noenter' or 'obstacle' class. With this procedure, the alarm data is labeled into 'noenter' class.

3.3. Feature analysis for failure status detection

We propose a set of features to characterize alarm data in our collections. 35 domain experts de-fined the features. These include some hardware features specific to the Hyundai Co. Ltd industrial plant but most are quite generic so can be applied to other plant environments. We identify three types of features depending on their scope: hardware-based features, time-based features, and size-based features.

1. **Hardware-based features** consider the individual hardware type in the industrial plant. It includes each alarm id and facil-

Table 1
Examples of alarm training data.

Alarm ID	Time	Facility ID	Count	Lifetime	Ratio	Status
DRV_183	17	H1103364	1	3228	896.67	INTRUDE
ES_041	16	H1101349	10	112	31.11	HUNTING
MCC_323	23	H1103364	1	3600	1000	IMPACT
APC_014	8	H1101349	4	22	6.11	BUR
PAG_004	1	H1101613	13	43	11.94	LEAK
PRC_090	9	H1101349	4	21	5.83	CARBONIZATION
PRC_058	7	H1105709	1	30	8.33	NORMAL
PRC_071	22	H1102579	1	82	22.78	NO LINK
GRS_008	10	H1105709	1	20	5.56	NO REVERSE
PRC_020	7	H1101613	1	4	1.11	CUT
...	...	...	...	...	...	...

ity id. Alarm id represents the message type that was produced in the alarm data. The facility id shows the identifier for each facility in the industrial plant.

2. **Time-based features** consider the characteristics of time factor for each alarm data. It contains time and lifetime of alarm data. Time feature represents the starting time (e.g. 17 means 5pm) of the specific alarm. The lifetime describes the length of time that the specific alarm is alive in one hour. The length would be described as millisecond.
3. **Size-based features** consider the size of each alarm data. It includes occurrence and the capacity of the specific alarm. Count feature shows the occurrence of the alarm data in one hour. Ratio feature represents the percentage of resources taken by the specific alarm.

Based on those three features, we produced 6 individual features (alarm_id, facility_id, time, lifetime, count, ratio) into the training dataset for knowledge acquisition through machine learning technique. Some examples of alarm training data are shown in Table 1. In the table, we demonstrated first 10 alarm data in the training dataset. Each row represents 6 different conditions attributes and its failure status of a specific alarm.

4. Failure detection framework

The goal of this research is to propose new failure detection framework for industrial plant by using alarm data and RDR knowledge based system. The proposed RDR knowledge-based system for detecting failure allows acquiring knowledge by applying machine learning technique and maintaining them by domain experts who have experience in detecting failure from large industrial plants. The proposed framework can be described in the diagram. We would like to briefly introduce the proposed failure detection framework before describing the detailed process. The proposed framework can be seen in Fig. 6.

We would like to briefly introduce the proposed failure detection framework before describing the detailed process. First, we built a training dataset with 6 features/attributes and a 'status' class as described in the previous section. Then, using the training dataset, we built a supervised classifier by using RDR-based machine learning, Induct RDR. InductRDR adopts knowledge acquisition approach of the traditional machine learning techniques, which allows creating a knowledge base from the structured training dataset, but produces the rule-based knowledge base in Ripple Down Rule format. Therefore, InductRDR would enable human domain experts to modify the existing knowledge base, which is developed by machine learning technique. For example, if there is any incorrect classified data based on the testing dataset, human experts can add exception rules (either additional or refine rule) where data are incorrectly classified.

Then, it finds out incorrectly classified data based on the given testing dataset. The knowledge based system acquired rules from human experts to add exception rules (additional rule) where data are incorrectly classified.

The following sections, "Knowledge Acquisition by RDR-based Machine Learning" and "Human Knowledge Acquisition using RDR Framework", include the detailed process of knowledge acquisition with InductRDR and knowledge maintenance with human experts. Note that we have updated several functionalities of original InductRDR [31] in order to achieve better performance with the large size of real-time alarm data.

4.1. Knowledge acquisition by RDR-based machine learning

First, we would like to discuss how it became possible to build the rule-based knowledge system with alarm training dataset by using the updated InductRDR. The basic idea of InductRDR is generating rules in a RDR structure with a rule induction algorithm. A rule at a single node in RDR structure is called as a clause, and it includes one or more terms in a form of attribute-relation-value.

Rule generation process of InductRDR can be described in the following three steps: First, the most frequently occurring class in the training data is selected as the default class value for the root-level rule. Then, it applies standard binomial distribution and searches a class that has the smallest m-value. The selected class is used for splitting the dataset into two subsets: true and false cases. If either of these two subsets has more than one class, the rule will be generated recursively. However, the original InductRDR does not fit into the alarm training data because of its size and complexity.

