INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(2), 175-191, April 2018.

A Hybrid Failure Diagnosis and Prediction using Natural Language-based Process Map and Rule-based Expert System

D. Kim, Y. Lin, S. Lee, B.H. Kang, S.C. Han

Dohyeong Kim, Sungyoung Lee
Department of Computer Science and Engineering, Kyung Hee University,
1732, Deogyeong-daero, Giheung-gu, Yongin-si Gyeonggi-do 17104, Republic of Korea
dhkim@oslab.khu.ac.kr, sylee@oslab.khu.ac.kr

Yingru Lin, Byeong Ho Kang
School of Engineering and ICT,
University of Tasmania,
Private Bag 87, Hobart, Tas 7001, Australia
Yingru.Lin@utas.edu.au, Byeong.Kang@utas.edu.au

Soyeon Caren Han*
School of Information Technologies, University of Sydney,
Sydney, NSW, Australia
*Corresponding author: Caren.Han@sydney.edu.au

Abstract: Preventive maintenance is required in large scale industries to facilitate highly efficient performance. The efficiency of production can be maximized by preventing the failure of facilities in advance. Typically, regular maintenance is conducted manually in which case, it is hard to prevent repeated failures. Also, since measures to prevent failure depend on proactive problem-solving by the facility expert, they have limitations when the expert is absent, or any error in diagnosis is made by an unskilled expert. In many cases, an alarm system is used to aid manual facility diagnosis and early detection. However, it is not efficient in practice, since it is designed to simply collect information and is activated even with small problems. In this paper, we designed and developed an automated preventive maintenance system using experts' experience in detecting failure, determining the cause, and predicting future system failure. There are two main functions in order to acquire and analyze domain expertise. First, we proposed the network-based process map that can extract the expert's knowledge of the written failure report. Secondly, we designed and implemented an incremental learning rule-based expert system with alarm data and failure case. The evaluation results shows that the combination of two main functions works better than another failure diagnosis and prediction frameworks.
Keywords: expert's knowledge, preventive maintenance, failure prediction, alarm management, knowledge reuse.

1 Introduction

When a failure or error occurs in massive industrial facilities, it could cause industrial disaster or the factory operation shutdown, which brings enormous financial and social consequences [13]. Therefore, to maintain the normal workflow and maximize the profit, companies are running safety management processes, working with best efforts for the preventive maintenance and failure prediction. Preventive maintenance is the most important process to prevent failure before it occurs, and it aims to maintain the facilities in the best condition [9].

However, despite the periodic care, preventive maintenance cannot stop the continuous failures, since there are facilities where periodic maintenance is not effective enough. Furthermore, excessive maintenance without considering the cause behind the malfunction is inefficient and economically unproductive. After a failure occurs, it is necessary to take proper actions as soon as possible not only to restart the operation, but also to recover the rate of operation to avoid further financial damage. However, these actions are entirely dependent on the skilled and experienced human experts. The main problem here are that there is a limited number of skilled human experts, and the failure rate of such actions rises dramatically with the absence of such experts. Furthermore, the experts treat these errors based on experiences which may not always be recalled correctly. Their experiences may also not cover all sorts of problems as the system is prone to unknown failures. The possibility of human errors cannot be waived too. This makes depending on only expert experience undesirable [7].

Alarm system supports the facility maintenance by collecting the status of the facility from the installed sensors in real time and providing collected information to the manager in the control room. However, continuous operation of the facility causes deterioration, exchange of the components, and requires additional sensors. It changes the design of the initial alarm system and brings new types of alarms and occurrence patterns. Such phenomenon is known as alarm flooding which makes field workers ignore the alarms as the excessive amount of information cannot be handled. Many research projects have been undertaken to solve this problem, but most of them are about solving the alarm flooding problem. They focused on reducing the number of alarms to an affordable level, by extracting unusual alarms with a categorization of alarm types and analysis of the alarm patterns using statistical models. The limitation of such projects is that even if they succeed and the alarms are provided in manageable level, there are many additional works which require enormous time and efforts such as analysis of alarms, diagnosis, and failure prediction [1].

Since the cause of those system failures are not one dimensional, knowledge and experiences of an individual human expert is not enough to generalize the failure knowledge. Hence, failure knowledge must be built based on the continuously accumulated knowledge from several experts. However, this takes a lot of time that results in actions being delayed. Even when we can identify the abnormal facility status, financial damages are inevitable. In this paper, we propose a preventive maintenance system which finds the abnormal status in massive industrial facilities such as steelworks, and automatically provides the knowledge required for the maintenance and prevention of failures. We developed an expert system that recognizes the status of the facilities by using alarm data which can be used to find the abnormal status of the facilities. In this system, expertise regarding the alarms is arranged in a systematic structure, so that the system could recognize the status of the facilities with automatically generated alarms.

Furthermore, we also developed a system which contributes to the knowledge base by analysing failure reports so that it can reuse the accumulated failure records. Failure to reports include a description of analysis, diagnosis and actions taken related to the failure occurred. Thus, it contains knowledge regarding the causal relationship of failures as well as the implicit knowledge of experts regarding every step of the actions taken. This makes the failure report a primary source of solving the problem when failure occurs. Even though the failure circumstance is documented as a report, the report is still not systematically managed and well obtained the the relevant information.

A proper understanding the casual relationship of failure circumstances can enable us to diagnose the failure and to predict when the same problem occurs. We analyzed and processed failure reports which are composed of natural language, and converted them into structured knowledge, called as "knowledge map", for systematic storage and management. The knowledge about the alarm and failure reports are designed to have consistency in format so that they can operate interactively. When an alarm occurs, the status of alarm is detected and mapped with the process map by the expert system, which enables us to recognize the problem based on the status of facilities and to diagnose the process which made the facility status abnormal.

2 Related works

There are numerous researches related to alarm, thanks to its usefulness in maintaining and predicting faults of facilities in various fields. Outlier detection methods which are based on the analysis of alarm data aim to find the abnormal patterns with outlier values [7, 12].

However, such methods are not available in real-time, since they require to analyze massive amount of data, and interception of experts is also required in order to determine that the found abnormal alarms are indeed useful information related to the malfunctioning. There are also researches based on the machine learning, such as predicting alarms and temperature level of the steel production facilities using LS-SVM [8].

Knowledge based ALAP system is a rule based expert system, which aims to diagnose the faults with alarms [4]. Furthermore, a research has been done, which tries to handle the alarms efficiently by prioritizing the meaningful alarms from the alarm flooding using fuzzy logic [15]. There are also researches which focus on the design of alarm system.

