



Democratic and Popular Republic of Algeria Ministry of
Higher Education and Scientific Research
University Abbes Laghrou - Khenchela
Faculty of Science and Technology
Department of Mathematics and Computer Science



Dissertation

Presented to obtain the Master's diploma in Computer Science

Specialty: Software engineering and distributed systems

Modeling of fault detection in an industrial system using metaheuristic approaches

Presented by:

LAHMARI ABDELHAMID

BENELHABES AOUF RAZKI

Supervised: by:

Dr. MAHDAOUI RAFIK

Board of Examiners

Dr. HOUASSI HICHEM

Dr. MAHDAOUI RAFIK

Dr. RAHAB HICHEM

President

Supervisor

Examiner

University of Khenchela

University of Khenchela

University of Khenchela

2022/2023

Acknowledgment

First, we would like to express our deepest gratitude and appreciation to our supervisor Mr. MEHDAOUI Rafik for his support, guidance, encouragement, and patience throughout the process of completing this research and for the help and references he provided us with.

We would like to show our appreciation to the committee members for accepting to read, evaluate and comment on this dissertation.

*We would like to express our deepest appreciation to all our teachers at the Department of Mathematics and computer sciences at Abbess Laghrour University.
and to every individual who played a part in its accomplishment.*

Dedication

I dedicate this work ...

To my beloved parents ...

To my brothers and sisters ...

To my little family: my wife and my children...

To all my friends.

Abdelhamid

Dedication

dedicate this work ...

To my dear parents ...

To my brothers ...

To my friends.

May Allah keep them

Razki

Abstract

The process of extracting knowledge from data is defined as the process of analyzing data from many perspectives and discovering models from collections of useful data to predict the outcomes that will help us make the best decision.

This procedure involves several steps, from data pre-processing (including data collection, data cleaning and integration), to the acquisition of results, their confirmation and interpretation, and finally the integration of acquired knowledge.

There are numerous data analysis techniques that can be used to generate models, including classification techniques that generate models (learning-phase) that can be used to predict the future data of an object (prediction-phase) based on data from other similar objects.

In this work we will implement a new approach for fault detection in industrial systems. It will be based on Grey Wolf Optimization algorithm which belongs to swarm intelligence techniques. This approach will allow generating effective classification models to predict any potential failure.

Our work is led by a Python and Weka application that allows us to collect new data and learn how to develop classification models.

ملخص

تعرف عملية استخراج المعرفة من البيانات على أنها عملية تحليل البيانات من العديد من وجهات النظر واكتشاف النماذج من مجموعات البيانات المفيدة للتنبؤ بالنتائج التي ستساعدنا في اتخاذ القرار الأفضل.

وينطوي هذا الإجراء على عدة خطوات، من التجهيز المسبق للبيانات (بما في ذلك جمع البيانات، وتنظيف البيانات ودمجها)، إلى الحصول على النتائج، وتأكيدها وتفسيرها، وأخيراً إدماج المعارف المكتسبة.

هناك العديد من تقنيات تحليل البيانات التي يمكن استخدامها لتوليد النماذج، بما في ذلك تقنيات التصنيف التي تولد نماذج (مرحلة التعلم) التي يمكن استخدامها للتنبؤ بالبيانات المستقبلية لكائن (مرحلة التنبؤ) بناءً على بيانات من كائنات مماثلة أخرى.

في هذا العمل سننفذ نهجاً جديداً لاكتشاف الأخطاء في الأنظمة الصناعية والذي سيعتمد على خوارزمية

Grey Wolf Optimization التي تنتمي إلى تقنيات ذكاء السرب. سيسمح هذا النهج بتوليد نماذج تصنيف فعالة للتنبؤ بأي فشل محتمل.

يعتمد عملنا على تطبيقي **Python** و **Weka** اللذان يسمحان لنا بجمع بيانات جديدة ومعرفة كيفية تطوير نماذج

التصنيف.

Table of contents

List of figures.....	viii	
List of tables.....	ix	
General introduction.....	10	
Chapter 1	INDUSTRIAL DIAGNOSTIC	12
1. Introduction.....	13	
2. Terminology.....	13	
3. Industrial diagnostic.....	14	
4. Diagnostic methods.....	15	
4.1. Model-based methods	16	
4.1.1 Physical redundancy.....	17	
4.1.2 Analytical redundancy.....	17	
4.1.3 Parameter-estimation methods.....	17	
4.2. Model-free methods.....	18	
4.2.1 Pattern recognition methods.....	18	
4.2.1.1 Expert systems.....	18	
4.2.1.2 Fuzzy approach.....	18	
4.2.1.3 Classification rules.....	18	
4.2.2 Methods based on static tools.....	18	
Conclusion.....	19	
Chapter 2	DATA MINING	20
1. Introduction.....	21	
2. Definition.....	21	
3. Data, information and knowledge.....	22	
4. Types of data in data mining process.....	22	
4.1 Relational database.....	22	
4.2 Data warehouse.....	23	
4.3 Transactional data.....	23	
4.4 Object-oriented relational database.....	23	
4.5 Other types of data	23	
5. Data mining techniques and algorithms.....	23	
5. 1 Supervised data mining.....	23	
5.1.1 Classification.....	23	
5.1.1.1 Decision Tree (DT)	24	
5.1.1.2 Neural network (NN)	25	
5.1.1.3 Rule-based classification.....	25	
5.1.1.4 Naive Bayes (NB)	26	
5.1.1.5 K -Nearest Neighbors (KNN)	26	
5.1.1.6 Support Vector Machines (SVMs)	26	
5.1.2 Regression.....	26	

5.1.2.1 Logistic regression.....	26
5.1.2.2 Linear regression.....	26
5.2 Semi-supervised learning.....	26
5.3 Unsupervised learning.....	26
5.3.1 Clustering.....	27
5.3.1.1 K-mean.....	27
5.3.1.2 K-medoids.....	27
5.3.2 Association.....	27
5.3.2.1 Apriori.....	27
5.3.2.2 Multilevel.....	27
6. Data mining applications.....	27
6.1 Manufacturing.....	28
6.2 Sales and marketing.....	28
6.3 Banking and insurance.....	28
6.4 Telecom, media, and technology.....	28
Conclusion.....	28
Chapter 3	NOVEL GWO ALGORITHM FOR
	INDUSTRIAL DIAGNOSTIC
	29
1. Introduction.....	30
2. Grey Wolf Optimizer algorithm.....	30
2.1 Encircling prey.....	31
2.2 Hunting.....	33
2.3 Search for prey (exploration)	33
2.4 Attacking the prey (exploitation)	34
3. Case study description.....	35
3.1 Abstract of used tools.....	35
3.1.1 WEKA.....	35
3.1.2 Python.....	36
3.1.3 Python's commonly used libraries.....	36
3.2 Tennessee Eastman process (TEP) presentation.....	37
3.2.1 TEP faults.....	40
3.2.2 Evaluation of DiscreteGWO on TEP process.....	41
3.2.3 Experimental results.....	47
3.2.4 Model evaluation criteria.....	48
3.2.5 Discussion.....	49
Conclusion.....	49
General conclusion.....	50
Bibliography.....	51

List of Figures

- 1.1 [Progression of fault toward malfunction or failure](#)
- 1.2 [Diagnostic procedure](#)
- 1.3 [Industrial diagnostic methodology classification](#)
- 1.4 [Parameter estimation method](#)
- 2.1 [Knowledge discovery process](#)
- 2.2 [Data mining techniques and algorithms](#)
- 2.3 [A classification model with different representations](#)
- 3.1 [Generic structure of a classification rule](#)
- 3.2 [Hierarchy of grey wolves](#)
- 3.3 [The position of a wolf can be defined around a prey with considering a 2D mesh around the prey](#)
- 3.4 [TEP diagnosis model generation methodology](#)
- 3.5 [WEKA tools](#)
- 3.6 [Tennessee Eastman Process schema](#)
- 3.7 [Position update in one iteration](#)
- 3.8 [Position update code1](#)
- 3.9 [Position update code2](#)
- 3.10 [Wolves' fitness calculation code1](#)
- 3.11 [Wolves' fitness calculation code2](#)
- 3.12 [New generated rule code](#)
- 3.13 [Classification Diagram](#)

List of tables

- 3.1 [Process measurements](#)
- 3.2 [Composition measurements](#)
- 3.3 [Manipulated variables](#)
- 3.4 [TEP Faults](#)
- 3.5 [TEP dataset](#)
- 3.6 [TEP dataset' features discretization](#)
- 3.7 [TEP dataset after discretization](#)
- 3.8 [TEP ruleset for fault # 04](#)
- 3.9 [DiscreteGWO results compared to different models.](#)

General introduction

By the increasing of human demand for more goods, industrials have been obliged to increase the rate of production. Automated processes allowed producing more products in short time with good quality, while increasing the flexibility and precision. Since 1975, integration of automated systems in the industry grows rapidly supported by the reasonable cost. In addition, the great development in computer systems enhanced the presence of automates everywhere.

This development wasn't without cost. Automates made industrial processes complicated and diagnostic became expensive in term of time and money. To cope with this issue, and because classic methods have demonstrated their inadequacy, scientists invented new solutions. Those solutions are essentially based on Artificial Intelligence (AI) methods such as Expert Systems (ES) and Artificial Neural Networks (ANNs).

The term diagnostic appeared initially in the medical field. It refers to the process of identifying a disease or injury from its signs or symptoms. An accurate diagnostic reposes on several factors such as the timing and the sequence of symptoms, past medical history and recent exposure to the disease. An exact diagnostic allows for prescribing a successful treatment and saves lives.

Human body organs are similar to machines and automate in an industrial process. Thus, the term diagnostic was rapidly spread in the industry, and it consists of the search for causes and nature of a technique failure in a physical apparatus. Conceptually speaking, medical diagnostics and technical diagnostics are basically equivalent. Both refer to finding causes of a malfunctioning in a machine or in human body in order to prescribe the right cure.

More precisely, diagnostic in industrial field consists of detecting the origin of failure and to locate the damaged parts in case of physical failure or to locate the fragment of code causing anomaly in case of software malfunction. In complex systems, where the number of manipulated variables is considerably big, simple mathematical methods become obsolete. In the same context, scientists have developed new methods based on Artificial Intelligence (AI), especially meta-heuristic methods, that showed their efficiency to face complicated situations.

Metaheuristic techniques mimic the greatest qualities found in nature, motivated by natural selection and social adaptation [De León-Aldaco et al. 2015]. The basic principle behind metaheuristic techniques is to manage the search process so that it can efficiently explore the search region while avoiding becoming trapped in a confined location, such as a local optimum scenario.

Metaheuristic methods are classified into two types: trajectory-based methods and population-based methods. The primary distinction between the two methods is the number of tentative solutions employed in each algorithm's iteration. Trajectory-based methods (e.g., hill climbing) starts with one solution which will be replaced by another solution found in the neighborhood at each iteration.

In the other hand, population-based algorithms start with a set of solution initialized randomly. At each iteration, some population members are replaced by newly created individuals (those who show better behavior to solve the problem at hand). [De León-Aldaco et al. 2015]

Swarm Intelligence (SI) algorithms are population-based metaheuristic methods that came for solving optimization problems. Originally, SI was inspired from the collective behavior of insects, birds, animals etc. Particle Swarm Optimization (PSO) mimics social behaviors observed in flocks of birds, packs of wolves, and so on. For instance,

Grey Wolf Optimization (GWO) algorithms are inspired by the hierarchy and social interaction of grey wolves in nature [Mirjalili et al., 2014].

In this work we will present our approach for diagnosing problems encountered in an industrial gas factory. GWO algorithm will be the core of our approach. It consists of acquiring values of multiple variables from several parts in the process, organized in a training dataset, and exploring the relation between them to detect any eventual failure. This will generate a set of classification rules, named classification rules model, which allows instantly control of the whole system behavior.

The proposed solution will be discussed in detail through chapters of the present master dissertation:

The first chapter is an introduction to industrial systems. We will present definitions of some industrial terms related to the objective of this work.

The second chapter will be an abstract definition of data mining and its methods.

In the third chapter we will illustrate the novel proposed version of GWO algorithm for generating classification rules model.