We propose three core updates in the original InductRDR. First, we updated the clause selection mechanism. The original InductRDR searches all possible combinations of terms in order to find the best class. In this process, m-value is an indicator that shows the quality of a term. Only appropriate terms are added to the clause until it only selects true positive data. Unfortunately, this process would produce severe computational issue if the domain has a large training dataset. In order to solve this issue, the updated InductRDR ordered terms first. Since m-value is used as a quality assessment function for each term and only terms with smallest m-value can be added to the clause, terms can be sorted by m-value in ascending order. The possible best terms will be always combined and assessed at the early stage, and that allows finding the best clause in a short period of time.

Secondly, we modified the approach to evaluate the best clause. The original InductRDR applied m-function, the sum of the standard binomial distribution, for assessing the credibility of the clause [31]. Gains mentioned that m-value would produce the probability that the rule could be good at random, and that it derives no assumptions about sampling distributions. However, the problem would be occurred if the size of the dataset were too large. The m-values for all rules become to 0 with the big size of dataset so it is almost impossible for distinguishing the importance of the rules. This is because the original InductRDR just chose the attribute randomly in this case. In order to remove this random selection, we borrowed the attribute selection approach, information gain, from decision tree learning algorithms since it is the key to improving prediction accuracy in decision tree algorithm [44]. The updated Induct RDR would use information gain for the best clause evaluation when m-value becomes 0. Finally, we adopt the numeric data handling approach in the updated InductRDR. While nominal data has fixed values with specific meaning, numerical data is usually continuous and the meaning is not clear. Nominal data can be divided into groups by their values but it is almost impossible to do the same thing for numeric data. InductRDR uses only inequality signs for best clause selection but it is extremely clumsy with large and complied dataset. Due to the nature of InductRDR, we applied information gain for numeric value handling. With the above updates, we built the system that includes

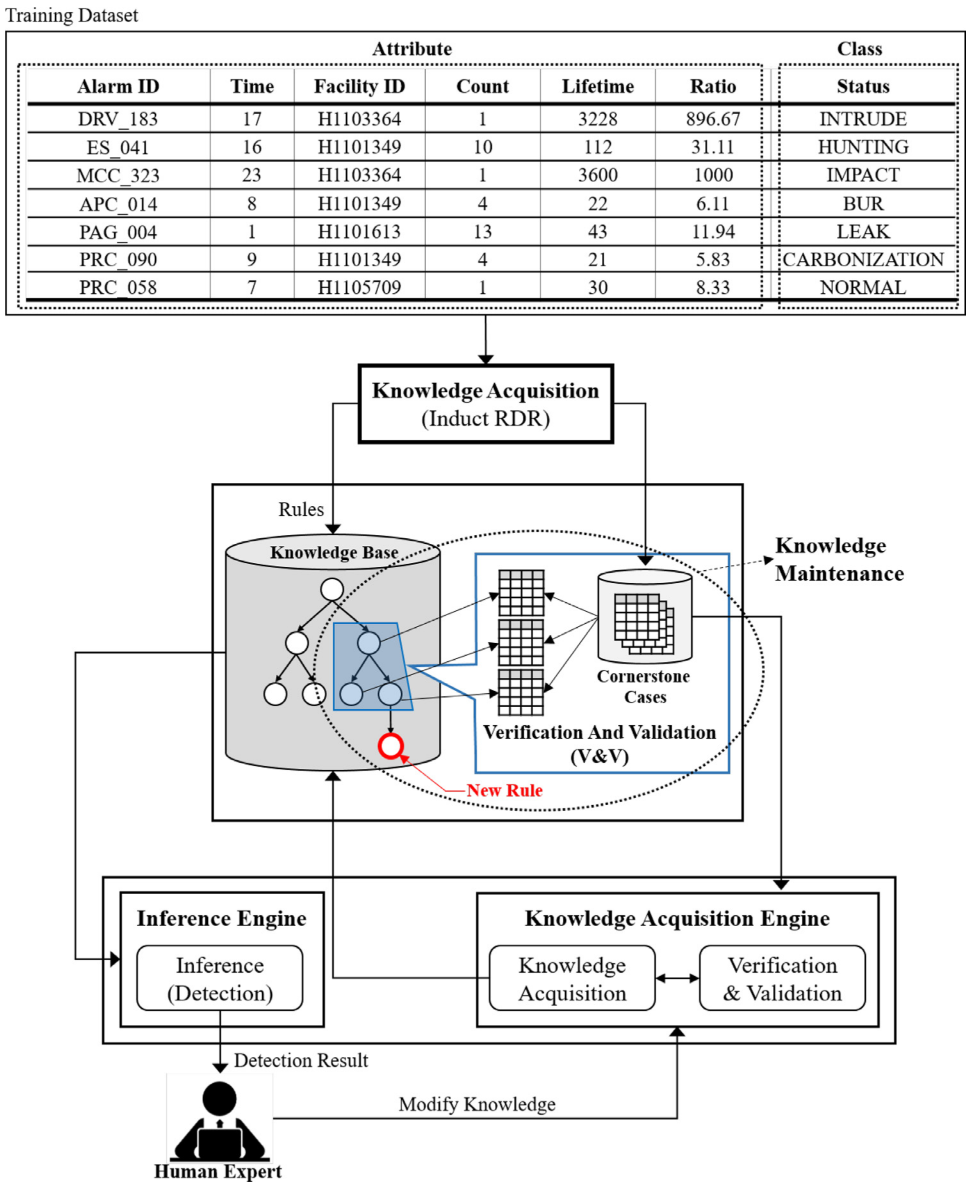

Fig. 6. Failure Detection Framework.