In the research of [3], an evaluation of the alarm filters which are suggested in [1] to find the optimal ones. In the research of [1] clustering of alarm is done by evaluating the consecutive occurrence pattern of alarms. Also, various alarm management techniques including frameworks, data filtering, alarm delay and alarm deadlines are suggested, in order to design an optimal alarm system [5]. The researches which converge different methodologies include [16], which aims to manage alarm flooding by using both dynamic alarm management and Bayesian measurement. Such researchers are focusing on the alarm management and features analysis, and are limited in a sense that they are not considering the actual relation between the faults and alarms, and whether the managers are actually interested in the alarm.

In the field of natural language processing, the researches which extract knowledge from document with natural language have been done actively. Researches for extracting information, including specific terminology, objects, and concepts using machine learning, rule base, dictionary and corpus are, having been performed continuously, and among them there are researchers which especially interested in extracting entities, a relationship between entities, and occurring events from the medical documents and newspapers [6,10,11,14]. However, such researches are restricted to simply extract the information from the unstructured knowledge written in natural language. However, proposing fault analysis system is capable of reconstructing the extracted information into structured knowledge, which enables the systematic management of knowledge.

Previous researches for alarm processing mainly try to recommend meaningful alarm among the massive amount of alarms, or reduce the number of alarms to make it easy to take action after monitoring. Such researches suffer from high dependency on human experts, since the experts have to analyze the meaning of alarms after the alarm processing step after all. Proposing a system constructs an expert system on the relationship between alarms and facility, which enables to automatically recognize the status of a facility based on the occurring alarms. Furthermore, we converted the historic records of fault reports into structured system knowledge which include causal relationships. The structure of such knowledge is designed to be interoperable with the knowledge related to alarm. Therefore, fault diagnosis can be done by inferencing the status of the facility, mapping the inferred status to previous fault records, and tracking the problem occurred in the mapped case using causal relationship knowledge.

Prediction of fault which will happen after the current alarm can be done also, by reviewing the problem which occurred after the mapped previous case. Namely, the series of processes including diagnosis of the facility based on alarms, and fault diagnosis/prediction with the previous fault history is done automatically by the system.

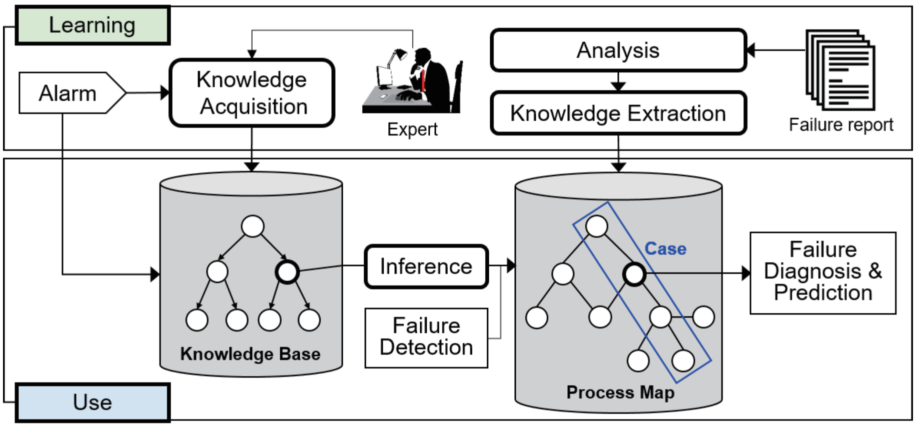

Figure 1: Overview of the proposed preventive maintenance system

3 Proposed preventive maintenance system

In this section, the architecture of proposing system and the main components are described. The proposed system is composed of two closely interoperating main components, a rule-based expert system for processing alarms and a failure analysis system to predict future system failures. A knowledge base for finding the abnormal status of the facility and preventive or diagnosis knowledge base about abnormal signs are essential to perform preventive maintenance automatically with the system. Those two kinds of knowledge come from distinct sources, but they share equivalent unit for the failure diagnosis and prediction. The expert system generates and manages the knowledge for finding the failure signs. The failure analysis system extracts knowledge from the failure reports, converts into causal relationship knowledge for failure diagnosis and prediction, and stores it into knowledge storage named as process map.

Figure 1 represents the preventive maintenance process via the expert system where failure prediction is done by failure prediction system. The proposed system uses alarms, which include signs of failures, and failure knowledge which includes causal relationship. All the alarms occur in real-time. Alarm system collects those alarms periodically for every hour and sends them to the expert system. An alarm, includes identification of facility in which the failure occurred, contents of the alarm, and quantitative numbers measured during an hour. Furthermore, it is stored in alarm storage for a certain period, and is used in alarm analysis and construction of the knowledge base by the field experts. The expert system possesses an inference engine which acquires the knowledge required for the recognition of problems from alarms, and knowledge acquisition engine which is used when the experts directly generate and manage knowledge in the knowledge base. Failure prediction is done using the inference engine and failure knowledge of the expert system after the alarm occurs.

The inference engine indicates towards facilities that are problematic during the occurrence of an alarm. It also searches for knowledge related to the problem, acquires and provides the cause of the problem and predicts possible additional failures. Thus, the time required for the field experts to recognize and solve the problem is greatly reduced, since the problem description, cause of the problem and viable actions are automatically provided by the system. When a failure occurs despite the preventive maintenance, the problem is diagnosed and solved by directly investigating in the field. A report of the entire process is written by the field experts.

The failure analysis system analyzes the failure reports in order to reuse expert experience knowledge. Failure reports include information related to the problem such as current status, cause, and actions taken, so that it can represent a causal relationship of the problem. Thus,