Chapter 1

INDUSTRIAL DIAGNOSTIC

1. Introduction

Automated systems in the industrial processes are known by their increased complexity. This complexity makes them vulnerable to failure which is the major source of important costs essentially in terms of security (accidents risks, pollutions ...) and in terms of availability (decrease in productivity). This vulnerability justifies the introduction of monitoring module in the process. Monitoring consists to evaluate the situation of an installation, to provide information to operators, and to react (in case of automatic control) according to this situation. One of the most important tasks of the monitoring process is diagnostic which refers to the detection and localization of failures in the system [G. Oum Keltoum et al., 2019]. Recently, Artificial Intelligence methods were widely used to deal with the complexity of diagnostic in industrial systems. These methods came to help or to replace field experts in controlling industrial process behavior.

Before to go further in this chapter, some definitions should be presented.

2. Terminology

a) Diagnostic

Cambridge dictionary defines diagnostic as methods or systems for discovering the cause of a problem.

In Wikipedia, diagnosis is the identification of the nature and cause of a certain phenomenon.

The process of identifying a fault, locating its source, and determining its reasons is known as diagnostic.

b) Fault

The International Federation of Automatic Control (IFAC) SAFEPROCESS Technical Committee defines a fault as the unpermitted deviation of at least one characteristic property of a system variable from its normally accepted behavior, leading to the inability of one or more system components to fulfill their intended purposes [Park et al., 2020]. This can be due to a variety of reasons such as incorrect operation, defects in design, or severe conditions. A fault can cause the system to behave abnormally or to stop working.

c) Failure

In industry, failure refers to the permanent nonsuccess of a device to accomplish a required function under specified conditions [Iserman et al., 1997]. It's related to a fault occurred in a part of the device.

d) Malfunction

Malfunction is an occasional interruption of a process execution caused by a fault [Park et al., 2020].

(Fig. 1.1) illustrates the difference between failure and malfunction [Miljković, 2011].

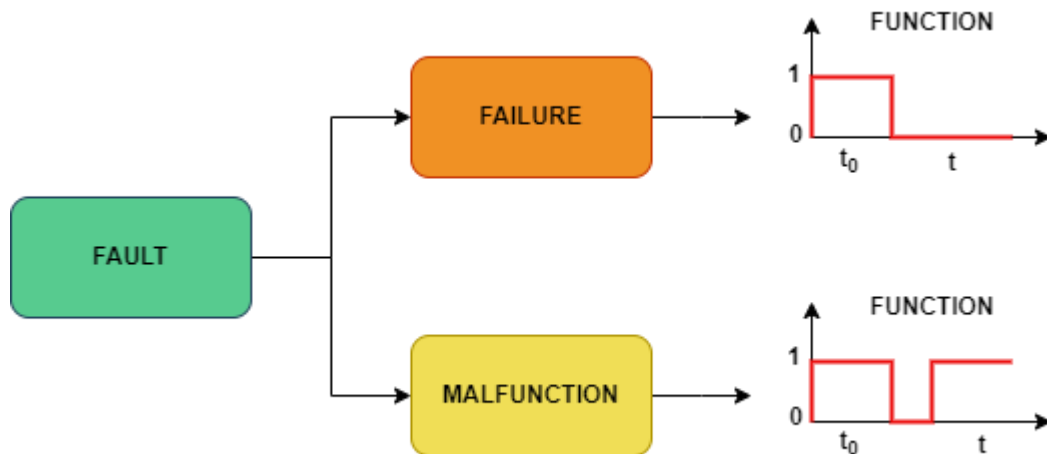


Figure 1.1 Progression of fault toward malfunction or failure [Miljković, 2011].

e) Error

An error is a difference between an output variable's measured value and its theoretically correct value [Iserman et al., 1997].

f) Residual

A residual is a fault indication that is based on a discrepancy between measurements and computations using model equations [Iserman et al., 1997].

g) Symptom

A symptom is a deviation from normal behavior in an observable quantity [Iserman et al., 1997].

h) Model

A model describes the system in its normal condition (fault-free). As a result, in the event of a fault, it is possible to specify discrepancies between the current system behavior and the predicted behavior in a normal condition.

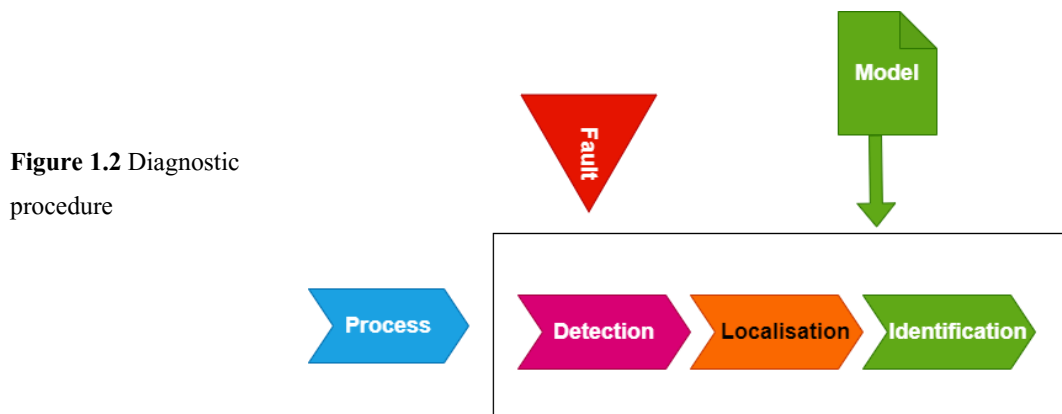
3 Industrial diagnostic

In industry, failure diagnostic is a procedure that aims to detect the type, the size, the localization, and the apparition time of the failure. The fundamental idea of diagnostic is to check whether a given process's situation is coherent with the normal behavior. AFNOR (French Normalization Association) defines an Industrial Diagnostic as: the identification of the probable cause of a failure with logic reasoning based on a set of information gathered from an inspection, control or test.

Two essential tasks in diagnostic process:

- observation of failure symptoms
- cause detection by a logical reasoning based on observations.

Generally, three phases are distinguished in the industrial diagnostics: detection, localization and identification (Fig. 1.2). Fault symptoms are formulated in detection phase. Process faults are pointed out in the isolation phase based on patterns of detected symptoms. [Jan et al., 2004]



Detection

Fault detection is the technique of discovering an occurrence of a fault in a part of the observed process based on measurements that are provided by the system. It is performed by observing dependencies between different measurable signals or variables [Webert et al., 2022].

Localization

After detection of fault, it is essential to find the exact location of the affected component by analyzing the events and tracing all the symptoms.

In the industry, fault localization is important for several reasons, including improving reliability and stability of systems and software, reducing the time and effort required to fix problems, and maintaining customer satisfaction by providing prompt and effective solutions.

4 Diagnostic methods

A plenty of fault detection methods are constantly proposed in literature, however, two classes of methods can be distinguished: methods with application of system models (or internal methods) based on mathematical tools, and methods without the models (or external methods) based essentially on Artificial Intelligence techniques. (Fig. 1.3) demonstrates some of the most common techniques found in the literature.

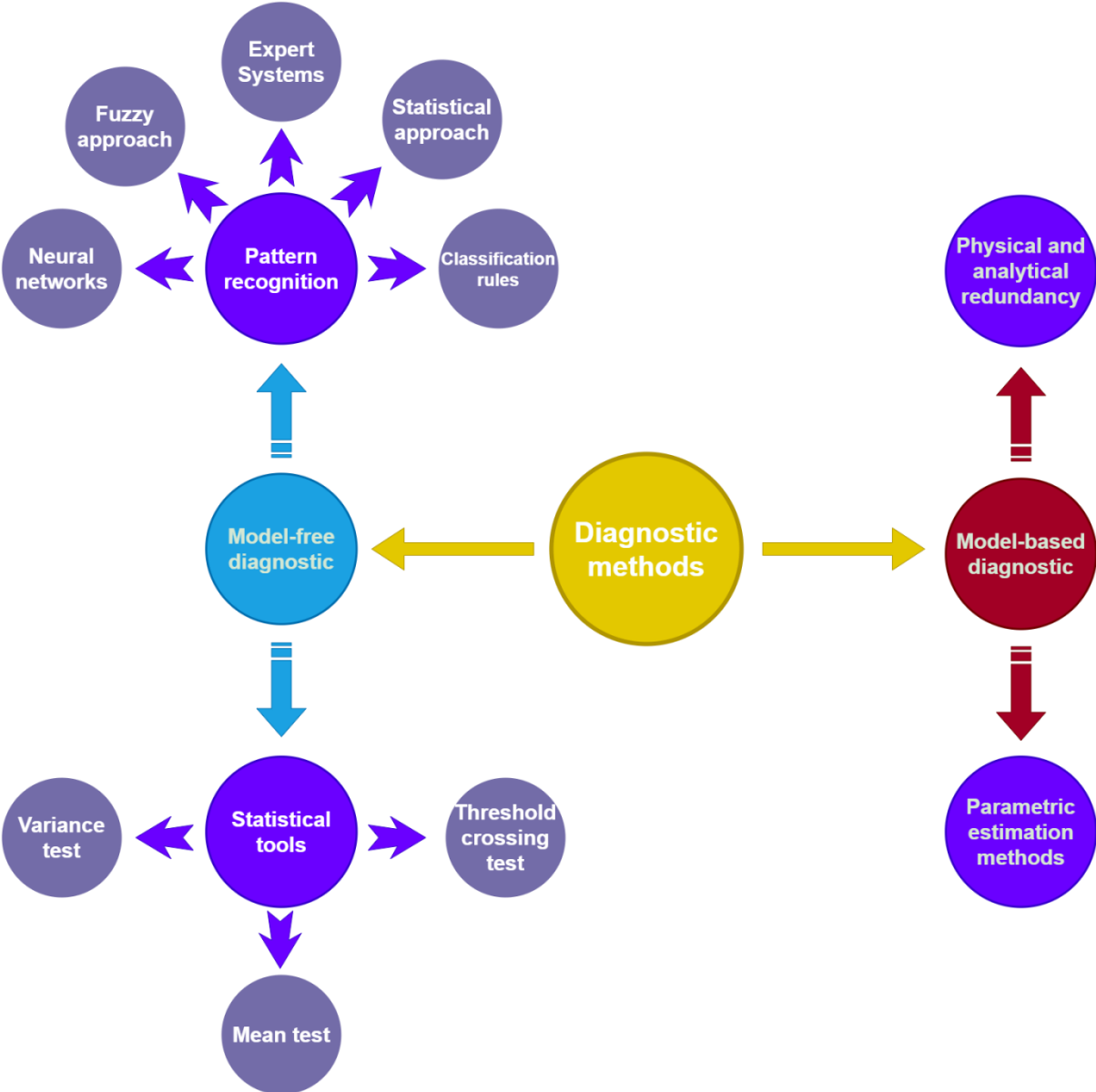


Figure 1.3 Industrial diagnostic methodology classification [Sekiou, 2013]

4.1 Model-based methods

Fault detection consists of model-based residual extraction, residual evaluation, and eventually decision-making processes for classifying identified fault symptoms. Typically, models depict the system in fault-free state (normal state). Thus, it is feasible to identify differences between the current system behavior and what is anticipated in a normal condition in the event that faults occur. Residual can be defined as the discrepancy between a process's variable value and a value determined from the model.

Residual values should be as close to zero as possible in the normal state of the system. As a result, an increase in the residual value may be regarded as a fault sign.

Model-based diagnostic methods are composed essentially of two technics: physical and analytical redundancy methods and parametric estimation methods.

4.1.1 Physical redundancy

It consists of using multiple sensors to measure the same variable to ensure the accuracy of diagnostic data. By having redundant sensors, if one sensor fails or provides inaccurate readings, the other sensors can still provide reliable data.

4.1.2 Analytical redundancy

Analytical redundancy is based on the use of additional information acquired from models rather than sensors, which allows the generation of quantities of the same nature as those obtained from the sensors. This information will be compared under same conditions as in the simple redundancy method by evaluating their coherence. These models are just mathematical connections that relate specific measured data to output quantities. [Sekiou, 2013]

4.1.3 Parameter-estimation methods

After applying knowledge laws, the method uses the structural parameters of a knowledge model to extract system parameters in order to locate and identify faults. The choice of the knowledge model is crucial to the method's effectiveness.

The main idea in parameter-estimation methods is to obtain the relationship between model parameters and system parameters (Fig . 1.4) [Yu, 1997].