the updated InductRDR for knowledge acquisition, and the interface for knowledge maintenance with human expert. We put the alarm data, which is collected and processed in the section “Data Collection”, into the updated InductRDR, and it produces the RDR-structure rule (knowledge) base. This knowledge base can be seen on the left side of the Fig. 7. It took 3 seconds to build the knowledge base, and has 177 rules in total. The prediction/classification performance of the updated InductRDR would be discussed in the section “Evaluation”.

The right side of Fig. 7 represents the list of conclusion and the case browser. The case browser shows the training dataset; each case is a row of the alarm dataset that we trained with InductRDR. The first row contains the value of 6 attributes (alarm_id, count, facility_id, lifetime, ratio, and time) and a value ’INTRUDE’ as a class ‘status’.

Fig. 8 shows the correctly classified instance in the knowledge base. The value of a class for the second case (CaseID = 2) in the case browser is ‘INTRUDE’ which is predefined. From the knowledge base, the rules contain the condition, which is matched with any value of 6 attributes, would be fired. As seen at the left side of Fig. 8, the rule 1 and 132 are fired by the selected case, case 2. The rule no. 1 should be fired if the time is later than 6am, and the value of time attribute for the case is 17(5pm). The facility status should be classified as ‘TRANSFORM’. However, before concluding this classification, the system checks the current case with the child rule no.132. The rule no.132 contains a condition to check whether the lifetime is over 3519 ms, and the value of lifetime for the case is 3600 ms, which has satisfied the condition. The final conclusion would be ‘INTRUDE’ as there is no more child rule to check. Therefore, the final inference conclusion and the original conclusion are equal, which means ‘correctly classified’.

4.2. Human knowledge acquisition using RDR framework

However, not all cases are correctly classified. As has been mentioned repeatedly in this paper, one of the challenges in machine learning is the fact that not all instances will be classified correctly, a byproduct of issues such as over-fitting and over-generalization. Fig. 9 shows an example, which is incorrectly classified. The proposed RDR framework system supports the function, which enables

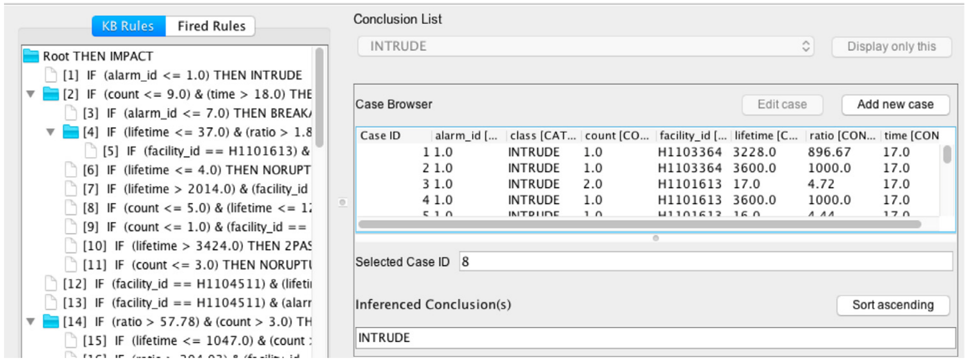

Fig. 7. An example of the generated knowledge base with InductRDR.

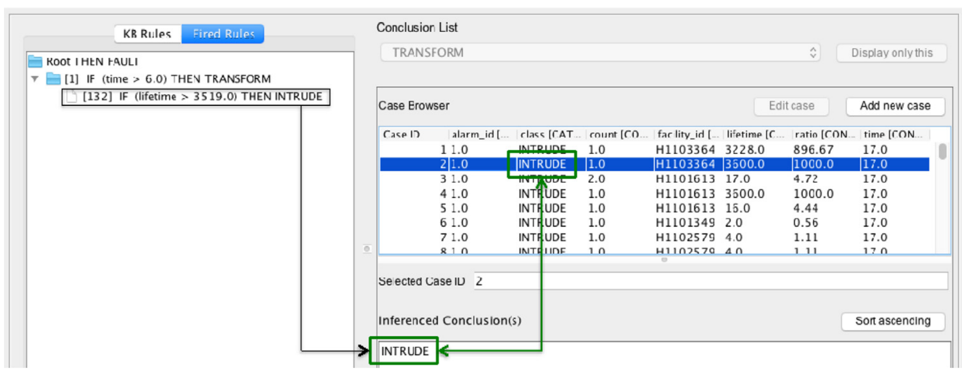

Fig. 8. An example of correctly classified instances.