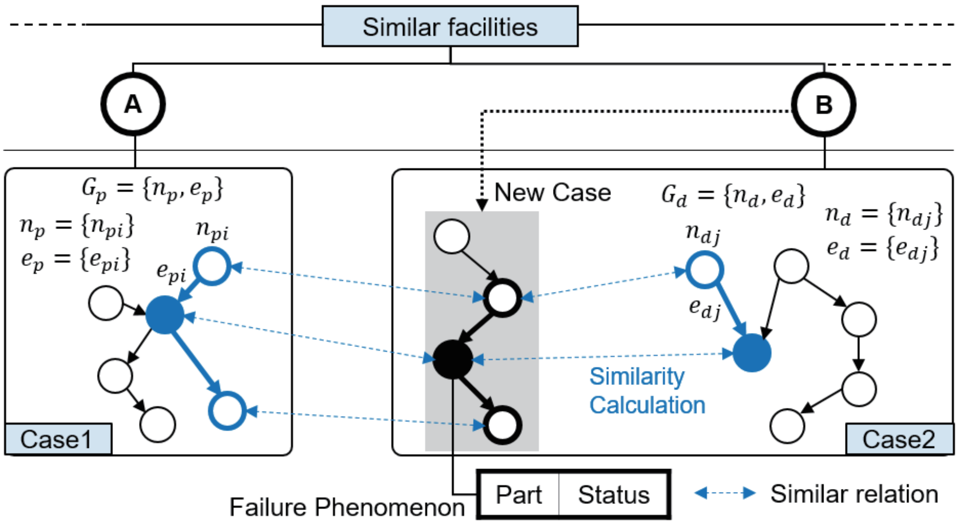

Figure 2: Conceptual design of process map

it can be used as a knowledge base for failure diagnosis and prediction. Failure reports include domain dependent terminologies, implicit representation, and are written in a unique way for each writer. This is the reason why the system utilizes not only the failure reports, but also the domain terminologies and domain knowledge. Constructed process map interoperates with the expert system to predict failure and to analyse of the cause of the failures. Figure 1 represents the conceptual process of the preventive maintenance. It represents the interaction between the expert system for alarm processing and the process map based on the failure report analysis. The failure cases which were acquired via such interaction can be used for failure diagnosis and failure prediction.

Knowledge is obtained by analyzing failure reports, extracting minimum semantic units from failure reports written in natural language, constructing causal relationships between them, and mapping the failure of the target facility. This knowledge is then labelled as failure knowledge. After that, failure knowledge is stored and managed in a process map. In the failure report, the simplest and basic type of sentence is the minimum semantic unit. Two components of the facility corresponding to the subject are extracted from the sentence: the subject that refers to the target of the facility and the predicate that means the status/operation of facility. The acquired unit knowledge with the form of node is called 'failure phenomenon'. After extraction process, subject indicate as Part of the component and predicate indicates as the status of the component. We analyzed failure reports and extracted short sentences from unstructured natural language. Order between two short sentences and consistency of meaning is considered while constructing a relationship which is referred to as failure case. A series of processes including how the specific facility has caused the problem and how the problem is solved, is configured in the order of occurrence through the relation between the failure phenomena.

Figure 2 shows the concept of a process map. The process map is based on the facility. There are numerous facilities in the factory, and many facilities have similar functions, but the type of facilities usually varies. Therefore, managing individual knowledge of these facilities can make knowledge generation very complex and knowledge management very inefficient. The proposed process map designates representative facilities with functions and designates the facilities having similar functions after the representative facilities. Having a similar function means that the structure of facilities is similar, and the usage of facilities and the relation with other facilities are also similar. This knowledge model is suitable for complex domains such as large factories

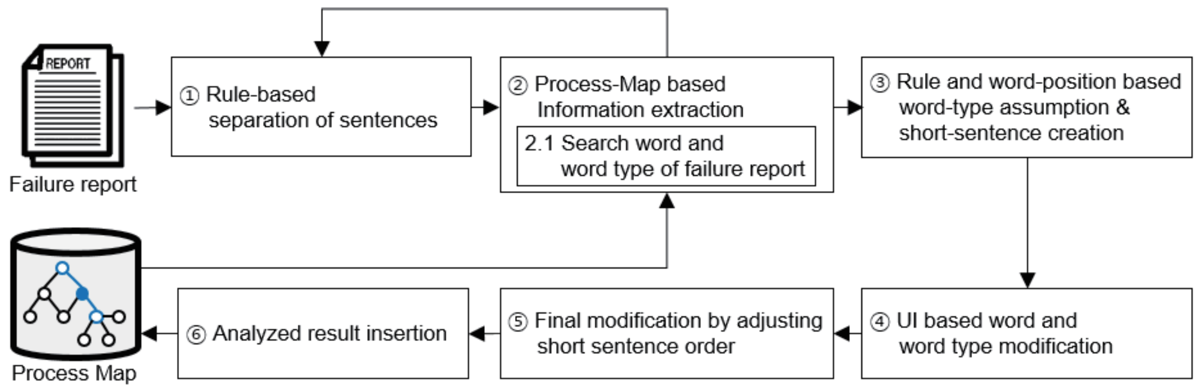

Figure 3: Process of failure report analysis

with various facilities. The failure report describes the failure analysis and failure action for the facilities that have a failure so that the failure cases analyzed from failure reports can be linked to these facilities. In Figure 2, A and B are facilities with similar functions where each facility has one or more failure cases and is represented by a single pass. If failure cases have the same failure phenomenon, failure phenomenon is expressed as multipath constructed by integrating and sharing failure phenomenon. For the similarity calculation, we applied following similarity formulation and text-based similarity algorithm:

$$Sim\left(n_{dj}, n_{pi}\right) = Sim_{Text}\left(n_{dj}, n_{pi}\right) + Sim_{Rel}\left(n_{dj}, n_{pi}\right) \tag{1}$$

$$Sim\left(n_{dj}, n_{pi}\right) = EditDistance\left(n_{dr}, n_{pir}\right) + EditDistnace\left(n_{djE}, n_{piE}\right) \tag{2}$$

Because similar failure phenomena of similar facilities have a relationship, whole structure of the process map is in a network. So, similar failure cases can be referenced in addition to direct failure case. In case that contents of failure knowledge are weak or non-existent, the process map that uses similar facilities and failure cases can easily be referenced in the indirect method. Thus, we can achieve high knowledge usability.

4 Process map construction

Failure reports written in natural language are transformed into structured forms using natural language processing techniques and then stored in the process map. The analysis is conducted automatically by the system and the results of that can be modified directly by the user through the user interface. The contents stored in the process map are used as information for analyzing the failure report, and then those analysed contents are connected to each other with the same relation so that the information can be easily accessed and expanded. Figure 3 shows the failure analysis process.

① **Rule-based separation of sentences:** This stage separates sentences into short clauses consisting of subject and predicate, which makes them the minimum units of a sentence. Sentences are separated using the regular expression based sentence separation rules.

② **Process-Map based Information extraction:** At this stage, the morpheme and part of speech from the short sentences separated in **step 1** are extracted and at **step 2.1.**, words are extracted from the process map as subject and predicate. After these two processes the type of that specific word is defined. If there are no exact matches with the words in the process map, partially matched words are retrieved. After that, if the extracted string from the process map is matched with the word in the short sentence, or if the string matches the search, the corresponding string marks in the short sentence and the string is registered as the candidate of the corresponding word type. Otherwise, if the string extracted from the process map does not completely coincide with the word within the short word, the corresponding string is marked and registered as a subordinate of the candidate word type.