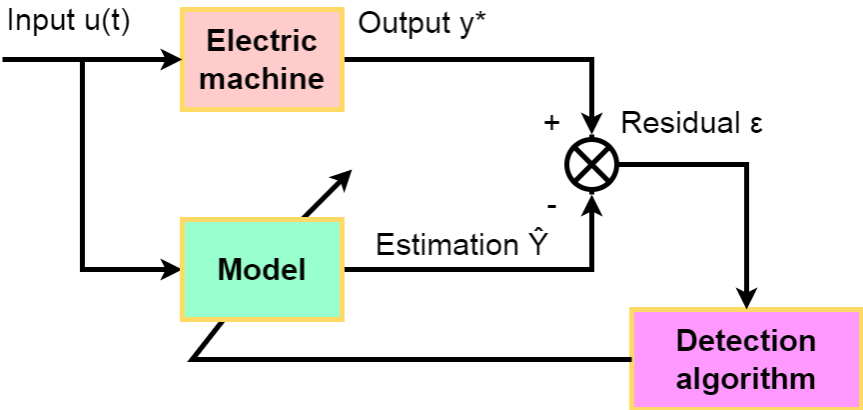


Figure 1.4 Parameter estimation method [Yu, 1997].

4.2 Model-free methods

Model-free diagnostic methods are essentially known under two branches:

- methods based on pattern recognition
- methods based on statistic tools

4.2.1 Pattern recognition methods

These methods are based on Artificial Intelligence technics such as expert systems (ES), artificial neural networks (ANN), fuzzy logic and classification rules.

4.2.1.1 Expert systems

Rule-based expert systems are commonly utilized for diagnostic tasks. The knowledge base and inference engine are the main components of this approach. Knowledge is represented in form of production rules. [Miljković, 2011]

4.2.1.2 Fuzzy approach

Diagnostic methods based on fuzzy approach are implemented considering the imprecision in industrial systems. Thus Operators are given information about the fault severity of the system as the output of fuzzy controllers rather than the basic binary decision of fault or no-fault. Based on the existing expert knowledge and measured data, a rule-based model with linguistic interpretability is developed. [Miljković, 2011]

4.2.1.3 Classification rules

Classification is a supervised learning task that requires dataset containing both predictors and the associated consequents (target class). The predictors in fault diagnosis relate to the features collected from raw signals using signal analysis techniques, and the target class represents the system's conditions, such as healthy, fault, or fault type.

Classification is commonly performed in two steps, named training and testing. First, the dataset is split into two subsets, training set and testing set. The training set is used to extract relationship between the features and the classes using the appropriate algorithm. The main idea in classification rules is to extract the ensemble of features the most significant for fault detection. The result of the first step is a set of rules, called classifier, in the form of (IF condition THEN consequent). Testing set is then used to check the accuracy of the obtained classifier.

4.2.2 Methods based on static tools

In this group of static tools the following methods can be distinguished:

- Threshold crossing test,
- Variance test,
- Mean test.

Conclusion

In this chapter we showed the importance of fault diagnostic in industry. Its necessity come from the complexity of industrial systems due to the presence of automates everywhere in manufacturing levels. Thus, efficient diagnostic strategies have to be installed in order to ensure availability and performance of industrial systems. A plenty number of approaches where proposed by scientists to solve diagnostic problem, some of them are based AI technics. In the next chapter we will present an overview of Datamining, one of AI field's branches.

Chapter 2

DATA MINING

1 Introduction

In recent years, the amount of data generated has increased quickly, due mainly to the increased use of technology in many aspects of life, as well as the expansion of computer storage capacities. Business activities and scientific research were essentially the domains that drove scientists to realize the immense value of these data. However, traditional data processing approaches have demonstrated their inability to deal with the event. That was the issue who led to the emergence of new technologies, such as Data Mining with the main task of extracting valuable information from this heap of data. This chapter aims at making an overview of Data Mining approach, and its techniques and application fields.

2 Definition

Generally talking, mining refers to digging underground for natural resources such as gold or iron etc.

Data mining refers to a collection of techniques developed from various disciplines such as mathematics, statistics, Artificial Intelligence (AI), expert systems and other classical data analytical technologies, with the goal to extract useful information, or "knowledge", from huge quantities of raw data.

Data mining has traditionally been a secondary analysis of data obtained for another purpose. In supermarket data, for example, information is gathered to calculate the bill the client is charged. This saved data can then be analyzed later to discover the transactional patterns of clients. Nevertheless, the field is developing, and in areas like proteomic research and micro array data, big datasets are gathered primarily for finding patterns, relationships and structures [Hand, 2006].

Data mining is often used interchangeably with the term "knowledge discovery from data" (KDD), whereas others see it as a step in the process of knowledge discovery. (Fig. 2.1) [Tarmizi 2019]. The knowledge discovery process is composed of the following iterative steps:

- *Data cleaning*: is the process of correcting or removing inaccurate, corrupted, incorrectly formatted, duplicate, or incomplete data from a dataset.
- *Data integration*: It is the process of compiling different data sources into a single database to facilitate data processing.
- *Data selection*: in this step, the relevant data for the analysis task are extracted from the database.
- *Data transformation*: Data Transformation is defined as the process of transforming data into appropriate form required by mining procedure.
- ***Data mining***.
- *Pattern evaluation*: pattern evaluation is defined as identifying strictly increasing patterns representing knowledge based on given measures. It finds interestingness score of each pattern, and uses summarization and visualization to make data understandable by user.
- *Knowledge representation*: this involves presenting the results in a way that is meaningful and can be used to make decisions.

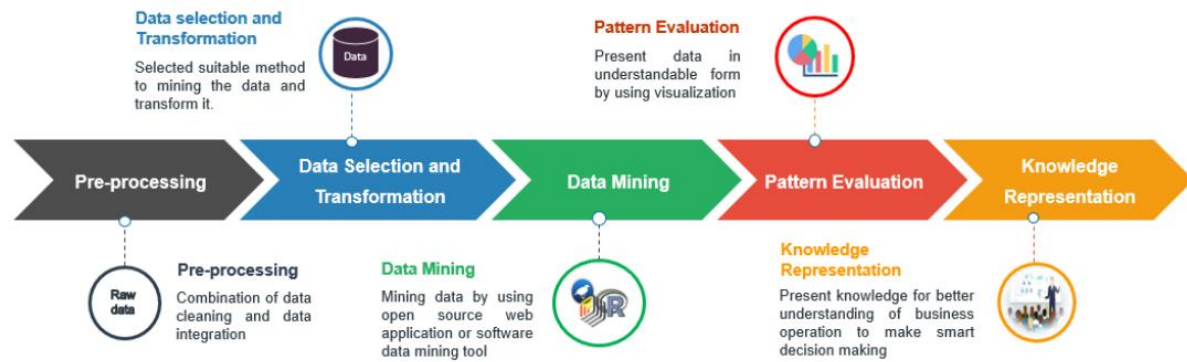


Figure 2.1 Knowledge discovery process [Tarmizi 2019].

3 Data, information and knowledge

Data: data are any facts, numbers, or texts that can be processed by a computer. Today organizations are accumulating vast and growing amounts of data in different formats and data bases.

This includes: operational or transactional data such as sales, costs, inventory, payroll and accounting.

Nonoperational data like industry sales, forecast data, and macroeconomic data [Sumathi, 2006].

Information: the patterns, associations, or relationships among all this data can provide information. For example, retail point-of-sale [Sumathi, 2006]. We can simply define information as: data with a sense.

Knowledge: is an information incorporated in an agent's reasoning resources, and made ready for active use within a decision process; it is the output of a learning process [Aamodt, 1995].

4 Types of data in data mining process

Any type of data can be used for data mining as long as they have relevance for the intended application. Database data, data warehouse data, and transactional data are the most fundamental types of data for mining applications. An overview of most popular data types is presented below:

4.1 Relational database

A relational database is a set of tables, each with a unique name. Each table consists of a set of attributes (columns or fields) and typically stores a large set of tuples (records or rows). Each tuple in a relational table represents an object identified by a unique key and described by a set of attribute values.

4.2 Data warehouse

Data warehouse is a single location for storing data that gathers information from several sources and organizes it into a single plan. Data is cleaned, integrated, loaded, and refreshed before being stored in a data warehouse. A data warehouse divides the data it stores into different categories.

4.3 Transactional data

It represents the documents that are recorded as transactions. These transactions include purchasing from a consumer, buying a flight, visiting a website, and others. There is a distinct ID for each transaction record. Additionally, it lists every component that went into the transaction.

4.4 Object-oriented relational database

Relational-object databases are built on the basis of an object-relational data model. This model extends the relational model by providing a rich data type for complex object management and object orientation. Conceptually, the object-relational data model inherits the essential concepts of object-oriented databases, where, in general terms, each entity is considered an object.

4.5 Other types of data

There are a lot of data types that are known with their structure, semantic meanings, and versatility. They are used in a lot of applications. Here are a few of those data types: data streams, engineering design data, sequence data, graph data, spatial data, multimedia data, and more.

5 Data mining techniques and algorithms (Fig. 2.2)

5.1 Supervised data mining

Supervised data mining is used when users have specific labeled data that needs to predict. It is like machine learning algorithms where it needs supervision on how to learn from example. Supervised mining is a process to find concept descriptions for classes with a pre-classified example. Classification and regression are the two branches of supervised data mining [Tarmizi 2019].

5.1.1 Classification

Classification is a supervised data mining algorithms that mostly used for a process of classifying and predicting values from training dataset based on previous learned classes to build a model that can be used to classify new values [Tarmizi 2019].

Another definition: classification is the process of identifying a model that explains and separates data classes or concepts. The model is created by analyzing a collection of training data, or data objects for which the class labels were already known. The model is used to predict the class label of objects for which the class label is unknown (Fig. 2.3).



Figure 2.2 Data mining techniques and algorithms

There are many algorithms for classification, including the following:

5.1.1.1 Decision Tree (DT)

The decision tree method is one of the most intuitive and popular methods of data mining, especially as it provides explicit rules for classification and supports heterogeneous, missing data and non-linear effects. These trees allow to distinguish different classes and to associate them one or more rules.

In a decision tree, which resembles a flowchart-like tree, each node represents a test on an attribute value, each branch represents the test's result, and the tree's leaves stand in for classes or class distributions.

The main advantage of decision trees in relation to artificial neural networks is that they are easy to understand rules.

5.1.1.2 Neural network (NN)

Neural networks are biologically derived networks that represent in a simplified way the interconnections of the human brain. Artificial neural networks use statistical methods such as probabilities and distributions. However, the results obtained are very low, and the resulting models are difficult to understand and very sensitive to the format of the data processed.

Neural networks are known for their good performances especially for pattern recognition, and for problems including possibly continuous noise variables.

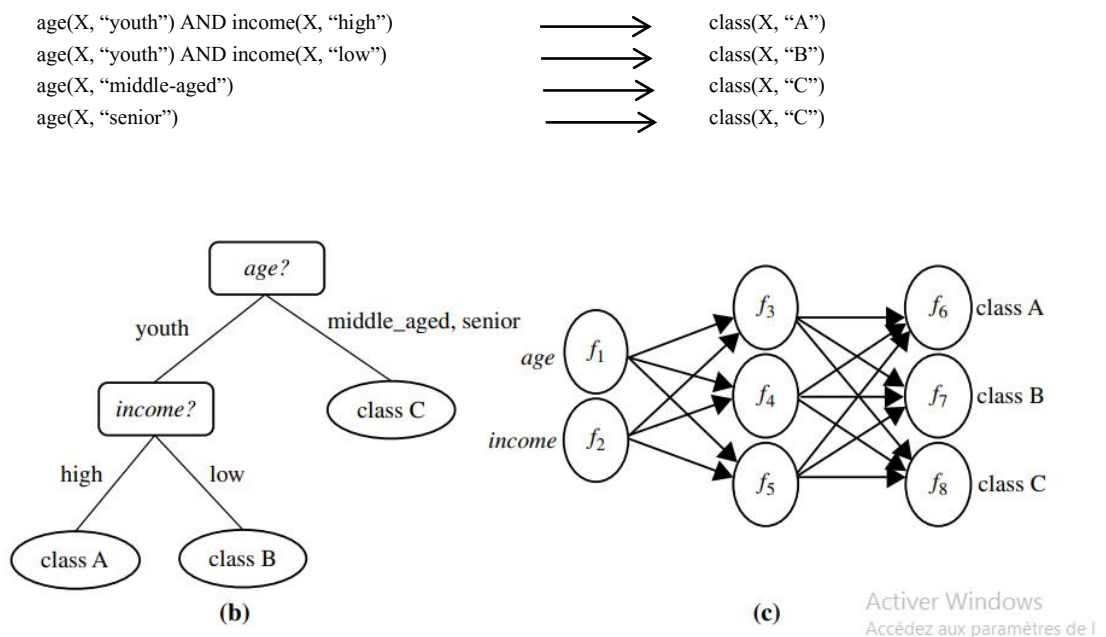


Figure 2.3 a classification model with different representation: a) classification rules, b) decision tree, c) neural network [Han, 2012].

