A COMPLETE ONLINE SVM AND CASE BASE REASONING IN PIPE DEFECT DECTION WITH MULTISENSORY INSPECTION GAUGE

Van-Khoa D. Le

Faculty of Science and Engineering

University Nottingham

This dissertation is submitted for the degree of

Doctor of Philosophy

School of Computer Science June 2022

I would like to dedicate this thesis to my loving parents . . .

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Van-Khoa D. Le

June 2022

Acknowledgements

I am indebted to my supervisor, Dr. Zhiyuan Chen, for her endless support throughout my work. Her superb assistance and invaluable guidance leading to the writing of this work. I am deeply grateful that you took me on as a student and continued to have faith in me over the years. I am extremely grateful to Dr. Yee Wan Wong, former Faculty of Science and Engineering, for her valuable suggestions and motivating guidance.

I gratefully recognize the help from all of my lab mates, Tuong, Ijaz, Moataz, Amer, Kasra, Mahmoud, Sheena, Weixiang through their review and discussion on my work. I am fortunate to have been a part of room BB41e. Thank you for helping me to keep a balance between the life outside and the exhausted lab life.

Most importantly, I am grateful for my family's unconditional, unequivocal, and loving support.

Abstract

An in-line inspection (ILI) robot has been considered an inevitable requirement to perform non-destructive testing methods efficiently and economically. The detection of flaws that could lead to leakages in buried concrete pipes has been a great concern to the oil and gas industry and water resource-based industry. The major problem is the difficulty in modeling the detection of cracks due to their irregularity and randomness that cannot be easily detected. Consequently, the use of an advanced modality system has emerged. Common defects detection systems favor non-destructive testing methods, which utilize specific sensory data. Only a few systems focus on fusing different types of sensory data. Moreover, the decision mechanism in this system required heavy-power consumption sensors with the configuration from the expertise domain. In addition, the outcome of the decision system is a consequence of rule-based settings rather than a mixture of learned features. This work covers the study of defect detection of non-destructive testing methods using fusion inspection sensors, light detection and ranging (LiDAR), and Optic sensors. The studies on ILI robots are reviewed to construct an efficient gauge. The prototype robot has been designed and successfully operated in a lab-scale environment.

Ultimately, the study proposed a replacement for the standard expert system - in the branch of the CBR system, which is the crucial contribution of this thesis. Recent developments in Case-based Reasoning systems (CBR) have led to an interest in favoring machine learning (ML) approaches to replace traditional weighted distance methods. However, valuable information obtained through a training process was relinquished as transferring to other phases. As a result, the complete SVM-CBR system in this thesis concentrates on solving this gap by presenting an effective transferring mechanism from phase to phase. This thesis proposed a full pipeline integration of CBR using the kernel method designated with support vector machine. SVM technique is the primary classification engine for the combined sensory data. Since the system requires a learning SVM model to be invoked in every phase, the online learning mechanism is nominated to update the model when a new case adjoins effectively. The proposed full SVM-CBR integration has been successfully built into a pipe defect detection. The achieved result indicates a substantial improvement in transferring learning information accurately.

List of Publications

Journal Papers

1. "A complete online-SVM pipeline for case-based reasoning system: a study on pipe defect detection system", D. Van-Khoa Le, Zhiyuan Chen, YeeWan Wong, Dino Isa, *Soft Computing* 2020

2 "Multi-sensors in-line inspection robot for pipe flaws detection", D. Van-Khoa Le, Zhiyuan Chen, R Rajkumar, *IET Science, Measurement and Technology* 2019

Conference Papers

1. Incremental learning with online SVMs on LiDAR sensory data, D. Van-Khoa Le, Zhiyuan Chen, YeeWan Wong, *International Conference on Digital Image and Signal Processing (DISP 2019)* 2019