③ **Rule and word-position based word-type assumption & short-sentence creation:** If the type of vocabulary recognized in **step 2** corresponds to the phenomenon (function-breakdown-action), it is partitioned into separate sentences based on the vocabulary. If the vocabulary is not recognized in **step 2**, the type estimation of the unregistered word is performed. If the vocabulary is not an abbreviation and the position of the word is at the end of the sentence, it is estimated as a status term. If there are duplicated semantic meanings, the similarity degree between short sentences is calculated. If the predefined threshold is exceeded, the one with highest similarity is selected, and the short sentences with lower similarities are removed. If the word corresponding to the object or phenomenon is omitted from the natural language itself, the word restoration is performed in three steps: 1) If the word corresponding to the object of the short sentence is omitted and there is a restorable object from the previous sentence, then the object of the short sentence is restored by using the object of the previous sentence. 2) If the target word of the short sentence is omitted, but the rest of the sentences cannot be restored, then the subject of the short sentences is restored using the subject of the next sentence. 3) If the status expression of the short sentence is omitted, then the state word from the following sentence is restated.

④ **UI based word and word type modification:** The analysis of the failure report is performed automatically up to the **step 3**, and at present stage, the user can directly supplement the contents through the UI. If the user wants to change the vocabulary type of the analyzed short sentences, they can change the vocabulary by searching them from the process map and the domain vocabulary using the interface.

⑤ **Final modification by adjusting short sentence order:** This allows the user to change the order of the short sentences which are automatically analyzed up to **step 3** through the UI. The user can change the order of the selected short sentences up, down and remove the selected short sentences. In addition, the user can define new short sentences and insert them at the desired positions.

⑥ **Analyzed result insertion:** It is the step of storing the result of the analysis that is finally completed in the process map. This is done after sorting the order of short paragraphs in the order of Phenomena-Cause-Actions and then save them in the process map. In the case of a failure-related short sentence, the object and status are inserted into the failure phenomenon part of the process map, and the cause and effect relationship between the input failure phenomena is established. Also, in case of a short sentence for an action, the object and phenomenon of the short sentence are entered in the action part of the process map, and the precedence relationship between the phenomena for the action is set up. Figure 4 is an example of a failure report and an analyzed failure case. The failure report includes the name of the equipment where the failure occurred, the date and time of the failure, the failure condition, and the cause of the failure. The analyzed failure cases are represented in the list on the right side of the failure report, and they can be reconstructed in the order of the failure cause, failure phenomenon, and countermeasure Method. It can also be used as causal knowledge. The blue letter is the object of the failure and the red letter indicates the status of the facility. In addition, by providing a user interface, field experts can directly edit the objects and phenomenon of the failure and adjust the order of the failure phenomenon.

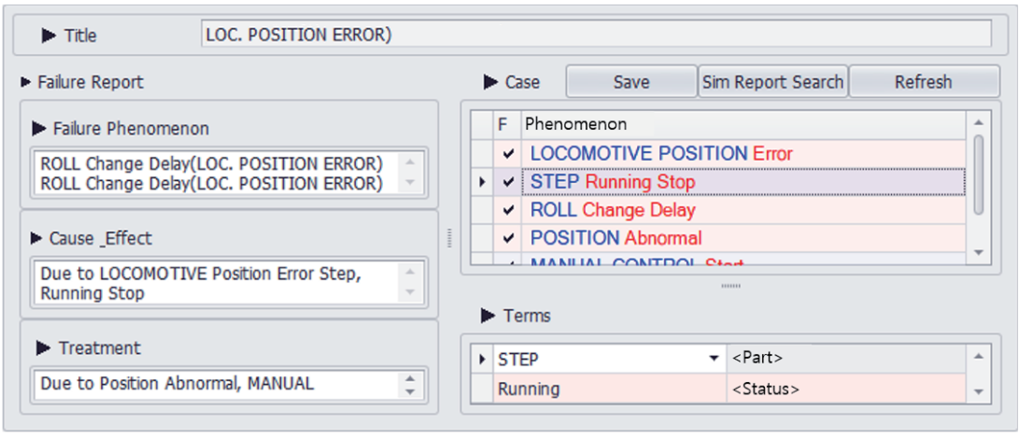

Figure 4: Sample failure report and its failure case

5 Alarm knowledge construction

The alarm knowledge is stored and managed in the knowledge base which is built by field experts to capture the problem of the system in real time. The expert system for alarm handling captures the failure indications from the alarm and retrieves the appropriate failure cases from the process map and provides them to field experts in predicting future failures or diagnosing current failures. The alarm knowledge constituting the system is generated after considering the relationship between failures and alarms using the user interface by field experts who have experienced a failure. Knowledge is modeled on an IF-THEN-based rule basis to represent human knowledge. The field experts generate and continuously manage the phenomena in the knowledge base which is generated by the alarm pattern. The knowledge acquisition engine also works based on the experience of the experts. When an hourly alarm is collected, the inference engine finds the appropriate rule to process the alarm from the knowledge base and derives the result. The derived results are used to find failure cases matched in the process map.

The main structure of the expert system is illustrated in Algorithm 1. The alarm list for each hour $Alarm_i$ is used as the input for inference against the expert knowledge base KB (Line 10). The inference starts from the root rule R_0. Once the current starting rule is satisfied, all the child rules will be evaluated. Only when there is no child rule for the current starting rule will the current starting rule become the inference result (Line 18). If there is any child rule can be satisfied by the $Alarm_i$, the inference will continue with its child rules of this satisfied rule until no more child rules can be satisfied (Line 23 - 24). Otherwise the current starting rule is stored as one of the inference-resulted rule (Line 27). The final result of the inference $Result$ will be a collection of rules that are satisfied by the $Alarm_i$. The conclusion of the each satisfied rule is in the form of the phenomenon that is stored in the process map PM. These phenomena from the conclusion will be used to find the corresponding failure case in the PM that contains the same phenomenon (Line 31-37). The failure diagnosis and prediction in the failure case can then be utilized.