5.1.1.3 Rule-based classification

A rule-based classifier uses a collection of IF-THEN rules for classification as in the expression below:

IF *condition* THEN *result*

The first part of the rule (IF part) is known as "**the antecedent**", while the second part (Then part) is called "**the consequent**".

The antecedent part contains one or more terms, where each term is composed of a variable name, an operator, and a value. In the cases where an antecedent contains more than one term, the terms are joined by the "AND" conjunction. The consequent part of the rule represents the class label associated with the rule. An example of rule-based classifier is illustrated in (Fig. 2.3).

5.1.1.4 Naive Bayes (NB)

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

5.1.1.5 K -Nearest Neighbors (KNN)

The k-means algorithm attempts to find discrete groupings within data, where members of a group are as similar as possible to one another and as different as possible from members of other groups (see the following figure). The user defines the attributes that the algorithm will to determine similarity.

5.1.1.6 Support Vector Machines (SVMs)

An SVM estimates the optimal separating hyper plane by maximizing the margin between hyper-plane and closest points of the classes. SVM is an universal classifier that use an appropriate kernel function to learn the polynomial and radial basic function networks [Tarmizi 2019].

5.1.2 Regression

In general, these methods are used to predict the value of a response (dependent) variable from one or more predictor (independent) variables, where the variables are numeric. There are various forms of regression, such as linear, multiple, weighted, polynomial, nonparametric, and robust (robust methods are useful when errors fail to satisfy normalcy conditions or when the data contain significant outliers)

Regression techniques include:

5.1.2.1 Logistic regression

Is the classifier for regression and it was used to make a prediction on the outcome of predictor variables and to measure the correlation between the categorical variable and dependent variable to change the probability score.

5.1.2.2 Linear regression

Linear regression is a classical method that also known as superior analytics because of ease of interpretation.

5.2 Semi-supervised learning

Semi-supervised data mining is a technique for finding a better classifier by learning from the labeled training dataset and exploits the structural information in available unlabeled data.

5.3 Unsupervised learning

Unsupervised learning is essentially a synonym for clustering. The learning process is unsupervised since the input examples are not class labeled. Typically, we may use clustering to discover classes within the data. For example, an unsupervised learning method can take, as input, a set of images of handwritten digits. Suppose that it finds 10

clusters of data. These clusters may correspond to the 10 distinct digits of 0 to 9, respectively. However, since the training data are not labeled, the learned model cannot tell us the semantic meaning of the clusters found.

Clustering and association are the techniques of unsupervised data mining.

5.3.1 Clustering

Clustering is a method of grouping objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities.

Classifiers for clustering algorithms are K-mean and K-medoids.

5.3.1.1 K-mean

The principle of K-mean algorithm is to split a set of 'M' samples into 'K' disjoint clusters, each is described by the mean of the samples in the clusters. Mean is commonly called as the cluster centroid where centroids are used to minimize the distance.

5.3.1.2 K-medoids

It is usually used to eliminate the distance from the updating calculation by replacing the cluster center with data that called "medoid".

5.3.2 Association

An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective.

5.3.2.1 Apriori

One of association algorithms which is used to find relations between two or more entities in certain data environment such as mathematics and databases.

5.3.2.2 Multilevel

In multilevel association rule, there will be many supports and confidence threshold as there are levels of abstraction except level 0. In this multilevel, support and confidence are called as minimum support and minimum confidence like the concept of hierarchy.

6 Data mining applications

Although we may sometimes be ignorant of it, data mining is a part of most people's daily life. The following are some fields that frequently use data mining:

6.1 Manufacturing

Manufacturing services can use data mining techniques to provide real-time and predictive analytics for overall equipment effectiveness, service levels, product quality, and supply chain efficiency. For example, manufacturers can use historical data to predict the wear of production machinery and anticipate maintenance. As a result, they can optimize production schedules and reduce downtime.

6.2 Sales and marketing

Companies collect a massive amount of data about their customers and prospects. By observing consumer demographics and online user behavior, companies can use data to optimize their marketing campaigns, improving segmentation, cross-sell offers, and customer loyalty programs, yielding higher ROI on marketing efforts. Predictive analyses can also help teams to set expectations with their stakeholders, providing yield estimates from any increases or decreases in marketing investment.

6.3 Banking and insurance

Financial services can use data mining applications to solve complex fraud, compliance, risk management, and customer attrition problems. For example, insurance companies can discover optimal product pricing by comparing past product performance with competitor pricing.

6.4 Telecom, media, and technology

High-competition verticals like telecom, media, and technology use data mining to improve customer service by finding patterns in customer behavior. For example, a company could analyze bandwidth usage patterns and provide customized service upgrades or recommendations.

Conclusion

The amount of digital data has expanded due to the growth of information technology, economics, and communication. Because of these enormous amounts of data, conventional analytical methods are no longer able to handle them.

Since this data is produced in the middle of social networking sites and a variety of institutions like banks and insurance companies, it is necessary to explore this amount to access knowledge that aids in decision-making. As a result, data mining was given tools and algorithms to help explore the enormous and growing amount of data.

In this chapter, we discussed the concept of data mining as well as the reasons of existence, methods, and algorithms. We also attempted to delve into the most well-known area of data mining application, particularly the field of industrial diagnosis, and we demonstrated the potential use of rule-based classification to forecast failure in industrial equipment.

Chapter 3

NOVEL GWO ALGORITHM FOR INDUSTRIAL DIAGNOSTIC

1 Introduction

The rule-based classification method is popular in the data mining community due to the classifier's excellent interpretability. Support Vector Machines (SVM) and Artificial Neural Networks (ANN) are commonly used because to their strength and accuracy, however they are usually criticized for their interpretability limits. As an alternative, the decision rules generated by rule-based classifiers are easily interpretable by decision makers. Rule-based classification is common in fields where the classifier's interpretability by domain experts is critical, such as medicine, genetics, industrial sectors and financial analytics. [Parpinelli et al., 2001] [D. Martens et al., 2007].

A rule in rule-based classifier consists of two parts: antecedent and consequent in the form of "IF-THEN" expression such as in (Fig. 3.1). The antecedent part contains one or more terms, where each term is composed of a variable name, an operator, and a value. When an antecedent has multiple terms, the "AND" conjunction is used to connect them. The rule's class label is represented by the consequent portion of the rule.

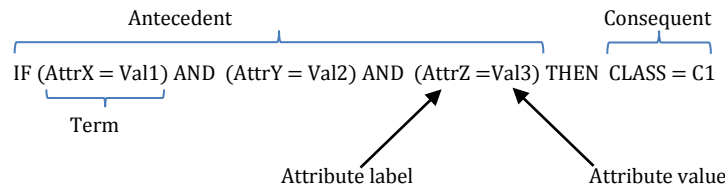


Figure 3.1 generic structure of a classification rule [Hossain et al., 2022]

The primary purpose of the process of developing a classification rules system is to identify a set of features and associated values linked by conjunctions that maximizes objective measurements and accuracy [Hossain et al., 2022]. When the dataset is considerably large, computational work for generating classification rules model can be expensive. To deal with this issue, many meta-heuristic search algorithms were implemented and showed their performance. Among those algorithms resides the Grey Wolf Optimizer algorithm (GWO) which is a swarm intelligence optimization algorithm proposed by Mirjalili in [Mirjalili et al., 2014]. GWO algorithm will be discussed in the next section.

2 Grey Wolf Optimizer algorithm

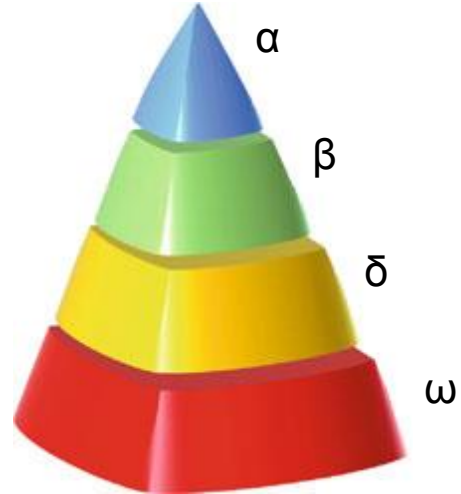
The grey wolf optimizer (GWO), like all other swarm intelligence (SI) algorithms, draws inspiration from the social structure and hunting methods of grey wolves. Grey wolves live in a pack of 5-12 members in average. The pack is organized in a fairly rigorous social dominating hierarchy as depicted in (Fig. 3.2). In the first rank of this hierarchy reside wolves called alphas which are responsible of making decision about hunting, sleeping places, time to wake, and so on [Mirjalili et al., 2020]. All the other wolves should acknowledge alpha's decisions and respect their orders. Strength is not the only characteristic of alpha wolves, managing skills are more important in leading a pack.

Betas wolves occupy the second position. They help alphas in decisions making. They carry out alpha's commands and distribute them to the entire pack, serving as counselors to alpha and pack disciplinarians. A beta wolf is the best candidate to become an alpha in the future. [Mirjalili et al., 2014]

In the third rank reside the delta wolves. They subordinate alphas and betas, but they dominate omegas. This category of wolves includes scouts, sentinels, elders, hunters, and caretakers. Scouts are responsible for keeping an eye on the territory's borders and alerting the pack to potential dangers. Sentinels protect and guarantee the safety of the pack. Elders are experienced wolves who were formerly alphas or betas, and hunters hunt alongside the alphas and betas to supply food for the pack. Finally, the caregivers are in charge of caring for the pack's sick, injured, and weak wolves [Mirjalili et al., 2020].

Omega wolves are the lowest rung of the hierarchy; they serve as scapegoats since they must always surrender to all other dominant wolves, and they are the last wolves allowed to eat.

Figure 3.2 hierarchy of grey wolves
(dominance decrease from top down)
[Mirjalili et al., 2014]



Group hunting is another feature of grey wolves, in addition to the social organization outlined above. The main phases of grey wolf hunting are as follow [Mirjalili et al., 2014]:

- Tracking, chasing, and approaching the prey.
- Pursuing, encircling, and harassing the prey until it stops moving.
- Attack towards the prey.

In order to mathematically model the social hierarchy of wolves for solving any optimization problem, alpha is considered the best option, followed by beta and delta. All other solutions are assumed to be omegas.

The following two operators serve as a representation of the hunting behavior [Mirjalili et al., 2014]:

2-1 Encircling prey

The encircling behavior of grey wolf is modeled by the following equations:

$$\overrightarrow{GWD} = |\overrightarrow{GWC} \cdot \overrightarrow{Pos}_p(it) - \overrightarrow{Pos}(it)| \quad (1)$$

$$\overrightarrow{Pos}(it + 1) = \overrightarrow{Pos}_p(it) - \overrightarrow{GWA} \cdot \overrightarrow{GWD} \quad (2)$$

Where it indicates the current iteration, \overrightarrow{GWA} and \overrightarrow{GWC} are coefficient vectors, \overrightarrow{Pos}_p is the position vector of the prey, \overrightarrow{POS} indicates the position vector of a grey wolf.