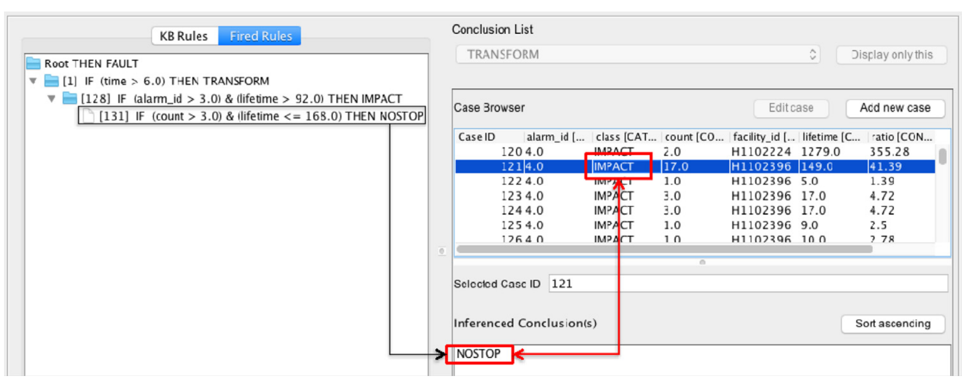

Fig. 9. An example of Incorrectly Classified Instances.

acquiring the human expert's knowledge based on the current context and adding that knowledge incrementally. As can be seen the Fig. 9, the case id 121 produced the 'NOSTOP' status as a conclusion since the rule no.1, 128, and 131 were fired. The last child rule includes the condition to check whether the alarm occurred more than 3 times and the lifetime is equal or shorter than 168 ms. The values of case id 121 are matched to the conditions but the class value 'IMPACT' does not match with the inference conclusion 'NOSTOP'.

In this case, the RDR framework will acquire the rules from human experts for refining the knowledge base where the data is incorrectly classified by adding new rules. Fig. 10 shows the output after the refine rule addition.

The new rule no.178 is added so it is now correctly classified as 'IMPACT'. However, not all of the human knowledge can be applied. There are two reasons summarized as follows. First, there are data, which have the same vector of attributes but belong to different classes.

This is because the existing attributes are not enough to tell the difference. Therefore, the class which the majority belong to will be decided at the conclusion and it is less possible to correctly classify the minority. Secondly, some rules applied might affect other correctly classified data. The knowledge created by the expert gives a hint about how these rules affect the whole dataset. If a rule has more incorrectly classified data than correctly classified data, it should not be applied.

The performance analysis of human rules addition will be conducted in the section "Evaluation".

5. Evaluation

In order to evaluate the performance of the proposed failure detection framework, we use 567,748 alarm data that was collected from a factory of Hyundai Steel Company, and processed by 35 human domain experts, employees in Hyundai Steel Co. The detailed

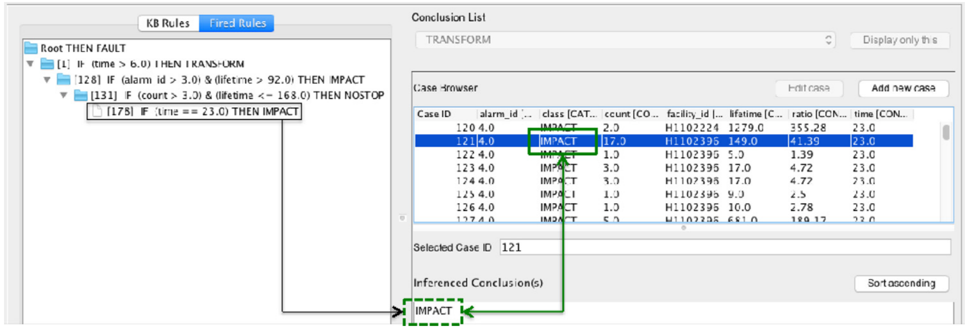

Fig. 10. An example of modified rule.

Table 2
Applied machine learning techniques.

No	Evaluation technique	Base algorithm
1	NaïveBayesSimple [45]	Naïve Bayes
2	MultilayerPerceptron [46]	Neural network
3	LIBSVM [47]	Support vector machine
4	C4.5 Decision tree [48]	Decision tree
5	The modified InductRDR	InductRDR
6	The modified InductRDR + Human RDR rule	InductRDR and human expertise

Table 3
The performance comparison with machine learning techniques and proposed InductRDR with human rules.

Evaluation techniques	Detection accuracy
Neural network	92.31%
The updated Induct RDR	92.05%
The updated Induct RDR with human rules	**100%**

collection and processing procedures are described in the section "Data Collection".