5.1 Input case

Input cases of the expert system are the alarm lists consisting of multiple alarms and which are used for failure prediction and knowledge acquisition. The alarm system collects alarms every hour in real-time and forwards them to the expert system after processing those alarms. The alarm data consist of seven attributes such as facility ID, facility name, alarm ID, alarm name,

Algorithm 1 Expert System Structure

```
LET Array Alarm_i ← Alarm list for the ith hour
LET rule set KB ← rule-base knowledge base
LET rule R_j ← the rule in the KB with the id j (R_0 is a root rule)
LET String Con_j ← conclusion of R_j
LET Array Result ← collection of satisfied rules
LET Object PM ← process map
LET case C_k ← the failure case in the PM
LET String Phe_k ← the phenomenon of C_k
LET Array Result_PM ← matched failure cases with inference result Result
function INFERENCE(Alarm_i, KB)
    if satisfied(R_0, Alarm_i) then
        if R_0 has child rules then
            for all child rule Child do
                if satisfited(Child, Alarm_i) then
                    if hasChild(Child)=TRUE then
                        Inference(Child, Alarm_i)
                    else
                        push(Result, Child)
                        pop(Result, R_0)
                    end if
                end if
            end for
            if no child rule satisfied then
                push(Result, R_0)
            end if
        else
            push(Result, R_0)
        end if
    end if
end function
function RESULTMATCHING(Result, PM)
    for all Con_j of R_j in Result do
        for all Phe_k in PM do
            if matched(Con_j, Phe_k) then
                push(Result_PM, Phe_k)
            end if
        end for
    end for
end function
```

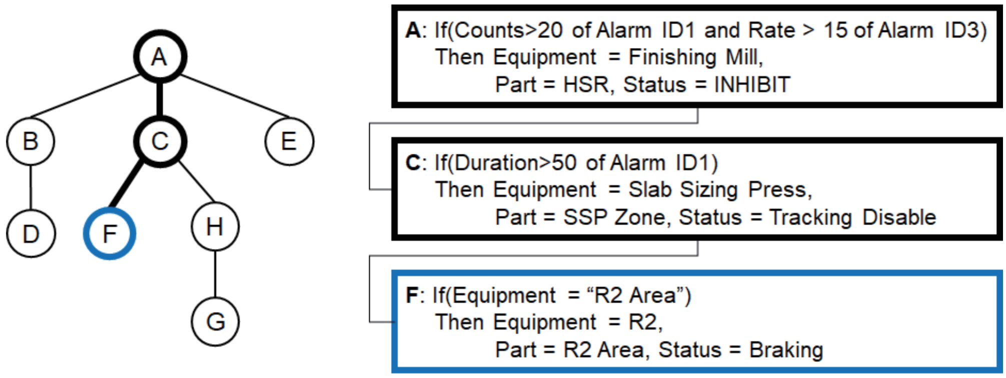

Figure 5: Knowledge base inference process

counts of alarm, the lifetime of the alarm and rate of the alarm.

5.2 Alarm knowledge representation

Field experts can create rules by handling alarms collected through the knowledge acquisition engine. The attributes used in the condition part of the rule are as follows: 1)Facility name: The facility that generates alarms (e.g. Finishing Mill), 2)Alarm message: The textual representation of an alarm (e.g. F3 BOT PC APC ERROR), 3)Counts: The number of times the alarm occurred in every hour (e.g. 1), 4) Duration: The duration of the alarm in every hour (e.g. 96), 5) Rate: The ratio of the alarm to one day (e.g. 26.67)

The conclusion of the rule is the same structure as the unit knowledge base in the process map. Thus, the conclusion part consists of the facilities, objects, and phenomena. The field expert who generates the rule can define the conclusion by searching and selecting the failure phenomenon in the process map. Therefore, since the inference results of the failure prediction system are mapped to the failure knowledge, a process map can be used for failure prediction. Figure 5 shows an example of a knowledge base where each rule defines the rule condition using five attributes of the alarm, and the conclusion is defined using the phenomenon which constitutes the case of the process map. For example, Rule 1 is defined as a condition with the number of occurrences (Count), alarm index (Rate), and alarm index number 3, and the conclusion of the rule is defined as Part and Status, which are phenomena of the case.

As another example, we can see in rule 2 that the conclusion of the case is based on the alarm message of alarm #1 and the conclusion of the rule is linked to the knowledge of the process map.

5.3 Preventive management system

The proposing preventive maintenance system monitors the current state of the facility with real-time alarms. It is also capable of conducting failure diagnosis and failure prediction via historic failure cases which is in causal-relationship format. Figure 6 represents an example of the operation of the preventive maintenance system. Four alarms are included in the alarm list collected during an hour, and the alarm list is used as the input case for the failure prediction system. Since the rule for the first alarm is stored in the knowledge base, inference engine evaluates the alarm list with the rule corresponding to the first alarm from the knowledge base and obtains the result when the rule is satisfied. The result of the rule is a failure phenomenon, and it represents that a certain phenomenon will occur at the target facility SSP zone. In order to predict possible failures from the acquired failure phenomenon, the system searches