Table of contents

List of figures

List of tables

List of Abbreviations

CBR Case-Based Reasoning 7

CNN Convolutional Neural Network 21

DT Decision Tree 8

EC Eddy Current 2

ER Electrical Resistivity 22

ERM Empirical Risk Minimization 23

GPR Ground-Penetrating Radar 22

HOG Histogram of Oriented Gradient 20

ILI In-line Inpsection 2

IMU Inertial Measurement Unit 40

kNN k-Nearest Neighbor 7

MFL Magnetic Flux Leakage 2

MTC Multi-Trotter Crawler 5

NDT	Non-Destructive Testing	1
PIG	Pipe Inspection Gauge	4
QP	Quadratic Programming	30
RBS	Rule-Based Reasoning	30
RPi	Raspberry Pi	37
SBC	Single-Board Computer	37
SMFL	SpirALL Magnetic Flux Leakage	3
SMO	Sequential Minimal Optimization	29
SRM	Structural Risk Minimization	23
SVM	Support Vector Machine	8
VC	Vapnik-Chervonenkis	22

Chapter 1

Overview of the study

Energy is the crucial key to the development of the industry. With the increase in energy demand, the related infrastructure to support the production of energy also enlarges. In this network, the energy transmission pipelines are considered the vessels of the industry. These systems are predominantly used to transport natural gas, water, and critical fossil resources over long distances and even across countries. Thousands of kilometers of power pipeline within developed and developing countries implies the reliability and feasibility of using a channel for power transmission. Consequently, maintaining a healthy pipeline is a critical objective. In practice, pipelines are vulnerable to the hazards such as metal loss, pitting, and cracks. The defects of the pipeline could result in substantial economic losses and environmental damage. Therefore, the research on pipeline inspection and monitoring for condition-based attracts lots of attention. In addition, a complete integrity management system for pipeline defects also fascinates many researchers.

The study on in-line pipeline inspection has been recognized over the years. Trending studies allocate on the Non-Destructive Testing (NDT) method, which refers to the testing without damaging or affecting the performance of the tested object. This approach is favored for pipeline discontinuity detection, and safety evaluation [1]. Conventional NDT methods are categorized based on the selection of inspection sensors. These solutions include radiographic testing, penetrant testing, ultrasonic testing, visual testing, Eddy Current (EC) testing, and magnetic particle testing Magnetic Flux Leakage (MFL) . Though being recognized as effective strategies, the result of those tests can be different when applied in the same pipeline. Depending on the pipe characteristics, testing methods with applicable principles are selected. In addition, various causes of defects also cause different types of damage. Therefore, appropriate detection methods according to the specific inspection purpose need to be reviewed. In order to operate an inspection mission, specific gadgets need to be designed. The design of inspecting equipment is usually limited by the detection principle and pipeline structure such as size, shape, material, etc. In addition, current pipeline integrity evaluation and defect detection system methods employ these historical data to identify and evaluate high-risk pipeline, while condition-based maintenance can be carried out. Hence, this work systematically introduce In-line Inpsection (ILI) of pipelines in association with robot-based instrumentation and the enhancement of defect detection system.

1.1 Problem Statement

Transportation of fluid from the point of production to different projections for end-users has been a necessity, and hence the importance of the construction of pipelines [2]. Buried pipelines are constructed to protect the pipes from environmental and atmospheric influence. The pipe cracks flaw could be a result of weather changes, prolonged use of the pipe (most pipes are designed to last for twenty-five years irrespective of what flows inside them), intentional and unintentional third party damage such as accident, sabotage, terror and theft [3]. Therefore, the oil and gas industries and other fluid transport-based industries tend to pay attention to leakage detection.

The detection of flaws that could lead to leakages in buried concrete pipes has been an area of great concern in the oil and gas industry and water resource-based industry. Since the occurrence of pipe cracks is irregular and random, building an effective defects detection system is a challenge [4]. The most common ways at present are time-consuming and labor intensive as it involves insulation removal and insulation replacement. This can also require that the equipment is shut down, adding an additional economic burden to plant maintenance and operation. Consequently, the cost to monitor and maintain these pipelines increases substantially, which leads to the necessity of using an automating process [5]. Due to various causes from different sources [6], the crack inside the buried pipeline can be classified into six categories: 1) Transverse inside cracks perpendicular to the pipe axis; 2) Longitudinal inside cracks parallel to the pipe axis; 3) Slant inside cracks at an angle to the pipe axis; 4) Gaping inside cracks remain open; 5) Outside Surface cracks open on the outside; 6) Subsurface cracks do not show on the surface.