The vectors \overrightarrow{GWA} and \overrightarrow{GWC} are calculated as follows:

$$\overrightarrow{GWA} = 2 \vec{c} \cdot \overrightarrow{rand}_1 - \vec{c} \quad (3)$$

$$\overrightarrow{GWC} = 2 \cdot \overrightarrow{rand}_2 \quad (4)$$

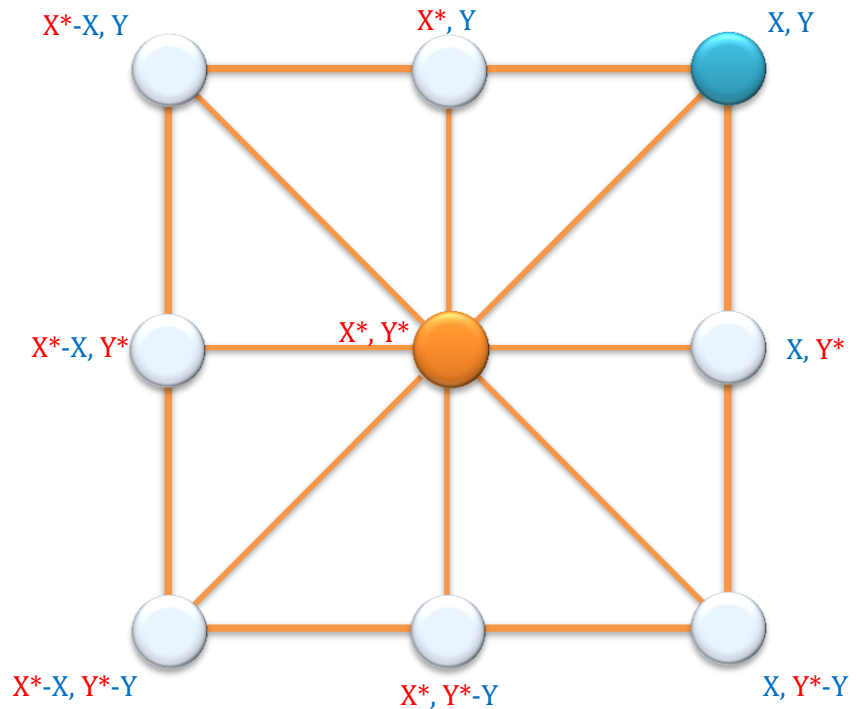
Where components of \vec{c} are linearly decreased from 2 to 0 over iterations as per equation (5), and $\overrightarrow{rand}_1, \overrightarrow{rand}_2$ are random vectors in $[0, 1]$.

$$\vec{c} = 2 - it \left(\frac{2}{Max_it} \right) \quad (5)$$

Where Max_it is the maximum number of iterations.

According to (Fig. 3.3), a grey wolf in position (X, Y) can reach any other position around the prey in 2D space when considering the position of the prey (X^*, Y^*) , by adjusting the values of \overrightarrow{GWA} and \overrightarrow{GWC} vectors. As an example, to get the position of a wolf equal to (X^*-X, Y^*-Y) , \overrightarrow{GWA} should be $(1, 1)$ and \overrightarrow{GWC} should be $(1, 1)$ according to equations (1) and (2). Vectors \overrightarrow{rand}_1 and \overrightarrow{rand}_2 allow wolves to reach any position between points (circles in the figure) presented in (Fig. 3.3). In the same way any dimension space could be considered.

Figure 3.3 The position of a wolf can be defined around a prey with considering a 2D mesh around the prey



2.2 Hunting

According to the above equations a wolf can move to any position in a hyper-sphere around the prey. Since this is not enough to simulate the social intelligence of grey wolves, the best solution found so far (the alpha wolf), should be considered as the position of the prey. This is very essential because in real-life problems, the position of the global optimum is unknown [Mirjalili et al., 2014].

Thus, the position of any wolf in the pack could be updated considering the positions of alpha, beta and delta, as per the following equations:

$$\overrightarrow{Pos}(it + 1) = \frac{\overrightarrow{Pos}_1 + \overrightarrow{Pos}_2 + \overrightarrow{Pos}_3}{3} \quad (6)$$

Where \overrightarrow{Pos}_1 , \overrightarrow{Pos}_2 and \overrightarrow{Pos}_3 are calculated as follow:

$$\begin{aligned} \overrightarrow{Pos}_1 &= \overrightarrow{Pos}_\alpha(it) - \overrightarrow{GWA}_1 \cdot \overrightarrow{GWD}_\alpha \\ \overrightarrow{Pos}_2 &= \overrightarrow{Pos}_\beta(it) - \overrightarrow{GWA}_2 \cdot \overrightarrow{GWD}_\beta \\ \overrightarrow{Pos}_3 &= \overrightarrow{Pos}_\delta(it) - \overrightarrow{GWA}_3 \cdot \overrightarrow{GWD}_\delta \end{aligned} \quad (7)$$

$\overrightarrow{GWD}_\alpha$, $\overrightarrow{GWD}_\beta$ and $\overrightarrow{GWD}_\delta$ are calculated as follow:

$$\begin{aligned} \overrightarrow{GWD}_\alpha &= |\overrightarrow{GWC}_1 \cdot \overrightarrow{Pos}_\alpha - \overrightarrow{Pos}| \\ \overrightarrow{GWD}_\beta &= |\overrightarrow{GWC}_1 \cdot \overrightarrow{Pos}_\beta - \overrightarrow{Pos}| \\ \overrightarrow{GWD}_\delta &= |\overrightarrow{GWC}_1 \cdot \overrightarrow{Pos}_\delta - \overrightarrow{Pos}| \end{aligned} \quad (8)$$

The variation in \overrightarrow{GWA} allows simulating the Exploration and Exploitation process:

2.3 Searching for prey (exploration)

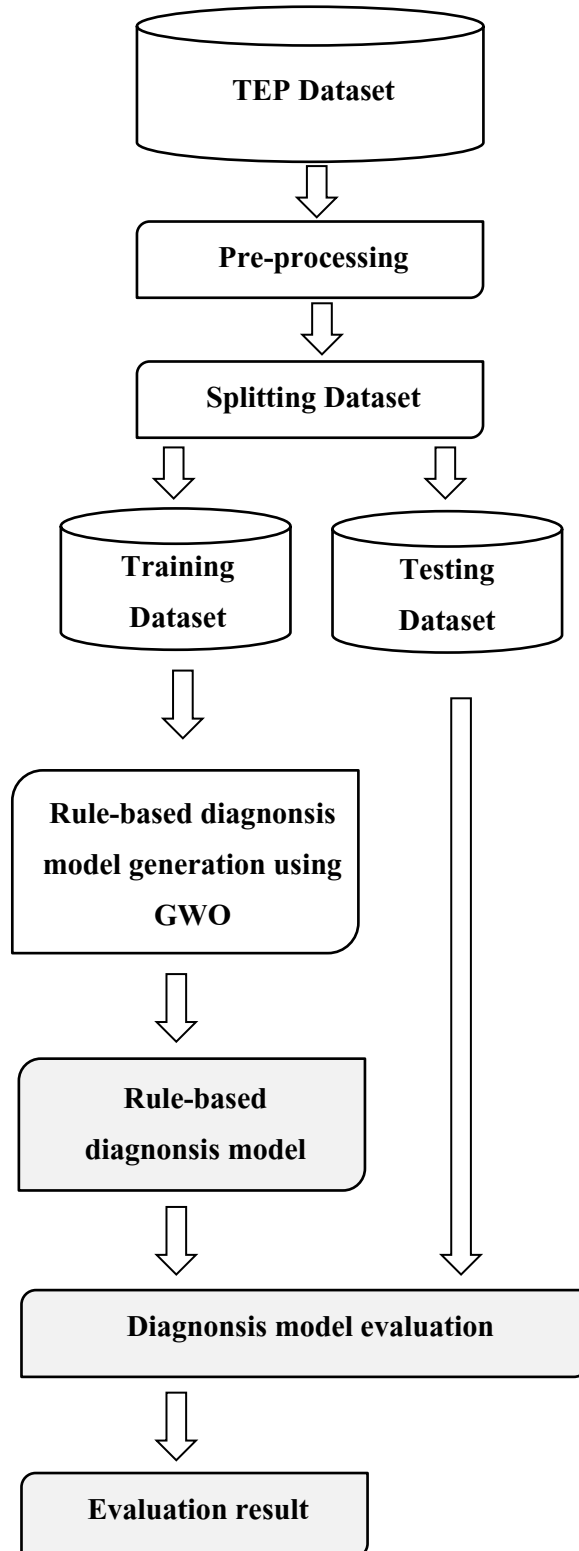
As mentioned above, grey wolves update their positions according to the position of alpha, beta and delta. Searching agents are forced to diverge from their target when \overrightarrow{GWA} values are kept at random less than -1 or greater than 1. In this way GWO search globally and emphasis exploration.

As may be seen in the equation (4), the \overrightarrow{GWC} vector contains random values in the range [0, 2]. This component offers random weights for prey in order to stochastically accentuate ($\overrightarrow{GWC} > 1$) or reduce ($\overrightarrow{GWC} < 1$) the role of prey in defining the distance in the equation (1). This parameter will emphasis exploration and ensure local optima avoidance.

2.4 Attacking the prey (exploitation)

In order to mathematically model attacking the prey the random values of \overrightarrow{GWA} should be in the range $[-1, 1]$, by decreasing the values of \vec{c} . Thus, the updated position of a search agent can be between the current position and the position of the prey.

Figure 3.4 TEP diagnosis model generation methodology.



3 Case study description

In order to prove the efficiency of DiscreteGWO algorithm, we will apply it on a chemical process: the TEP, which will be described further in this chapter. DiscreteGWO will essentially generate a set of rules capable to classify the process into two modes: normal functioning mode or fault # 04 mode. The overall illustration of the TEP diagnosis system is shown in the (Fig. 3.4)

3.1 Abstract of used tools

3.1.1 WEKA

Weka is an open source software developed in the University of Waikato, New Zealand. It is a data mining software that contains a set of machine learning algorithms for data analysis and predictive modeling. WEKA provides tools for data pre-processing, classification, regression, clustering, association rules, and visualization as in the diagram below (Fig. 3.5).

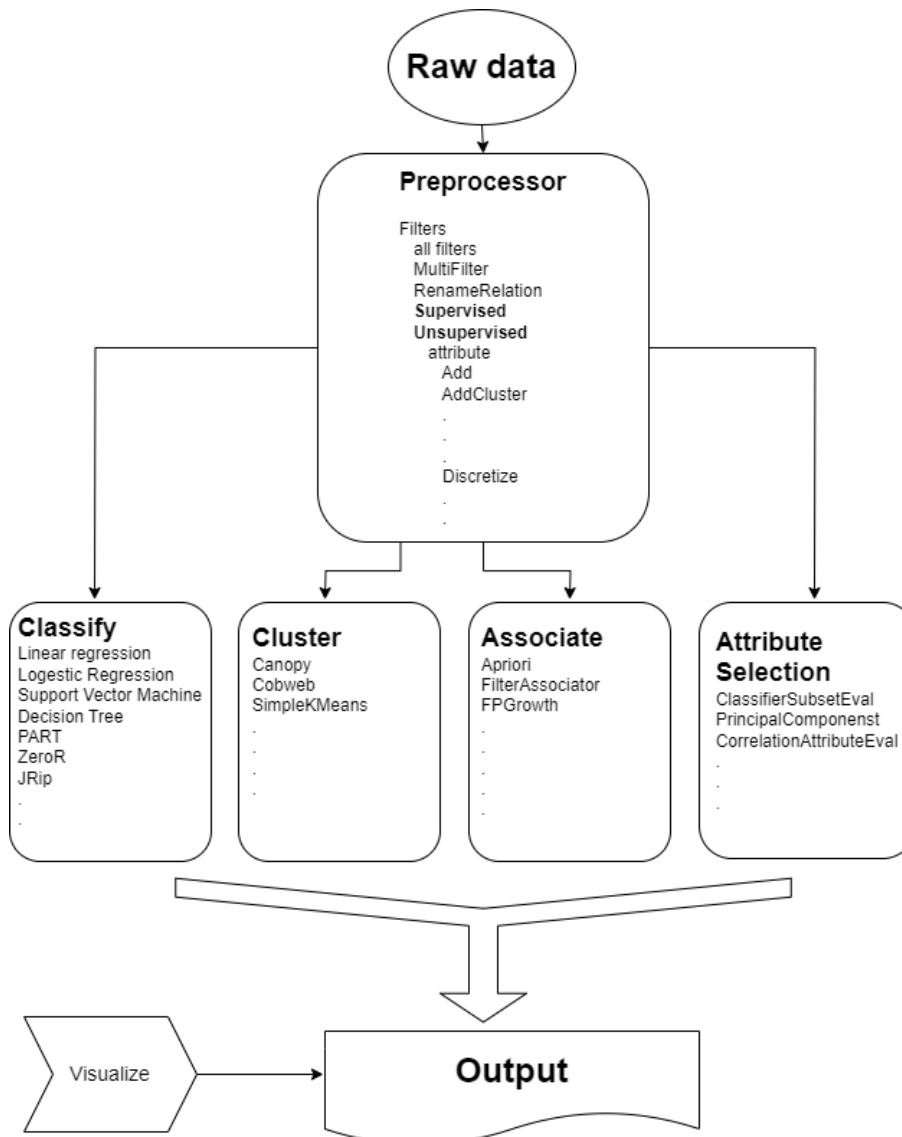


Figure 3.5 WEKA tools.