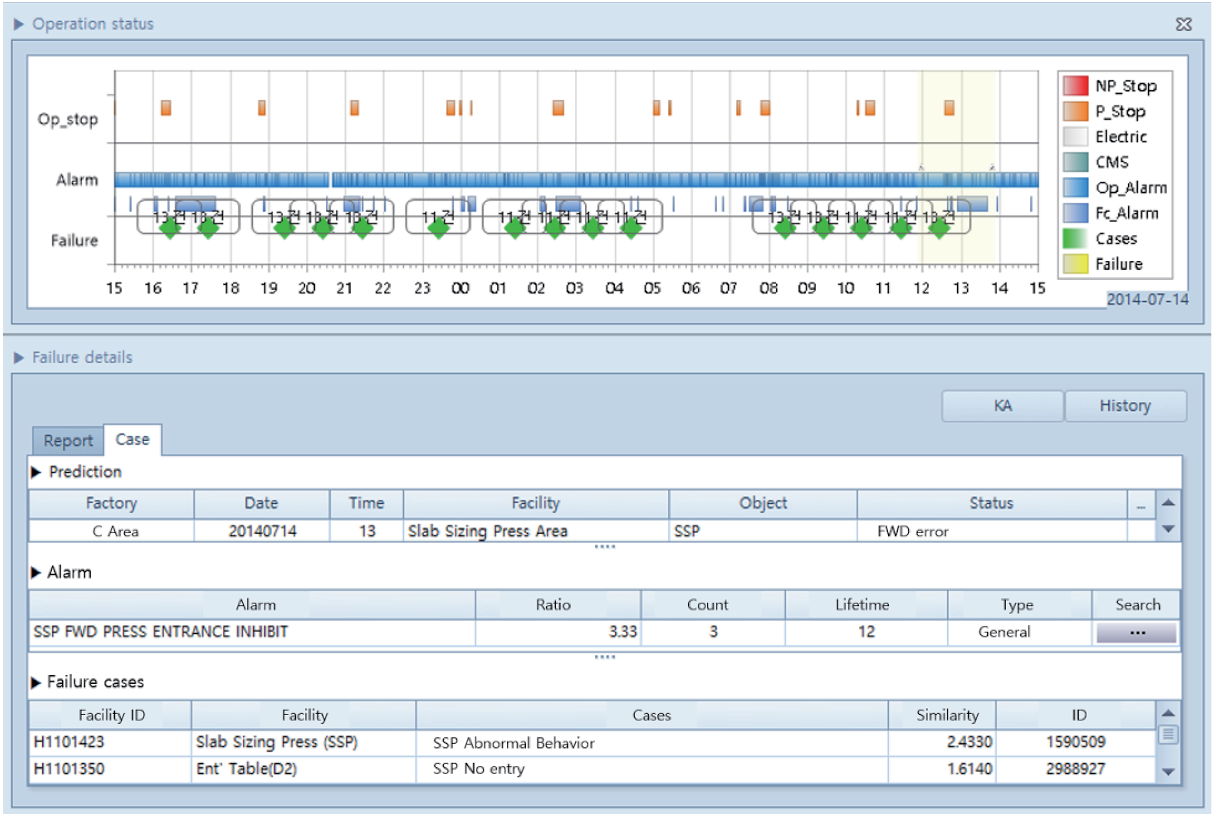

Figure 6: Preventive maintenance system interface

for failure cases which contains an equivalent failure phenomenon from the process map. The system provides the failure cases related with the facility Slab Sizing Press Area, the designated facility in the alarm, to the field expert who is monitoring the alarm. Field expert checks the phenomenon of the problematic facility and actions they can take. They are also capable of analyzing the cause of the failure by tracking the previous failure case from backward.

5.4 Implementation

In this paper, the example process of the proposed system is described, which currently operates in the steelworks industry. Figure 6 shows the implemented screen of the preventive maintenance system. In the alarm occurrence area located on the upper part of the screen, the occurrence status of the alarm is displayed every hour. It also displays the number of predictions for failures when the alarm occurred in the alarm occurrence area, which obtained using alarm data and failure knowledge. Users can check the details of the alarm occurrence and failure cases for the alarm via the user interface by adjusting the specific time zone. In the left bottom side of the screen, details of the alarm occurrence are displayed, including the start time and end time of the alarm, a facility which signalled the alarm, contents of the alarm, and the ratio that the individual alarm possesses. The alarms are classified as facility alarm or operation alarm. In the right bottom side of the screen, failure occurrence detail panel shows the details of the failures which occurred in the time zone that the user selected.

The failure prediction result is the inference result about the alarm that shows the SSP entrance prohibited status. The failure related alarm table shows the alarm which is primarily used during the inference process. The pseudo-failure case shows all the failure cases, including

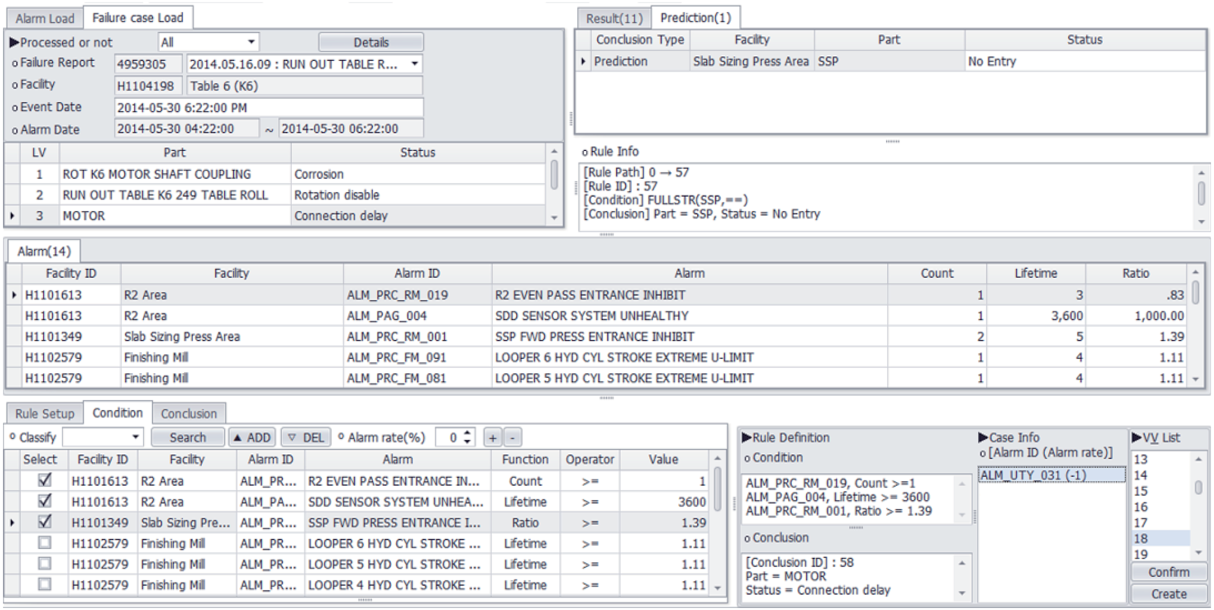

Figure 7: Knowledge acquisition from domain expert

the inference result, and it also measures, sorts, and displays the degree of similarity in order to show if the inference result is exactly included. Figure 7 represents the details of the failure case, which is displayed when the user selects a specific failure case from a failure case list in order to add a new knowledge. On the left side, the contents of the failure cases, which are generated by analyzing failure reports, sorted in the causal relationship. Each failure phenomenon is classified as failure and action, based on its attributes, which in this example, enables field experts to predict when SSP, SSP PRE-FORMING will occur. Based on those steps, the field experts can check the problem and add new failure prediction knowledge.

6 Evaluation

The failure prediction performance of the proposed system is evaluated through experiments, in order to prove that the failure cases of the process map indeed include causal relationships.

6.1 Experiment data

The data used in the experiment includes the failure reports for the failures that occurred more than once in domestic steelwork, and the alarm data collected 1 hour before and after the failure occurrence. All the alarm data are collected in real-time. In the proposed system, the data is pre-processed in the unit of 1 hour with the number of occurrences, occurred periodically, and the ratio of occurrence. From October 2012 to July 2016, a total number of 502,308 alarm data were collected. The failure report uses the failure cases built in the process map by analyzing 400 failure reports among 713, which includes the failures that occurred more than once, and are collected during the same period. The number of occurrences for the identical failures is not constant, but 4 iterated failures have occurred on average. The knowledge base of the expert system consists of 237 rules built by two field experts. For the experiment, training data and test data were used in a 6 to 4 ratio. 100 failure data and 200,923 alarm data were used as the test data.