Automation has widely been used in industries to enhance capability. The research of robotics solving the aid of automation machines in order to assist and replace the task of humans in a risky and sensitive environment. The studies on robotics design have been advanced recently. New robot models concentrate on optimizing the performance in specific environments. To operate in practical tasks as in a pipeline environment, the design has to be specialized.

1.1.1 The current problem of ILI Gauge

Recently, as a result of competing in inspection instruments companies, the research and development of inspection gauge and equipment of ILI has gain lots of momentum. The SpirALL Magnetic Flux Leakage (SMFL) was presented by T.D. Williamson in 2011. This inspection gauge incorporates the advantages of the SMFL structure and is also able to integrate with the uniaxial magnetic field. Other company also compiles different inspection methods in a single ILI gauge. The design of the equipment introduced by the Rosen company has the capability to produce MFL and EC testing. This combination is recognized to improve the measurement performance of thick-walled pipelines. The design adopts the advantage of scanning EC in abnormal metal loss of the inner tube to maintain high accuracy with the aid of comprehensive geometric inspection information. In addition, the use of simultaneously MFL reveals information on the mid-wall and exterior features. The use of different sensors with different principles for inspection is well-established in the industry. Those ILI gauges present a robust inspection performance with high sensitivity and high precision.

ILI tools have been acknowledged to be reliable and accurate in different circumstances throughout the past decades [7]. Despite the high accuracy of the above ILI gauges, the design needs to be pluggable to successfully scan the pipe. Consequently, these types of gauges are limited to some specific pipeline system. Applicable pipeline for these PIG gauges requires the launchers and receivers. On the other hand, some pipelines are un-piggable due to wear or damage affecting their pigging capacity. Furthermore, bends, external damage to the pipe, the build-up of solids on the pipe bore, and changes or external damage in pipeline cross-section result un-piggable state. Literally, nearly half of the world's petroleum or natural gas pipelines have been built as "un-piggable" [8].

In-pipe robots have a long history of development and, according to movement patterns, can be classified into wheel type, track type, walking type screw-type, and inchworm type [9, 10]. Compared with ground robots, the most significant difference between in-line robots is that their task space is the pipe, which limits the operation in a less space environment. Oil and gas pipelines distribute at least hundreds of kilometers in a three-dimensional space in various routes, including vertical, horizontal branching, and elbow shapes. Because of energy limitations, existing in-pipe robots are all attached to an electrical cable, which restraints the performance in a long-distance and in-service pipe. In practical, in-pipe robots can only operate in a several kilometers pipeline, which often is used to inspect some un-piggable short pipeline. In-pipe Pipe Inspection Gauge (PIG) systems are principally compiled of the mechanical system, and inspection system, where the robotic system controls the locomotion and the inspection system scans the pipe flaw.

1.1.2 Current issues of Inspection Technology

The inspection system can be integrated with different techniques, using various sensors like a visual, laser, sonar, and other NDT sensors. A design of un-piggable inspection, which refers to non-fully cover pipeline inspection gauge, method composed of a special robotic unit, which is based on a Multi-Trotter Crawler (MTC) combined with a bidirectional MFL inspection module[11]. Up to now, CCTV is still the most favorable in-pipe robotic method. The vision approach is usually consists of illumination and lighting system, imaging sensors and cameras, digital camera interfaces and computation units [12].

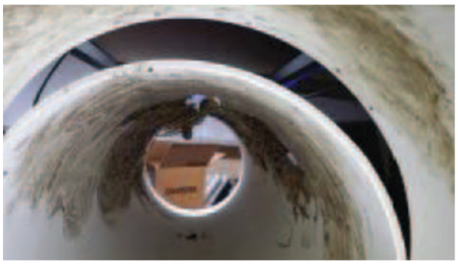


Fig. 1.1 A typical CCTV image capturing pipe health in lab-based environment

The technique is built on the principle of light reflection, light intensity, and absorption. In term of reflection, the sensor emits a beam of light from the sender source and capture data at the receiver. The data is composed of the intensity and the traveling duration. A camera or imaging sensor captures the reflected information for further analysis. Due to the difference in light traveling speed, the selection of the light source is built upon the goals of detection, the scale of the scanning surface and requirements of wavelength and brightness. The research of optical solution in ILI categorized image sensor output into two types, charge-coupled apparatus, and complementary metal-oxide-semiconductor. A camera with various built-in options is necessary when constructing a vision system. Standard interfaces are designed with capture boards (frame grabber), USB, FireWire, GigE, and Camera. Link.