3.1.2 Python

Python is a programming language released in 1991. It supports procedural, object-oriented and structured programming. Python has a simple syntax that allows programmers to develop applications with few lines. It is also characterized by the following features:

Software quality: Python code is readable, reusable, and maintainable, with uniformity and support for advanced software reuse mechanisms.

Developer productivity: Python code is smaller and runs faster than other languages, boosting programmer productivity.

Program portability: Python is easy to port between platforms and offers multiple options for coding portable graphical user interfaces.

Support libraries: Python has a standard library and can be extended with homegrown libraries and third-party application support software, such as NumPy.

Component integration: Python can be used as a product customization and extension tool, integrating with C and C++ libraries, Java and .NET components, frameworks, devices, and networks.

3.1.3 Python's commonly used libraries

Random: Pseudorandom number generators are implemented in Python for a variety of distributions, including uniform selection, permutations, and sampling. It is basically used to generate random float uniformly in the half-open range $0.00 \leq X < 1.0$.

Pandas: Pandas is a Python package for data exploration, cleaning, analyzing, and manipulation.

NumPy: NumPy provides a multidimensional array object, derived objects, and routines for fast operations on arrays.

Json: is a lightweight data interchange format inspired by JavaScript.

Csv: is the most popular import and export format for databases and spreadsheets, but there are discrepancies between data from different sources. However, the general structure is similar enough to create a single module that can effectively manipulate data without programmer knowledge.

Google Colab: Notebook Colab is an interactive environment for code writing and running it in your browser without configuration, with a free access to the GPU and simplify the sharing.

Variable	Description	Units
XMEAS(1)	A Feed (Stream 1)	Kscmh
XMEAS(2)	D Feed (Stream 2)	kg/hr
XMEAS(3)	E Feed (Stream 3)	kg/hr
XMEAS(4)	Total Feed (Stream 4)	kscmh
XMEAS(5)	Recycle Flow (Stream 8)	kscmh
XMEAS(6)	Reactor Feed Rate (Stream 6)	kscmh
XMEAS(7)	Reactor Pressure	kPa gauge
XMEAS(8)	Reactor Level	%
XMEAS(9)	Reactor Temperature	Deg C
XMEAS(10)	Purge Rate (Stream 9)	Kscmh
XMEAS(11)	Product Sep Temp	Deg C
XMEAS(12)	Product Sep Level	%
XMEAS(13)	Prod Sep Pressure	kPa gauge
XMEAS(14)	Prod Sep Underflow (Stream 10)	m ³ /hr
XMEAS(15)	Stripper Level	%
XMEAS(16)	Stripper Pressure	kPa gauge
XMEAS(17)	Stripper Underflow (Stream 11)	m ³ /hr
XMEAS(18)	Stripper Temperature	Deg C
XMEAS(19)	Stripper Steam Flow	kg/hr
XMEAS(20)	Compressor Work	kW
XMEAS(21)	Reactor Cooling Water Outlet Temp	Deg C
XMEAS(22)	Separator Cooling Water Outlet Temp	Deg C

Table 3.1 Process measurements (3 minutes sampling interval).

Variable	Component	Stream	Sampling Interval (min.)
XMEAS(23)	A	6	6
XMEAS(24)	B	6	6
XMEAS(25)	C	6	6
XMEAS(26)	D	6	6
XMEAS(27)	E	6	6
XMEAS(28)	F	6	6
XMEAS(29)	A	9	6
XMEAS(30)	B	9	6
XMEAS(31)	C	9	6
XMEAS(33)	D	9	6
XMEAS(34)	E	9	6
XMEAS(35)	F	9	6
XMEAS(36)	A	11	15
XMEAS(37)	B	11	15
XMEAS(38)	C	11	15
XMEAS(39)	D	11	15
XMEAS(40)	E	11	15
XMEAS(41)	F	11	15

Table 3.2 Composition measurements.

☒ Units are mole %. Dead time is equal to the sampling interval-

Variable	Description	Units
XMV (1)	D Feed (Stream 2)	Kg/hr
XMV (2)	E Feed (Stream 3)	Kg/hr
XMV (3)	A Feed (Stream 1)	Kscmh
XMV (4)	Total Feed Flow (Stream 4)	Kscmh
XMV (5)	Compressor Recycle Valve	%
XMV (6)	Purge Valve (Stream 9)	%
XMV (7)	Separator Pot Liquid Flow (Stream 10)	m ³ /hr
XMV (8)	Stripper Liquid Product Flow (Stream 11)	m ³ /hr
XMV (9)	Stripper Steam Valve	%
XMV (10)	Reactor Cooling Water Flow	m ³ /hr
XMV (11)	Condenser Cooling Water Flow	m ³ /hr
XMV (12)	Agitator Speed	

Table 3.3 Manipulated variables.

Gaseous reactants (**A, C, D and E**) fed the reactor where they react and form, using a catalyst, G and H gases. A cooling system extracts the majority of the heat produced during this reaction.

The gas issued from the reactor is cooled through a condenser and then fed to a vapor-liquid separator. Through a compressor, the vapor that exits the separator is recycled to the reactor feed. Components from the separator (stream 10) are transferred to a stripper (stream 4) where the remaining reactants are discarded.

Stream 4 is used to strip the remaining reactants from Stream 10, which are combined with the recycle stream via Stream 5. The final products G and H resulting from the last step exit to a unit which is not included in the diagram in (Fig. 3.6) 53 variables are included in the process: 41 measured and 12 manipulated variables (Table 3.1, Table 3.2 and Table 3.3). Measured variables, XMEAS (1) through XMEAS (22), are sampled every 3 minutes. Composition measurements, XMEAS (23) through XMEAS (41), are taken from streams 6, 9, and 11. All the process measurements include Gaussian noise.

3.2.1 TEP faults

Tennessee Eastman process simulation is exposed to 21 different faults (table 3.4). Faults are of different nature: step change as faults 1-7, increase in variability of some variables (faults 8-12) or actuator error such as in the case of a sticking valve. Some of the listed faults are unknown.

As described in Table 3.4, fault F4 is an increasing in temperature of the cooling water in the reactor inlet.

Fault	Description	Type
F1	A/C Feed Ratio	Step
F2	B Composition	Step
F3	D Feed Temperature	Step
F4	Reactor Cooling Water Inlet Temperature	Step
F5	Condenser Cooling Water Inlet Temperature	Step
F6	A Feed Loss	Step
F7	C Header Pressure Loss	Step
F8	A, B, C Feed Composition	Random Variation
F9	D Feed Temperature	Random Variation
F10	C Feed Temperature	Random Variation
F11	Reactor Cooling Water Inlet Temperature	Random Variation
F12	Condenser Cooling Water Inlet Temperature	Random Variation
F13	Reaction Kinetics	Slow Drift
F14	Reactor Cooling Water Valve	Sticking
F15	Condenser Cooling Water Valve	Sticking
F16	Unknown	Unknown
F17	Unknown	Unknown
F18	Unknown	Unknown
F19	Unknown	Unknown
F20	Unknown	Unknown
F21	Unknown	Unknown

Table 3.4 TEP Faults.

3.2.2 Evaluation of DiscreteGWO on TEP process

In this section we will use the DiscreteGWO algorithm to detect faults in the TEP process. TEP data is available on the address <http://web.mit.edu/braatzgroup/>.

Dataset is downloaded and split into training dataset and testing dataset. Training dataset is composed of 480 observations for each fault and 480 observations for normal functioning. Testing dataset, in its turn, contains 960 observations for each fault and 960 observations for normal mode (Table 3.5). It is noted that variable XMV (12) is not considered in datasets since it concerns agitator speed which is constant.

In the present study, only Fault 04 is considered. Instances in the dataset are classified in two classes: Normal Mode class and Fault 04 class.

Before to start data processing with DiscreteGWO algorithm, and because classification rules mining works only on categorical features, datasets were discretized in 04 intervals as shown in (Table 3.6). Weka discretization tools were used to achieve this goal.

Mode	Training data	Testing data
Normal	480	960
Fault1	480	960
Fault2	480	960
...
Fault21	480	960

Table 3.5 TEP dataset.

In the first step, DiscreteGWO algorithm starts by initializing all parameters and a population of wolves (a set of vectors) with random values. In the second step, the algorithm move instances of a chosen class from the training dataset to a new dataset. The fitness function is then used to determine grades of wolves in the pack; the wolf with the first highest fitness function's value is an alpha wolf, the second is a beta wolf, and the third is a gamma wolf.

Then, in the third step, DiscreteGWO algorithm uses positions of the three wolves to update all wolves' positions by applying equations (1) to (8), this step is repeated until the maximum number of iterations is reached. After each iteration the order of wolves is updated according to fitness function's values. At the end of iterations, the alpha wolf vector is returned as a classification rule and all instances covered by this rule are classified and removed from the new dataset.

Steps 2 and 3 are repeated until the new dataset is empty.

A new class is chosen and the process is repeated until the whole dataset is classified.

At the end of processing a set of rules is generated to form our classification rules model.

Figures 3.8 through 3.12 illustrate DiscreteGWO code.

An example of position update is illustrated in (Fig. 3.7) bellow.

Variables	Interval 1	Interval 2	Interval 3	Interval 4
XMEAS(1)	$-\infty$ to 0.128048	0.128048-0.218205	0.218205-0.308362	0.308362 to $+\infty$
XMEAS(2)	$-\infty$ to 3618.95	3618.95-3678.4	3678.4-3737.85	3737.85 to $+\infty$
XMEAS(3)	$-\infty$ to 4448.675	4448.675-4511.15	4511.15-4573.625	4573.625 to $+\infty$
XMEAS(4)	$-\infty$ to 9.2219	9.2219-9.363	9.363-9.5041	9.5041 to $+\infty$
XMEAS(5)	$-\infty$ to 26.58325	26.58325-26.9055	26.9055-27.22775	27.22775 to $+\infty$
XMEAS(6)	$-\infty$ to 41.976	41.976-42.335	42.335-42.694	42.694 to $+\infty$
XMEAS(7)	$-\infty$ to 2698.6	2698.6-2707.8	2707.8-2717	2717 to $+\infty$
XMEAS(8)	$-\infty$ to 74.18425	74.18425-75.0665	75.0665-75.94875	75.94875 to $+\infty$
XMEAS(9)	$-\infty$ to 120.405	120.405-120.47	120.535-inf	_1
XMEAS(10)	$-\infty$ to 0.3148	0.3148-0.33553	0.33553-0.35626	0.35626 to $+\infty$
XMEAS(11)	$-\infty$ to 79.56	79.56-79.95	79.95-80.34	80.34 to $+\infty$
XMEAS(12)	$-\infty$ to 48.178	48.178-49.956	49.956-51.734	51.734 to $+\infty$
XMEAS(13)	$-\infty$ to 2626.8	2626.8-2636.8	2636.8-2646.8	2646.8 to $+\infty$
XMEAS(14)	$-\infty$ to 23.26975	23.26975-25.0665	25.0665-26.86325	26.86325 to $+\infty$
XMEAS(15)	$-\infty$ to 48.458	48.458-50.335	50.335-52.212	52.212 to $+\infty$
XMEAS(16)	$-\infty$ to 3096.525	3096.525-3104.85	3104.85-3113.175	3113.175 to $+\infty$
XMEAS(17)	$-\infty$ to 21.901	21.901-22.99	22.99-24.079	24.079 to $+\infty$
XMEAS(18)	$-\infty$ to 64.97175	64.97175-65.6395	65.6395-66.30725	66.30725 to $+\infty$
XMEAS(19)	$-\infty$ to 213.5975	213.5975-229.715	229.715-245.8325	245.8325 to $+\infty$
XMEAS(20)	$-\infty$ to 338.5175	338.5175-340.785	340.785-343.0525	343.0525 to $+\infty$
XMEAS(21)	$-\infty$ to 94.36525	94.36525-94.5675	94.5675-94.76975	94.76975 to $+\infty$
XMEAS(22)	$-\infty$ to 76.76775	76.76775-77.2165	77.2165-77.66525	77.66525 to $+\infty$
XMEAS(23)	$-\infty$ to 31.76075	31.76075-32.2495	32.2495-32.73825	32.73825 to $+\infty$
XMEAS(24)	$-\infty$ to 8.716125	8.716125-8.88735	8.88735-9.058575	9.058575 to $+\infty$
XMEAS(25)	$-\infty$ to 25.87725	25.87725-26.3555	26.3555-26.83375	26.83375 to $+\infty$
XMEAS(26)	$-\infty$ to 6.74445	6.74445-6.9065	6.9065-7.06855	7.06855 to $+\infty$
XMEAS(27)	$-\infty$ to 18.376	18.376-18.827	18.827-19.278	19.278 to $+\infty$
XMEAS(28)	$-\infty$ to 1.61135	1.61135-1.6534	1.6534-1.69545	1.69545 to $+\infty$
XMEAS(29)	$-\infty$ to 32.4395	32.4395-32.987	32.987-33.5345	33.5345 to $+\infty$
XMEAS(30)	$-\infty$ to 13.641	13.641-13.805	13.805-13.969	13.969 to $+\infty$
XMEAS(31)	$-\infty$ to 23.5615	23.5615-24.111	24.111-24.6605	24.6605 to $+\infty$
XMEAS(33)	$-\infty$ to 1.12354	1.12354-1.28386	1.28386-1.44418	1.44418 to $+\infty$