Table 1: Top 10 satisfied rules

No.	Frequency	Rule Id	Failure Description
1	13.87%	0	Default (root) rule
2	9.98%	2	Detect the LEAK using the starting time
3	8.87%	17	Detect the BURN using lifetime
4	3.48%	201	Detect the DEFECTION using alarm id
5	2.99%	38	Detect the HUNTING using lifetime and facility id
6	2.61%	120	Detect the NO LINK using ratio and count
7	2.02%	7	Detect the NO REVERSE using alarm id and count
8	1.81%	79	Detect the SLIP using lifetime and ratio
9	1.47%	22	Detect the TRANSFORM using lifetime
10	1.03%	19	Detect the GAP using alarm id and facility id

6.2 Experiment method

The experiment done here consists of the following two parts. The first part shows the possibility of failure prediction method which is based on the alarm and failure knowledge, by evaluating the success rate of failure prediction with the degree where the inference results of the alarm and failure phenomenon of the failure case are mapped. The second part shows the superiority of the proposed system by comparing failure prediction accuracy of the proposed system and three previous types of research on failure prediction. Failure prediction is performed by inputting alarms into the system in the order of occurrence time, and by evaluating if the order of the reasoning results is equivalent to the order of failure phenomenon in that failure case. The evaluation is done by the following processes. For the reliability of the constructed alarm knowledge and failure cases, two field experts who are in charge of the facility monitoring at the actual steelworks evaluated the inference result of the alarm generated by the expert system. They compared the order of inference results and failure case to confirm if the cause of the actual failure and occurred failure is equivalent.

6.3 Experiment result

Knowledge analysis

The ten most-frequently satisfied rules are ranked and shown in Table 1, highlighting the most frequent failure cases. The most-frequently rule was the normal (rule 0) which does not have any failure to predict. Among 400 failure type, the system found LEAK (rule 17 - If the count is larger than 9 and lifetime is more than 8 hours, Then the failure is LEAK) and BURR (rule 201 - If lifetime is less than 2 hours, Then the failure is BURR) Figure 8 shows the seasonal frequency of satisfied rules coupled with the real depth (which is their level in the decision tree), indicating failure prediction conceptual depth. Note that the root (default) rule is level 1. As can be seen in the figure, the most common rule depth is 4. Rules at this depth level includes the combination of various types of attribute sets.

Performance evaluation of failure prediction

To evaluate the superiority of the proposing failure prediction method, we conducted a comparison with the previous failures prediction methods. The studies compared with the paper are as follows:

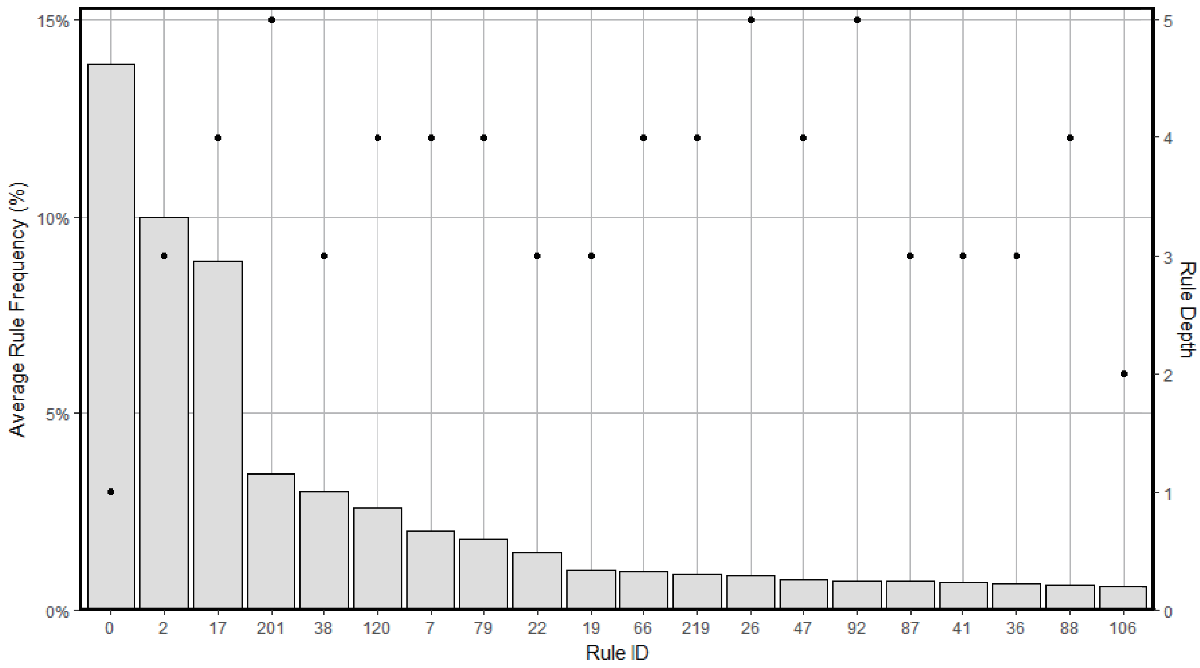

Figure 8: Inferred rule frequency and depth

Table 2: Review of failure prediction by previous failure prediction system

Author	**Description**	**Accuracy**
Santos et al. (2010)	Applied different machine learning techniques (incl. Bayesian Network, SVM, and decision tree)	81.4%
Liu and Jiang (2008)	Used particle filter with Bayesian Inference	64.2%
Chen et al. (2015)	Applied knowledge-based neural fuzzy inference	90.3%
Proposed System	Natural Language-based Processing Map + knowledge-based alarm prediction system	**95.7%**

- Santos et al.(2010) [13]: The authors proposed prediction method based on machine learning, in order to predict the major failures in a casting factory domain. They compared Bayesian network, SVM, and decision tree, and concluded that decision tree has the high prediction success rate.
- Liu and Jiang (2008) [9]: The authors used particle filter, which is frequently applied in signal processing, Bayesian inference, and machine learning. The research tried to predict failures with a hybrid system in the discrete-continuous composite environment.
- Chen et al.(2015) [2]: The authors used knowledge-based neural fuzzy inference for the failure prediction of turbines in wind power plant. The performance of failure prediction method proposed in this paper is evaluated by comparing with the algorithms of [13], [9], and [2] in an equivalent environment. To be specific, decision tree of [13], particle filter of [9], and neural fuzzy inference of [2] are used with the alarm data proposed in this paper. On Table 2 failure prediction accuracy between each experiment are compared.