5.1. Failure detection performance evaluation

To conduct the evaluation of our proposed failure detection framework, we compared the performance of the modified InductRDR on the alarm dataset against four common machine learning classifiers, including Naïve Bayes, neural network, decision tree and support vector machine. The algorithmic approach and underlying philosophy of each algorithms are fundamentally different, however, all of them are considered as highly successful techniques in predictive modelling.

We tested the performance with six machine learning techniques by using 10-fold cross validation. The Table 2 describes the algorithms that are applied for the evaluations.

The performance of failure detection with machine learning techniques can be found in the following Fig. 11. In this domain, it shows that Neural Network and InductRDR achieved over 92% detection accuracy. The applied techniques (no.6) builds the initial knowledge base with the modified InductRDR, and then adopts human knowledge. The test dataset is used to examine this knowledge base to find incorrectly classified data. A simulated expert is used to find correct rules for those incorrectly classified data. In the case of InductRDR (machine learning only) and C4.5 Decision Tree, they are based on machine learning only, so their prediction accuracy is based on predicting the test dataset using the knowledge base acquired from the training dataset.

As can be seen in Table 3, it has been found that the updated InductRDR only can achieve 92.05% of prediction accuracy. After adding human rules, the result can be improved up to 100%. However, upon updating the classifier with domain expertise, the prediction accuracy markedly improved, classifying all training instances with 100% accuracy. Therefore, one can surmise that adding human knowledge to the knowledge base generated by the machine learning classifier does improve the classification accuracy and can mitigate some of the pressing concerns of machine learning as we can handle issues of noise or anomalous data to some extent. It is important to note that the 100% accuracy achieved with the incorporation of the human rules applies to the current fixed dataset and we would expect the accuracy to be reduced in a real-world clinical setting. However, the benefit of this approach is the ability to adapt through incremental learning and so the system is able to improve in the real-world settings, even when the performance reduces.

Although Neural Network had the best prediction accuracy (92.31%) among machine learning techniques, the updated InductRDR with human rules outperforms it eventually. Therefore, it can be concluded that adding human knowledge to the knowledge base created by machine learning does improve the prediction accuracy. The prediction accuracy becomes low if there are significantly over-generalization and over-fitting problems. In this case, prediction accuracy has been improved so that it implied that over-generalization and over-fitting problems have been solved to some extent.

In addition to the high performance of failure detection, the proposed approach allows human experts to incrementally add and maintain the knowledge in the knowledge base with no rebuilding or initialization process.

5.2. Failure detection performance at feature level

In this section, we study how specific subsets of features perform in the task of automatic assessment of credibility. To do this, we train the InductRDR (machine-learning) algorithms considering subsets of features. We consider 3 subsets of features grouped as follows: (1) Time-based, (2) Size-based, and (3) Hardware-based. The detailed information of those feature levels can be found from the Section 3.3.

We train the updated InductRDR with each subset feature a training set.

The instances in each group were split using a 10-fold cross validation strategy. In this evaluation, we aggregate all 47 faulty related classes as a "FAULTY" class while labelling 'normal' class as just "NORMAL" class as can be seen in Table 4. The results indicate that among the features, the time-based features and size-based features are very relevant for diagnosing the failure/faulty status. We observe that hardware-based features are not enough by themselves for this task. On the other hand, "NORMAL" class is in general more difficult to detect.

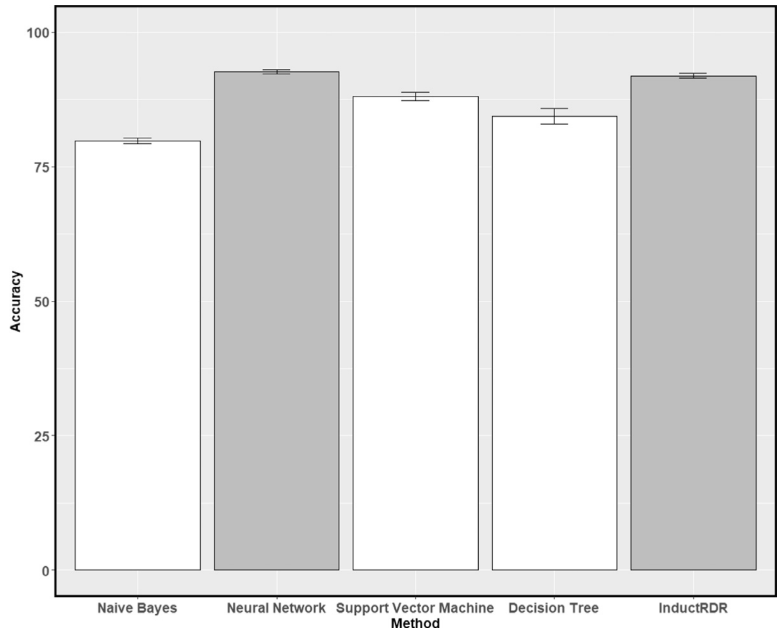

Fig. 11. The accuracy of failure detection with machine learning techniques.