Image captured from the inspection gauge is sent to the analyzing system for further investigation. In emerging research, a study proposed an intensity-based optical system for internal pipe inspection [1]Near-infrared reflectography and infrared thermography were also used for NDT in [13]. In other approaches, to enhance the quality of the image, an additional laser profiler was wielded besides the common CCTV [14]. The system in [15] achieved high-quality accuracy when establishing a computer vision to characterize the nature of the scanning surface. However, the challenge with the optical inspection system requires an appropriate and robust image processing algorithm. Despite the development of recent computer vision, robust image processing algorithms are still the challenges of optical and vision inspection system, especially in a typical environment like the pipeline.

1.1.3 Current issues of ILI inspection system

Many factors can affect the failure models and mechanisms of the ILI inspection system. Various non-destructive evaluation technologies have advantages and limitations in pipeline inspection. The improvement is proposed by monitoring the historical data and its extracted information.

To effectively analyze these stream data, a comprehensive modality inspection system has to be developed. Typically, the system must gather different information from the scanning, such as geometry regarding the length, width, depth, and location of flaw anomalies. Those data need to be compiled for integrity assessment and subsequent effective planning of repair and maintenance.

Currently, various expert tool configurations are available, and each design has been optimized to adapt to the inspection requirements of the pipeline industry. Multimodality inline inspection tools can provide important data regarding the characteristics of flaws and anomalies detected in a pipeline surface. Then the data analysis methods and models are utilized for defect quantification and classification. This information anticipates the defect growth rate and prediction model for condition-based maintenance . Based on the overview above, the challenges and trends of development are as follows [16]:

1. Multi-physical integration and fusion inspection are needed.

2. The challenge of operating time and adaptivity to varied environments

3. The accuracy of locations and shape of defects.

4. Multiple parameter measurement and characterization, e.g., integration of inspection and structural health monitoring, e.g., defect detection and pressure characterization.

5. Lifetime prediction, AI-assisted condition-based maintenance through intelligent data management.

1.1.4 Current issue of Case Base Reasoning system

Case-Based Reasoning (CBR) has long built a solid cornerstone in the expert system domain. The application is favorable and not limited to faults and troubleshooting problems. As approaching problems from a human perspective (i.e., judgment is concluded according to the outcome learned from experience) is seen as essential, industrial organizations are keen on consolidating systems following the CBR concept. Consequently, the process of CBR is designed to indicate the phases of analyzing experience, abbreviated into 4-R cycles: retrieve, reuse, revise, and retain. However, designing an explicit CBR endures inevitable restraints. Many existing CBR systems require indicators using expert domains [17][18][19][20][21]. Generally, most CBR systems are designed to retrieve similar cases automatically. The common methods include clustering techniques such as k-Nearest Neighbor (kNN) [22], kernel methods [23], distance measurement [18] Rule-based approaches are favorable [24] for some particular systems[20] [25] [26]. Unlike clustering methods, some rule-based systems obligate experts to construct the similarity computation [20] [25]. While the retrieval phase occasionally involves expert contribution, the knowledge of experts overwhelmingly allocates the procedure in the later phases. As being restricted to expert knowledge, most of the existing CBR systems can only manipulate machine learning to adapt the retrieval phase or somehow the reuse phase. Only a few studies proposed a complete workflow design for CBR system [27] [28]. Nevertheless, those studies were only applicable for a specific domain and lacked descriptions on the generic extension [27].