Due to the nature of alarm data, the decision tree of [13] classified various variables into exact horizontal and vertical relations, which is not adequate for classification. As a result, the complexity of the tree grows rapidly and pruning becomes hard, which resulted in an accuracy of 81.4%, slightly lower than the result of applying general decision tree. Since the particle filter used in [9] is based on Monte-Carlo method, which requires a massive amount of sample data, [9] showed the lowest accuracy of 64.2%. The neural-fuzzy inference method of [2] is a method which introduced learning ability of neural network into the conventional fuzzy logic method. Since it is a method which enables continuous learning by granting the learning ability to the expert knowledge-based fuzzy logic system, its key features are well utilized in the processing of complex and continuously accumulated real-time alarm data, which is reflected when it shows the similarities result recorded in its paper, 90.3%. The failure prediction success rate of the proposed method is 95.7%, which is superior to the methods used in the comparison. The reason behind this is that firstly the experts who have diverse experience in failures constructed the knowledge base directly, and secondly, the knowledge which contains real failure cases and a causal relationship is used to predict failures. Therefore, we can interpret the high accuracy as a result of utilization of high-quality knowledge which appropriately represents the actual failure cases.

7 Conclusion

The preventive maintenance system proposed in this paper is an effective alternative directly related to the popular smart factory for two reasons. It is based on the knowledge of experts, which greatly lowers the dependency towards human labor, and it enables effective failure prediction and diagnosis by the system. The proposed system can be utilized in various domains since it focused on the knowledge of experts which was not easily reusable before in specific domains. Proposed failure analysis system is meaningful in a perspective that it suggests new methods for knowledge sharing and generalization, which was long considered impossible. The knowledge of facilities or failures are easily obtained from manuals or reports, but the problem was that the total amount of information was enormous, and it was not easy to find exactly the information we wanted. With those reasons, the usability of failure reports went down, which made many field experts write the failure reports perfunctorily, resulting in lower low quality of reports. Such problems can be solved by improving the working environment with the help of the system. Although the technical approach of the system is meaningful to a certain extent, the system must be understood in order to be utilized in actual operation. The problem with the current failure report is that it is not easy to understand the reports for both human and systems, due to the excessive use of shortened words and specific terminology. If the human understands the working process of the system, they will write the failure reports in a way that the system could understand, which can result in a fairly accurate analysis of the system, which will reduce the dependency towards human labor. If the quality and usability of failure cases go up, the usefulness of expert system which interoperates with failure cases will also go up. In order to utilize the failure cases, the field experts will accumulate the experiential knowledge of alarm into the knowledge base regularly, and if the knowledge with high accuracy is continuously gathered to a certain extent, the system could utilize such expert knowledge to improve the accuracy of failure prediction and diagnosis with alarms. Therefore, with this process, we can overcome the disasters and human injuries, by maximizing the efficiency of preventive maintenance in the actual industrial field.

Acknowledgements

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 work was supported by the Industrial Core Technology Development Program (10049079, Develop of mining core technology exploiting personal big data) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea). This work was also funded by the US Office of Naval Research grant, #GRANT12154887.

Bibliography

[1] Ahmed, K.; Izadi, I.; Chen, T.; Joe, D.; Burton, T. (2013); Similarity analysis of industrial alarm flood data, *IEEE Transactions on Automation Science and Engineering*, 10(2), 452-457, 2013.

[2] Chen, B.; Matthews, P. C.; Tavner, P. J. (2015); Automated on-line fault prognosis for wind turbine pitch systems using supervisory control and data acquisition, *IET Renewable Power Generation*, 9(5), 503-513, 2015.

[3] Cheng, Y.; Izadi, I.; Chen, T. (2013); Optimal alarm signal processing: Filter design and performance analysis, *IEEE Transactions on Automation Science and Engineering*, 10(2), 446-451, 2013.

[4] Foong, O.; Sulaiman, S.; Rambli, D. R. B. A.; Abdullah, N. (2009); ALAP: Alarm prioritization system for oil refinery, *Proc. of the World Congress on Engineering and Computer Science*, 2, 2009.

[5] Izadi, I.; Shah, S. L.; Shook, D. S.; Kondaveeti, S. R.; Chen, T. (2009); A framework for optimal design of alarm systems, *IFAC Proceedings Volumes*, 42(8), 651-656, 2009.

[6] Ju, Z.; Wang, J.; Zhu, F. (2011); Named entity recognition from biomedical text using SVM; *Bioinformatics and Biomedical Engineering,(iCBBE) 2011 5th International Conference on*, 1-4, 2011.

[7] Kang, B. H.; Kim, Y. S.; Chen, Z.; Kim, T. (2013); Detecting significant alarms using outlier detection algorithms, *Interdisciplinary Research Theory and Technology (IRRT 2013)* 1-8, 2013.

[8] Langone, R.; Alzate, C.; Bey-Temsamani, A.; Suykens, J. A. (2014); Alarm prediction in industrial machines using autoregressive LS-SVM models, *Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on*, 359-364, 2014.

[9] Liu, Y.; Jiang, J. (2008); Fault diagnosis and prediction of hybrid system based on particle filter algorithm, *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, 1491-1495, 2008.

[10] Mohapatra, H.; Jain, S.; Chakrabarti, S. (2013); Joint Bootstrapping of Corpus Annotations and Entity Types, *EMNLP*, 436-446, 2013.

[11] Morwal, S.; Jahan, N.; Chopra, D. (2012); Named entity recognition using hidden Markov model (HMM), *International Journal on Natural Language Computing (IJNLC)*, 1(4), 15-23, 2012.

[12] Orair, G. H.; Teixeira, C. H.; Meira Jr, W.; Wang, Y.; Parthasarathy, S. (2010); Distance-based outlier detection: consolidation and renewed bearing, *Proceedings of the VLDB Endowment*, 3(1-2), 1469-1480, 2010.

[13] Santos, I.; Nieves, J.; Bringas, P. G. (2010); Enhancing fault prediction on automatic foundry processes, *World Automation Congress (WAC)*, 1-6, 2010.

[14] Sawsaa, A.; Lu, J. (2011); Extracting information science concepts based on jape regular expression, *WORLDCOMP'11The 2011 World Congress in Computer Science, Computer Engineering, and Applied Computing*, 18-21, 2011.

[15] Zhao, W.; Bai, X.; Wang, W.; Ding, J. (2005); A novel alarm processing and fault diagnosis expert system based on BNF rules, *Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES*, 1-6, 2005.

[16] Zhu, J.; Shu, Y.; Zhao, J.; Yang, F. (2014); A dynamic alarm management strategy for chemical process transitions, *Journal of Loss Prevention in the Process industries*, 30, 207-218., 2014