Table 4
Experimental results obtained for the classification of failure detection.

Time-based					
Class	TP Rate	FP Rate	Prec.	Recall	F1
NORMAL	0.623	0.082	0.610	0.623	0.616
FAULTY	0.918	0.377	0.923	0.918	0.921
W.Avg.	0.868	0.327	0.869	0.868	0.869
Size-based					
Class	TP Rate	FP Rate	Prec.	Recall	F1
NORMAL	0.147	0.033	0.478	0.147	0.225
FAULTY	0.967	0.853	0.847	0.967	0.903
W.Avg.	0.828	0.714	0.785	0.828	0.788
Hardware-based					
Class	TP Rate	FP Rate	Prec.	Recall	F1
NORMAL	0.749	0.454	0.643	0.749	0.692
FAULTY	0.546	0.251	0.666	0.546	0.600
W.Avg.	0.652	0.357	0.654	0.652	0.648

5.3. Cost evaluation

In order to solve the core issue of machine learning, over-generalization and over-fitting is traditionally accompanied with inserting new data to the existing dataset to enrich the patterns. In this case, the previously created knowledge base will be removed, and a new knowledge base is constructed. The amount of knowledge can be quantified as the number of nodes and conditions in a knowledge base, so the cost of solving the problems can be quantified as how many nodes and conditions are reconstructed. This is the case of machine learning. In the case of adding human knowledge, the cost is how many nodes and conditions are added to the original knowledge base.

The following Table 5 summarises the result of reconstructed or increased nodes and conditions after solving over-generalization and over-fitting problems. By applying human knowledge, the increased ratio of nodes for improving 1% of accuracy is 28.57%, much smaller than InductRDR only (109.78%). Similarly, the increased ratio of conditions for improving 1% of accuracy is 60.15%, much smaller than InductRDR only (99.66%). As mentioned above, the reason that pure machine learning models cost much is because they remove previous knowledge base and create a new one every single time that it encounters a new data case which cannot be explained by the existing knowledge base.

Therefore, it can be concluded that the reconstructed or increased ratio of the knowledge base is much smaller by combining human knowledge and machine learning than those approaches based on machine learning only.

6. Discussion

Detecting failure status in large industrial plants has been noted as complex and dynamic problem area because of its enormous size of alarm and sensor data, and experiential knowledge requirements. Either machine learning technique or human expert system has been applied to acquire and maintain the knowledge for failure detection but neither did work successfully. In this project, we collected and analyzed the alarm data with 35 domain experts in Hyundai Steel Co., and propose a novel approach that uses Ripple-down Rule (RDR) to maintain the knowledge from human experts with knowledge base generated by the updated Induct RDR. Based on the experiment, we found that it improves accuracy to 100% with the fixed dataset. It is important to note that the 100% accuracy achieved with the incorporation of the human rules applies

Table 5
Cost evaluation result of knowledge increased.

Models	Updated InductRDR (%)	Updated InductRDR with human rules (%)
Increased ratio of nodes	261.54	58.25
Increased ratio of conditions	222.58	124.20
Increased ratio of nodes per 1% of accuracy improvement	109.78	28.57
Increased ratio of conditions per 1% of accuracy improvement	99.66	60.15

to the current fixed dataset and we would expect the accuracy to be reduced in a real-world clinical setting. However, the benefit of this approach is the ability to adapt through incremental learning and so the system is able to improve in the real-world settings, even when the performance reduces.

7. Conclusion

The proposed approach in this paper allows human experts to incrementally add and maintain the knowledge in the knowledge base without having to rebuild or re-initialise the knowledge base, unlike pure machine learning approaches which rebuild the knowledge base from scratch each time. Moreover, the proposed failure detection framework can reduce the time of human expertise acquisition and the cost of solving over-generalization and over-fitting problems in machine learning technique. The proposed failure detection framework has never been reported previously. Moreover, this framework can be successful detection approach in the domain if it requires handling big size of the dataset and human expertise. The contribution can be summarised as follows: (1) machine learning to generate knowledge base that alleviates the knowledge acquisition bottleneck, (2) the human expertise maintenance that enables for incremental learning and (3) the mitigation of the failure detection problems reflected in previous research. Through this contribution, we have confidence in this adoption of this framework across multiple modalities.

Acknowledgments

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2017-0-01629) supervised by the IITP(Institute for Information & communications Technology Promotion). This research was supported by the MIST(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP(Institute for Information & Communications Technology Promotion)(2017-0-00093).

This work was also funded by the US Office of Naval Research grant, #GRANT12154887.

References

[1] R. Langone, C. Alzate, A. Bey-Temsamani, J.A. Suykens, Alarm prediction in industrial machines using autoregressive LS-SVM models, in: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2014, pp. 359–364.