XMEAS(34)	$-\infty$ to 18.06825	18.06825-18.5635	18.5635-19.05875	19.05875 to $+\infty$
XMEAS(35)	$-\infty$ to 2.21745	2.21745-2.2596	2.2596-2.30175	2.30175 to $+\infty$
XMEAS(36)	$-\infty$ to 4.737325	4.737325-4.84515	4.84515-4.952975	4.952975 to $+\infty$
XMEAS(37)	$-\infty$ to 2.2114	2.2114-2.3103	2.3103-2.4092	2.4092 to $+\infty$
XMEAS(38)	$-\infty$ to 0.007061	0.007061-0.020502	0.020502-0.033944	0.033944 to $+\infty$
XMEAS(39)	$-\infty$ to 0.808505	0.808505-0.83433	0.83433-0.860155	0.860155 to $+\infty$
XMEAS(40)	$-\infty$ to 0.086976	0.086976-0.102301	0.102301-0.117626	0.117626 to $+\infty$
XMEAS(41)	$-\infty$ to 52.858	52.858-53.591	53.591-54.324	54.324 to $+\infty$
XMV (1)	$-\infty$ to 61.97625	61.97625-63.0075	63.0075-64.03875	64.03875 to $+\infty$
XMV (2)	$-\infty$ to 53.13325	53.13325-53.8495	53.8495-54.56575	54.56575 to $+\infty$
XMV (3)	$-\infty$ to 19.1345	19.1345-24.376	24.376-29.6175	29.6175 to $+\infty$
XMV (4)	$-\infty$ to 59.18475	59.18475-61.4645	61.4645-63.74425	63.74425 to $+\infty$
XMV (5)	$-\infty$ to 21.42525	21.42525-22.1855	22.1855-22.94575	22.94575 to $+\infty$
XMV (6)	$-\infty$ to 37.55225	37.55225-39.9355	39.9355-42.31875	42.31875 to $+\infty$
XMV (7)	$-\infty$ to 32.7365	32.7365-37.969	37.969-43.2015	43.2015 to $+\infty$
XMV (8)	$-\infty$ to 42.965	42.965-47.309	47.309-51.653	51.653 to $+\infty$
XMV (9)	$-\infty$ to 43.42075	43.42075-47.4885	47.4885-51.55625	51.55625 to $+\infty$
XMV (10)	$-\infty$ to 41.374	41.374-43.382	43.382-45.39	45.39 to $+\infty$
XMV (11)	$-\infty$ to 15.44525	15.44525-18.0115	18.0115-20.57775	20.57775 to $+\infty$

Table 3.6 TEP dataset' features discretization.

XMEAS(1)	XMEAS(2)	XMEAS(3)	XMEAS(4)	XMEAS(5)	XMEAS(6)	XMV(9)	XMV(10)	XMV(11)	Class
3	3	2	2	2	2	2	1	3	C1
3	3	2	1	4	1	2	2	4	C1
3	3	4	2	2	4	4	2	2	C1
3	3	3	1	2	2	4	1	2	C1
.....
3	2	3	3	3	3	3	3	2	C2
3	2	2	2	2	2	4	3	3	C2

Table 3.7 TEP dataset after discretization

Number of maximum iteration = 45

Number of wolves = 25

Iteration N° 9

$c = 2 - 9 \cdot 2 / 45 = 1.6$

Initial position:

1	1	0	0	1	1	1	0	1	0
4	1	3	2	3	1	2	4	2	4

GWA1:

0,84	0,22	-0,03	-0,75	-0,25	-1,19	0,80	0,77	-0,92	0,14
------	------	-------	-------	-------	-------	------	------	-------	------

GWA2:

0,41	0,02	-0,55	-0,64	-0,07	-0,72	0,3	0,75	0,95	-0,63
------	------	-------	-------	-------	-------	-----	------	------	-------

GWA3:

-1,03	-0,17	-0,07	0,61	0,50	-0,16	-0,08	0,61	0,88	1,07
-------	-------	-------	------	------	-------	-------	------	------	------

GWC1:

0,4	1,82	1,60	1,29	0,15	0,09	1,80	0,78	1,54	0,73
-----	------	------	------	------	------	------	------	------	------

GWC2:

0,1	0,80	1,96	1,62	1,48	0,32	0,04	1,58	0,79	1,40
-----	------	------	------	------	------	------	------	------	------

GWC3:

0,83	0,48	1,27	0,21	1,55	1,05	0,84	1,13	0,20	0,86
------	------	------	------	------	------	------	------	------	------

GWD_α:

3,4	0,82	0,60	0,29	0,85	0,91	0,80	0,22	7,16	3,73
-----	------	------	------	------	------	------	------	------	------

GWD_β:

3,1	0,20	0,96	0,62	0,48	0,68	0,96	0,58	0,21	4,40
-----	------	------	------	------	------	------	------	------	------

GWD_δ:

3,83	0,52	0,27	0,79	0,55	0,05	0,16	0,13	0,80	3,86
------	------	------	------	------	------	------	------	------	------

Pos1:

3,86	1,18	0,98	0,78	0,79	0,08	1,64	1,17	10,59	1,53
------	------	------	------	------	------	------	------	-------	------

Pos2:

2,27	1,00	0,47	0,60	0,97	0,52	1,29	1,44	1,20	1,77
------	------	------	------	------	------	------	------	------	------

Pos3:

2,93	0,91	0,98	1,48	1,27	0,99	0,99	1,08	1,70	5,13
------	------	------	------	------	------	------	------	------	------

New position:

1	0	0	0	0	0	0	0	1	0
3	1	1	1	1	1	1	1	3	3

Figure 3.7 Position update in one iteration.

```

def calculate_newposition(dataset, p, wv):
    global matrix1, matrix2, a, j, temp_vector, temp_vectorI, fit, times

    GWA1 = c*(2 * random.random() - 1)
    GWA2 = c*(2 * random.random() - 1)
    GWA3 = c*(2 * random.random() - 1)
    GWC1 = 2 * random.random()
    GWC2 = 2 * random.random()
    GWC3 = 2 * random.random()

    GWD_alpha = abs(C1 * alpha_vector[j] - matrix2[k][j])
    GWD_beta = abs(C2 * beta_vector[j] - matrix2[k][j])
    GWD_gamma = abs(C3 * gamma_vector[j] - matrix2[k][j])

    POS1 = abs(alpha_vector[j] - A1 * D_alpha)
    POS2 = abs(beta_vector[j] - A2 * D_beta)
    POS3 = abs(gamma_vector[j] - A3 * D_gamma)
    new_position = (X1 + X2 + X3) / 3

    maxval = max(dfc2[:,j])
    if round(new_position % int(maxval)) > 0:
        temp_vector[j] = round(new_position % int(maxval))
    else:
        temp_vector[j] = 1 #random.randint(1,int(maxval))

    r = random.random()
    if r < 1/3:
        temp_vectorI[j] = int(steps(alpha_I[j], GWA1, GWD_alpha) )
    else:
        if r >= 1/3 and r < 2/3:
            temp_vectorI[j] = int(steps(beta_I[j], GWA2, GWD_beta) )
        else:
            temp_vectorI[j] = int(steps(gamma_I[j], GWA3, GWD_gamma) )

    indices = np.argsort(-fitness_matrix, axis=0) Ascendant order of fitness matrix
    alpha_wolf = indices[0]
    beta_wolf = indices[1]
    gamma_wolf = indices[2]
    alpha_vector=matrix2[alpha_wolf] Alpha_wolf = vector with the highest fitness
    beta_vector=matrix2[beta_wolf] Beta_wolf = vector with the 2nd highest fitness
    gamma_vector=matrix2[gamma_wolf] Gamma_wolf = vector with the 3rd highest fitness

    alpha_I = matrix1[alpha_wolf]
    beta_I = matrix1[beta_wolf]
    gamma_I = matrix1[gamma_wolf]

```

Figures 3.8 and 3.9 Position update codes.

```

def calculate_fitness(vector1, vector2, dataset, nb_attr, fit_vector):
    ds = dataset.iloc[:, 0:52]      copy features values only (ignore class value)
    ds = ds.to_numpy()
    num_zeros = vector1.count(0)
    vec2 = vector2
    for i in range(52):
        if vector1[i] == 0:
            vector2[i] = -1
    vec = vector2[0]

    mask = vec != -1
    valid_values = vec[mask]
    valid_matrix = ds[:, mask]
    cor = np.sum(np.all(valid_matrix == valid_values, axis=1))
    count = np.count_nonzero(vec == 0)

    if float(num_zeros) == float(nb_attr):
        cor = 0
    return(cor)

```

replace insignificant values in vector2 with -1

Find the number of correspondences for each particule while ignoring -1 values

```

for k in range(wv):
    fit = [True for x in range(len(dfc1))]
    ICC= calculate_fitness(prtclFS, prtclFV,
        dfc1, 52, fit)
    fit = [True for x in range(len(dfc2))]
    NNCC = calculate_fitness(prtclFS, prtclFV,
        dfc2, 52, fit )
    NAI = len(dfc1)
    f = (NAI) / ((NAI-ICC) * (NAI+NNCC))
    if f > fitness_matrix[k]:
        fitness_matrix[k] = f

    matrix2[k] = temp_vector
    matrix1[k] = temp_vectorI

```

ICC number of correctly covered instances

NNCC number of not correctly covered instances

NAI = class1 dataset length

fitness function

Update particle 'k' with the new generated vector

Figure 3.10, 3.11 wolves' fitness calculation code.

- ☒ The above codes are repeated 'max_it' times. Where 'max_it' is the maximum number of iterations of DiscreteGWO algorithm

```
ruleI = matrix1[alpha_wolf]
rule = matrix2[alpha_wolf]
```

Figure 3.12 New generated rule code.

This code is executed when the maximum number of iterations is reached.

Alpha wolf's vector is the new generated rule.

3.2.3 Experimental results

DiscreteGWO algorithm is trained many times through TEP dataset, adjusting, each time, the number of wolves and the number of maximum iterations (Fig 3.13). This allows the particles to search for the best solution and to get an optimum classification rules model.

Table 3.6 represents the classification rules model obtained after running DiscreteGWO algorithm on TEP dataset with the following parameters:

- A population composed of 42 wolves,
- A maximum iterations number equal to 20

This model is composed of 07 classification rules: 03 rules for Normal functioning and 04 rules for FAULT # 04. Terms' numbers of these rules are in the lower side. The majority of the rules have high coverage indices with maximum accuracies which enhance their interpretability. For instance, rule # 04 means that if the value of the cooling water flow is between 43.382 m³/hr and 45.39 m³/hr then the system got a fault of type FAULT #04. This rule covers 663 instances in the testing dataset with 100% of accuracy.