While the CBR concept exists in a systematical order, in the existing CBR systems, the methods used in each phase do not consistently well inherit. Only cases' features are invoked throughout the process, not the learning perspective of the engine [29][30][31][32]. As a common practice, the weighted distance method is employed to evaluate the similarity between the based cases and a new case. In other systems, this approach is replaced with other machine learning techniques such as Decision Tree (DT), Support Vector Machine (SVM), or kNN to return similar cases. What has received limited attention is the inheritance of the learning model to be applied in the four later phases - retrieval, reused, revised, and retained phases. Instead, the current CBR systems confine the latter stages with different approaches, primarily referring to the rule-based method. This indicates a need to clarify the effectiveness of governing the CBR pipeline through a consistent model. In particular, the model used to extract similar cases also contributes to the reuse, revise and retain phases rather than being erased when integrated with a different model.

In [17], the authors introduced a CBR system to match the design fixture of machines. The system employed the Minkowski distance measurement method, specified in the Euclidean distance, to evaluate the similarity of retrieval cases. The retrieval phase was split into which were feature selection processes and the distance measurement process. In the first process, specific features in different categories are inferred to describe the specialty of the machine fixture (dimensions of the workpiece, number of milling and drilling features, volume of components). The preprocessed features were applied to obtain the retrieval score. A list of similar cases was ranked, and the one with the closest distance was also adopted for the reuse phase. Although the system has been designed carefully in the first two phases, the later retain and revise phases have been injected directly into the knowledge of the fixture designer. Lack of learning information during the first 2 phases has been inherited to support the later processes. The same omission also appears in other research works. The fault detection system for the production of drippers, proposed in [19], requires an extreme contribution from domain experts. The feature selection process required filtering the functional features in producing dripper and original features of the injection molding machine process. The latter step defines the case representation, which specifies the most common faults in different injection molding machines. The selected features have been vectorized by a domain expert. In brief, with numerical attributes, the local similarity is computed by evaluating the differences or the quotient in comparison to case features. In contrast, similarity from the combination of all possible sets of values is computed for symbolic attributes. The approach cannot preserve useful information for the phases and the need for prescribing from experts to define the reuse, retain, and revise phase is inevitable.

In realizing the complicated dependence on defining rules from experts, many CBR systems attempt to detach expert bonding. The breast cancer diagnosis system, as proposed in [18] used multiple medical testing indicators as the main features to support the decision. Instead of following the medical diagnostic tree (which results in complicated expert rules verification), the system advances with machine learning techniques. The system implemented a heterogeneous Euclidean overlapping metric to cluster the group of similar cases by considering the overlapping region for discrete attributes and differences for continuous attributes. The system also implemented a statistical method to evaluate the score on the probability dependency of non-overlapping areas. However, this approach centers on features as the primary learning target but not the stats of cases. This approach becomes less effective when applied in an online system. Specifically, the genetic algorithm is used to optimize the weights of attributes in the distance measurement. Nevertheless, the reuse phase predominantly induces the null adaption with treatment from the nominated case is applied. Another medical system also utilizes the convenience of the CBR concept but follows the traditional medical diagnosis tree for treatment [25]. In this insulin bolus advisor, the case representation has been encoded as a sequence of patient actions. Technically, the time of taking medicine, the daily medical testing indicators are employed. The order of these indicators contributes to the generation of the patient's tree. Again, weighted distance computes the differences between the new case and sample cases. The formula has been adjusted so that the order of the indicator tree has a significant contribution. During the reusing phase, an advanced bolus calculator formula is used, and also special insulin sensitivity is computed to determine the revising step. However, this formula is only specified for this little insulin medical test field. In our work, we also introduce an indicator to measure the confidence in reusing and revising cases that is achieved by learning information from the retrieval phase. Thus, domain experts are free from affecting the final decision.

From the reviewed systems, the lack of a coherent pipeline throughout the system becomes a firm drawback. Therefore, this work allocates a substantial solution that maintains the unity of learning information. Our proposed system successfully achieves the following benefits. The retrieval phase requires only a few contributions from a domain expert. The knowledge of the retrieval is overwhelmingly allocated to the procedure of the sub-sequenced phases. Since the adaptation and adoption of new cases are the advantages of CBR, the use of the same engine for all four cycles considers the ability to update the learning factor actively. Various CBR systems introducing offline learning mechanisms as the main training mechanism have been developed. However, there are certain drawbacks associated with the use of offline learning, particularly the extension of a new case if the model is reused in the later phases. In fact, offline learning is set for the retrieval phase only.