[2] O. Foong, S. Sulaiman, D. Rambli, N. Abdullah, ALAP: Alarm prioritization system for oil refinery, in: Proceedings of the of the World Congress on Engineering and Computer Science, 2, 2009.

[3] B. Abdolhamidzadeh, T. Abbasi, D. Rashtchian, S.A. Abbasi, Domino effect in process-industry accidents–an inventory of past events and identification of some patterns, J. Loss Prev. Process Ind. 24 (5) (2011) 575–593.

[4] I.C. Gauld, J.M. Giaquinto, J.S. Delashmitt, J. Hu, G. Ilas, T. Haverlock, C. Romano, Re-evaluation of spent nuclear fuel assay data for the three mile island unit 1 reactor and application to code validation, Ann. Nucl. Energy 87 (2016) 267–281.

[5] M. Atzmueller, D. Arnu, A. Schmidt, Anomaly detection and structural analysis in industrial production environments, in: Data Science–Analytics and Applications, Springer, 2017, pp. 91–95.

[6] Y. Cheng, I. Izadi, T. Chen, Optimal alarm signal processing: filter design and performance analysis, IEEE Trans. Autom. Sci. Eng. 10 (2) (2013) 446–451.

[7] Z. Gao, C. Cecati, S.X. Ding, A survey of fault diagnosis and fault-tolerant techniques part i: fault diagnosis with model-based and signal-based approaches, IEEE Trans. Ind. Electron. 62 (6) (2015) 3757–3767.

[8] H.N.A. Pham, E. Triantaphyllou, The impact of overfitting and overgeneralization on the classification accuracy in data mining, in: Soft Computing for Knowledge Discovery and Data Mining, Springer, 2008, pp. 391–431.

[9] P. Compton, R. Jansen, Knowledge in context: A strategy for expert system maintenance, in: Proceedings of the Australian Joint Conference on Artificial Intelligence, Springer, 1988, pp. 292–306.

[10] F. Dahlstrand, Consequence analysis theory for alarm analysis, Knowl. Based Syst. 15 (1) (2002) 27–36.

[11] C.S. Ashley, D.B. Granatelli, J.C.H. Cheung, D.M. Shepherd, R.A. Weiss, B.I. Schultz, Historical alarm analysis apparatus and method, 2016, US Patent 9,355,477. Washington, DC: U.S. Patent and Trademark Office.

[12] C. Nan, F. Khan, M.T. Iqbal, Real-time fault diagnosis using knowledge-based expert system, Process Saf. Environ. Prot. 86 (1) (2008) 55–71.

[13] L. Abele, M. Anic, T. Gutmann, J. Folmer, M. Kleinsteuber, B. Vogel-Heuser, Combining knowledge modeling and machine learning for alarm root cause analysis, IFAC Proc. Vol. 46 (9) (2013) 1843–1848.

[14] O. Aizpurua, R. Galan, A. Jimenez, A new cognitive-based massive alarm management system in electrical power administration, in: Proceedings of the Seventh International Caribbean Conference on Devices, Circuits and Systems, IEEE, 2008, pp. 1–6.

[15] W. Zhao, X. Bai, W. Wang, J. Ding, A novel alarm processing and fault diagnosis expert system based on BNF rules, in: Proceedings of the IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific, IEEE, 2005, pp. 1–6.

[16] S. Ebersbach, Z. Peng, Expert system development for vibration analysis in machine condition monitoring, Expert Syst. Appl. 34 (1) (2008) 291–299.

[17] M. Schleburg, L. Christiansen, N.F. Thornhill, A. Fay, A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system, J. Process Control 23 (6) (2013) 839–851.

[18] J. Folmer, B. Vogel-Heuser, Computing dependent industrial alarms for alarm flood reduction, in: Proceedings of the Ninth International Multi-Conference on Systems, Signals and Devices (SSD), IEEE, 2012, pp. 1–6.

[19] K. Ahmed, I. Izadi, T. Chen, D. Joe, T. Burton, Similarity analysis of industrial alarm flood data, IEEE Trans. Autom. Sci. Eng. 10 (2) (2013) 452–457.

[20] I. Izadi, S.L. Shah, D.S. Shook, S.R. Kondaveeti, T. Chen, A framework for optimal design of alarm systems, IFAC Proc. Vol. 42 (8) (2009) 651–656.

[21] J. Zhu, Y. Shu, J. Zhao, F. Yang, A dynamic alarm management strategy for chemical process transitions, J. Loss Prev. Process Ind. 30 (2014) 207–218.

[22] G. Yin, Y.-T. Zhang, Z.-N. Li, G.-Q. Ren, H.-B. Fan, Online fault diagnosis method based on incremental support vector data description and extreme learning machine with incremental output structure, Neurocomputing 128 (2014) 224–231.