Some evaluation criteria will be applied to the model to mine its features:

Rule #	Rule	Number of terms	Coverage	Rule's Accuracy
01	if XMV(10) in range -inf-41.374 then Normal mode	1	677/960 (70.5%)	86%
02	if XMEAS(7) in range 2698.6-2707.8 and XMV(10) in range 41.374-43.382 then Normal mode	2	154/960 (16%)	86%
03	if XMV(10) in range 41.374-43.382 then Normal mode	1	125/960 (13%)	85%
04	if XMV(10) in range 43.382-45.39 then FAULT # 4	1	663/960 (69%)	100%
05	if XMEAS(24) in range 8.716125-8.88735 and XMV(9) in range 47.4885-51.55625 and XMV(10) in range 45.39-inf then FAULT # 4	3	29/960 (3%)	100%
06	if XMEAS(4) in range 9.2219-9.363 and XMEAS(5) in range 26.58325-26.9055 and XMEAS(27) in range 18.376-18.827 and XMV(9) in range 47.4885-51.55625 and XMV(10) in range 45.39-inf then FAULT # 4	5	2/960 (0.2%)	100%
07	if XMV(10) in range 45.39-inf then FAULT # 4	1	105/960 (11%)	100%

Table 3.8 TEP ruleset for FAULT # 04 with processing time \approx 4 minutes.

3.2.4 Model evaluation criteria

Classification model evaluation is the measure of its capacity to correctly predict results. To achieve this goal, accuracy, precision and recall scores are widely used.

Accuracy score:

Model accuracy is the ratio of the correctly classified instances to the size of the testing set. Accuracy is calculated using the equation (9) below:

$$Accuracy = \frac{TN+TP}{TP+TN+FP+FN} \quad (9)$$

Precision (specificity) score:

Precision is the ratio of the number of instances classified as negatives to the number of the actually negatives instances. Precision is calculated using the equation (10) below:

$$Precision = \frac{TN}{TN + FP} \quad (10)$$

Recall (sensitivity) score:

Recall is the ratio of the number of instances classified as positive to the number of the actually positive instances. It is calculated as per equation (11):

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

Where:

- TP (true positive) is the number of instances classified by the model as positive and actually positive.
- FP (false positive) is the number of instances classified by the model as positive but actually negative.
- TN (true negative) is the number of instances classified by the model as negative and actually negative.
- FN (false negative) is the number of instances classified by the model as negative but actually positive.

Table 3.8 is a summary of comparison between the model obtained after running DiscreteGWO algorithm, and different classification models, based on the above criteria.

With:

TP = 799

FP = 0

TN = 960

FN = 160

Author	Algorithm	Number of rules	Accuracy	Precision	Recall
WEKA tool	ZeroR		50.54%	/	0.505
WEKA tool	DecisionTable	02	91.75%	0.929	0.917
WEKA tool	JRip	04	92.10%	0.926	0.921
WEKA tool	OneR		91.75%	0.929	0.917
WEKA tool	PART	46	92.63%	0.926	0.926
Our approach	DiscreteGWO	07	91.75%	1.00	0.833

Table 3.9 DiscreteGWO results compared to different models.

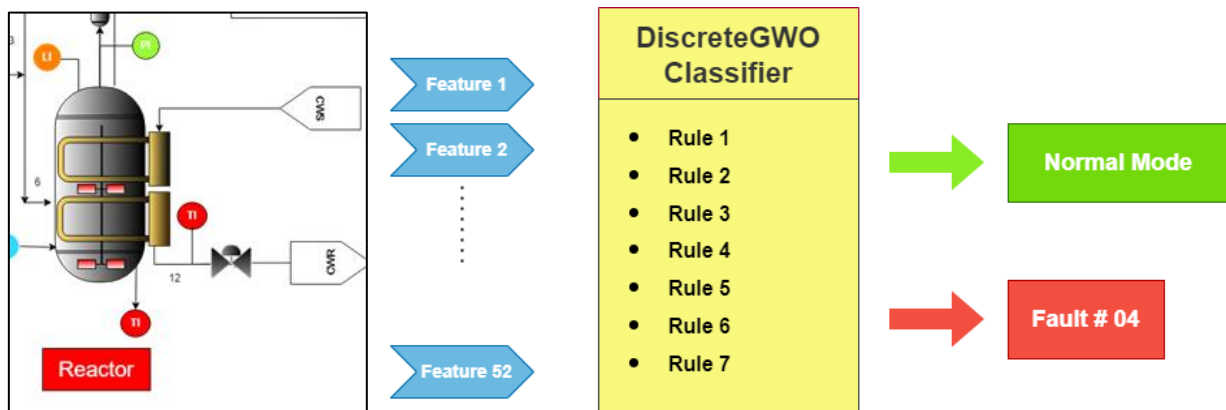


Figure 3.13 Classification Diagram.

3.2.5 Discussion

Results illustrated in table 3.8., prove the performance of this algorithm and its importance in error detection and decision making. The number of rules in the generated model is acceptable, facilitating users' model scanning. The lower number of terms of the rules and their high accuracy scores enhance the interpretability of the model.

In addition to these positives points, the comparison results made in table 3.8., shows that the proposed approach have a good position among the other classification rules generation methods, and could be the topic of further research in the future.

4 Conclusion

In this chapter we presented our approach to deal with the problem of the industrial diagnostic, and more precisely with fault detection. We implemented a program based on swarm intelligence to generate classification model to classify an industrial system behavior into two modes: normal functioning or Fault#04. Since classification is a discrete problem, Grey Wolf Optimization algorithm (GWO) was improved to generate a set of classification rules able to detect any potential anomaly in the system. MODULO and ROUND functions were used to form new equations to get discrete values when calculating particles (wolves) new positions.

General conclusion

The primary goal of the present study was to model, simulate, and implement an industrial diagnostic system for a manufacturing process, Tennessee Eastman Process (TEP), using data mining and swarm intelligence techniques. After providing an overview of diagnostic systems, methods, and techniques, we wrote an entire chapter on the concepts of industrial data failure bases. In the second chapter we presented an overview of data mining techniques and algorithms and their potential to tackle classification difficulties. In the third chapter we made a focus on our work, as it aims to classify industrial modes using swarm intelligence algorithms.

To implement our approach we first downloaded dataset from TEP website. We used WEKA tools to discretize our dataset into 04 bins (categories). This step is primordial because classification problem is a discrete one. Discretized dataset was split to training set and testing set. Training set was, then, processed with the novel GWO algorithm, called DiscreteGWO algorithm, to generate a classifier composed of a collection of classification rules.

The evaluation of the generated model gave motivating results and shown the importance of DiscreteGWO algorithm to solve optimization problems. Stochastic mechanism in DiscreteGWO algorithm allowed to search over all the search space and to avoid local optima.

Our future goal is to improve the DiscreteGWO algorithm do deal with more faults in TEP process and generalize it to other industrial processes.

Bibliography

[Aamodt et al. 1995]: Aamodt, Agnar, and Mads Nygård. "Different roles and mutual dependencies of data, information, and knowledge—An AI perspective on their integration." *Data & Knowledge Engineering* 16.3 (1995): 191-222.

[Chiang et al., 2000]: Chiang, Leo H., Evan L. Russell, and Richard D. Braatz. *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.

[Cimpoeşu et al., 2012]: Cimpoeşu, Elena M., Bogdan D. Ciubotaru, and Dan Ştefănoiu. "Short investigation on system diagnosis regarding parameter estimation." *2012 1st International Conference on Systems and Computer Science (ICSCS)*. IEEE, 2012.

[D. Martens et al., 2007]: D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, "Classification with ant colony optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 651–665, 2007.

[De León-Aldaco et al. 2015]: De León-Aldaco, Susana Estefany, Hugo Calleja, and Jesús Aguayo Alquicira. "Metaheuristic optimization methods applied to power converters: A review." *IEEE Transactions on Power Electronics* 30.12 (2015): 6791-6803.

[Downs et al., 1993]: Downs, James J., and Ernest F. Vogel. "A plant-wide industrial process control problem." *Computers & chemical engineering* 17.3 (1993): 245-255.

[Emary et al., 2016]: Emary, Eid, Hossam M. Zawbaa, and Aboul Ella Hassanien. "Binary grey wolf optimization approaches for feature selection." *Neurocomputing* 172 (2016): 371-381.

[G. Oum Keltoum et al., 2019]: G. Oum Keltoum, T. Oum El'kheir, "Modélisation et Diagnostic des Systèmes Incertains par Analyse en Composantes Principales", master thesis, 2018/2019, KASDI MERBAH University–Ouargla.

[Han 2012]: Han, Jiawei, Micheline Kamber, and Jian Pei. "Data mining concepts and techniques third edition." *University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University* (2012).

[Hand 2007]: Hand, David J. "Principles of data mining." *Drug safety* 30 (2007): 621-622.

[Hossain et al., 2022]: Hossain, Sayed Kaes Maruf, Sajia Afrin Ema, and Hansuk Sohn. "Rule-based classification based on ant colony optimization: a comprehensive review." *Applied Computational Intelligence and Soft Computing* 2022 (2022).

[Iserman et al., 1997]: Isermann, Rolf, and Peter Balle. "Trends in the application of model-based fault detection and diagnosis of technical processes." *Control engineering practice* 5.5 (1997): 709-719.

- [Jan et al., 2004]: Jan Maciej Kościelny, Michał Bartyś "Models in diagnostics of industrial processes", article, January 2004
- [Kościelny et al., 2004]: Kościelny, Jan Maciej, and Michał Bartyś. "Models in diagnostics of industrial processes." *Diagnostyka* 30 (2004).
- [Lutz 2009] : Lutz, Mark. 2009. *Learning Python*. Fourth Edition. Sebastopol, California : O'Reilly Media, 2009.
- [Malik et al., 2022]: Malik, Mohamed Mahdi, and Hichem Haouassi. "Efficient sequential covering strategy for classification rules mining using a discrete equilibrium optimization algorithm." *Journal of King Saud University-Computer and Information Sciences* 34.9 (2022): 7559-7569.
- [Miljković, 2011]: Miljković, Dubravko. "Fault detection methods: A literature survey." *2011 Proceedings of the 34th international convention MIPRO*. IEEE, 2011.
- [Mirjalili et al., 2014]: Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." *Advances in engineering software* 69 (2014): 46-61.
- [Mirjalili et al., 2020]: Mirjalili, Seyedali, et al. "Grey wolf optimizer: theory, literature review, and application in computational fluid dynamics problems." *Nature-inspired optimizers: Theories, literature reviews and applications* (2020): 87-105.
- [P.Rippol, 1999]: P.Rippol. Conception d'un système de diagnostic flou appliqué au moteur automobile. PHD thesis, Savoie university, 1999
- [Park et al., 2020]: Park, You-Jin, Shu-Kai S. Fan, and Chia-Yu Hsu. "A review on fault detection and process diagnostics in industrial processes." *Processes* 8.9 (2020): 1123.
- [Parpinelli et al., 2001]: Parpinelli, Rafael S., Heitor S. Lopes, and Alex A. Freitas. "An ant colony based system for data mining: applications to medical data." *Proceedings of the 3rd annual conference on genetic and evolutionary computation*. 2001.
- [Sekiou, 2013]: S. Sekiou. " Diagnostic des Défaillances des Systèmes Instrumentés de Sécurité : Simulation et Etude Expérimentale". Magister Thesis, 2013, El-Hadj Lakhdar University -Batna.
- [Sumathi et al. 2006]: Sumathi, Sai, and S. N. Sivanandam. *Introduction to data mining and its applications*. Vol. 29. Springer, 2006.
- [Tarmizi et al. 2019]: Tarmizi, Syaidatus Syahira Ahmad, et al. 2019. "A review on student attrition in higher education using big data analytics and data mining techniques." *International Journal of Modern Education and Computer Science* 10.8 (2019): 1.
- [Webert et al., 2022]: Webert, Heiko, et al. "Fault handling in industry 4.0: definition, process and applications." *Sensors* 22.6 (2022): 2205.
- [Yu, 1997]: Yu, D. "Fault diagnosis for a hydraulic drive system using a parameter-estimation method." *Control Engineering Practice* 5.9 (1997): 1283-1291.