[23] P.K. Wong, Z. Yang, C.M. Vong, J. Zhong, Real-time fault diagnosis for gas turbine generator systems using extreme learning machine, Neurocomputing 128 (2014) 249–257.

[24] L. Wenyi, W. Zhenfeng, H. Jiguang, W. Guangfeng, Wind turbine fault diagnosis method based on diagonal spectrum and clustering binary tree SVM, Renew. Energy 50 (2013) 1–6.

[25] V. Muralidharan, V. Sugumaran, V. Indira, Fault diagnosis of monoblock centrifugal pump using SVM, Eng. Sci. Technol. Int. J. 17 (3) (2014) 152–157.

[26] C. Li, J. Zhou, Semi-supervised weighted kernel clustering based on gravitational search for fault diagnosis, ISA Trans. 53 (5) (2014) 1534–1543.

[27] D. Chivala, L.F. Mendonça, J.M. Sousa, J.S. da Costa, Application of evolving fuzzy modeling to fault tolerant control, Evol. Syst. 1 (4) (2010) 209–223.

[28] S.C. Han, H.-G. Yoon, B.H. Kang, S.-B. Park, Using MCRDR based agile approach for expert system development, Computing 96 (9) (2014) 897–908.

[29] P. Compton, L. Peters, G. Edwards, T.G. Lavers, Experience with ripple-down rules, Knowl. Based Syst. 19 (5) (2006) 356–362.

[30] D. Richards, Two decades of ripple down rules research, Knowl. Eng. Rev. 24 (2) (2009) 159–184.

[31] B.R. Gaines, An ounce of knowledge is worth a ton of data: quantitative studies of the trade-off between expertise and data based on statistically well-founded empirical induction, in: Machine Learning, Elsevier, 1989, pp. 156–159.

[32] S.B. Pham, A. Hoffmann, A new approach for scientific citation classification using cue phrases, in: Proceedings of the Australasian Joint Conference on Artificial Intelligence, Springer, 2003, pp. 759–771.

[33] B. Kang, P. Compton, P. Preston, Multiple classification ripple down rules: evaluation and possibilities, in: Proceedings of the Ninth Banff Knowledge Acquisition for Knowledge Based Systems Workshop, 1, 1995. 17–1.

[34] I. Bindoff, B.H. Kang, T. Ling, P. Tenni, G. Peterson, Applying MCRDR to a multidisciplinary domain, in: Proceedings of the Australasian Joint Conference on Artificial Intelligence, Springer, 2007, pp. 519–528.

[35] S.C. Han, L. Mirowski, B.H. Kang, Exploring a role for MCRDR in enhancing telehealth diagnostics, Multimed. Tools Appl. 74 (19) (2015) 8467–8481.

[36] S.C. Han, H.-G. Yoon, B.H. Kang, S.-B. Park, Using MCRDR based agile approach for expert system development, Computing 96 (9) (2014) 897–908.

[37] H. Motameni, A. Peykar, Morphology of compounds as standard words in persian through hidden Markov model and fuzzy method, J. Intell. Fuzzy Syst. 30 (3) (2016) 1567–1580.

[38] G.C. Tjhai, S.M. Furnell, M. Papadaki, N.L. Clarke, A preliminary two-stage alarm correlation and filtering system using SOM neural network and k-means algorithm, Comput. Secur. 29 (6) (2010) 712–723.

[39] N.B. Anuar, H. Sallehudin, A. Gani, O. Zakaria, Identifying false alarm for network intrusion detection system using hybrid data mining and decision tree, Malays. J. Comput. Sci. 21 (2) (2008) 101–115.

[40] B.R. Gaines, P. Compton, Induction of ripple-down rules applied to modeling large databases, J. Intell. Inf. Syst. 5 (3) (1995) 211–228.

[41] J. Cendrowska, Prism: an algorithm for inducing modular rules, Int. J. Man Mach. Stud. 27 (4) (1987) 349–370.

[42] P. Domingos, A few useful things to know about machine learning, Commun. ACM 55 (10) (2012) 78–87.

[43] C.L. Devasena, T. Sumathi, V. Gomathi, M. Hemalatha, Effectiveness evaluation of rule based classifiers for the classification of Iris data set, Bonfring Int. J. Man Mach. Interface 1 (2011) 5.

[44] M.M. Mazid, S. Ali, K.S. Tickle, Improved C4. 5 algorithm for rule based classification, in: Proceedings of the Ninth WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, World Scientific and Engineering Academy and Society (WSEAS), 2010, pp. 296–301.
[45] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[46] F. Jia, Y. Lei, J. Lin, X. Zhou, N. Lu, Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data, Mech. Syst. Signal Process. 72 (2016) 303–315.
[47] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 27.
[48] J.R. Quinlan, Improved use of continuous attributes in C4. 5, J. Artif. Intell. Res. 4 (1996) 77–90.