1.2 The objectives of the study

Based on the list of current issues, this thesis aims to tackle the need to implement an end-to-end learning mechanism in the CBR system. The composure of a complete online

SVM-CBR is the primary subject matter in this thesis. The target of the thesis is to solve the research question:

1) How to construct a CBR system in which all phases require the same learning information of the core engine?

2) Is the proposed system feasible to apply into standard CBR practical problem, which is fault detection?

Consequently, this work addresses the effectiveness of replacing traditional offline learning with an online-learning mechanism while establishing a CBR system. Among various reported engines that have been successfully adopted into the CBR retrieval phase, such as decision tree, k nearest neighbors, and weighted feature distance, this work favors the SVM approach as the main engine. Due to the rich information when solving SVM, lots of the computation is reusable while optimizing the SVM. Sub-process during this learning is attachable to a CBR system, as referred to as the four-cycle design of CBR. The kernel trick evaluates the similarity of cases that contribute to the retrieved phase; the online SVM includes the processes of learning (adding support vectors which refer to as a critical case in SVM), and unlearning (removing non-support vectors refers as trivial cases). These processes are helpful in extracting the confidential scores, support vector allocation, and similarity value to attach to the reuse and revise phase. Thus, our proposed scheme maintains the learning information and completes the four-cycle design without adopting a different approach as in other CBR systems.

In order to validate the performance of the proposed system, a practical pipe defect detection problem is inquired, which is the second target of this search. As a result, this work also involves the design of inspection gadgets and the fusion of multi-sensor data. Consequently, in terms of inspection gauge, chapter 2 reviewed different designs of robots and also the experiments to evaluate the possibility of traveling inside pipeline structures properly. This work covers the defect detection of non-destructive testing methods using inspect sensors. Studies on in-line inspection robots are also reviewed to construct an efficient gauge. Chapter 3 described the proposed robot, which has the adjustable capability of carrying multiple sensors. We also introduced the use of specific Mindstorm sensors, including ultrasonic and color sensors, as well as the validation of the captured data quality. Information from sensors is used and compared with other third-party sensors to evaluate the suitability of NDT methods. The received signal from sensors will be processed and verified with the integration with CBR, the core engine, as presented in chapter 4. Nevertheless, intense study on the current support vector machine technique focuses on classifying the recorded data.

Chapter 2

Literature Review

2.1 A Review on inspection gauge

Together with indoor tasks, plenty of robots have been designed to conduct various outdoor missions. In operating in an open environment, the common tasks of a robot refer to exploration and inspection. In a space limitation and harmful condition environment, the replacement with the aid of a robot becomes an ideal solution. Specifically, in practical inspection efforts, the use of robots has achieved significant impact in many challenging conditions, such as electromagnetic, radiance field, dust, humidity, toxic, or dim light environment. As the robot's duty varies from different tasks, the design of the robot in a specific environment must be calculated carefully to ensure flexibility throughout its operation. Among various working conditions, tunnel inspection is currently a dominant area for the robot. The performance of the robot in this area was defined by the ability to operate under dim light conditions, narrow spaces, and stability when working for a long time with high precision.

Although lots of work has been studied on the design of a robot for in-pipe inspection, there is still not a universal model for working under all types of pipe. In the non-destructive testing (NDT) method, the involvement of the inspection robot attracts the interest of research as it tackles the limitations of traditional destructive methods [33]. Most pipeline system does not expose their construction outside. Instead, the system is designed to be buried underground or concealed behind a wall layer. This solution protects the pipe from surrounding environmental damage and enhances the elegance of the architecture. Consequently, locating the pipe by using destructive testing methods implies an increment in the cost of surrounding damage. A pipeline's structure is not always uniform; the structure alternates in terms of lengths, branches, and sizes for different purposes. The transmission pipeline for energy substances prefers straight and large channels. In contrast, the infrastructure pipeline for buildings contains multiple branches and complex skeletons due to the limited space of the construction. Lastly, the process of destructive testing methods may cause the leakage of harmful elements such as hazardous gas or overflowed substance. Therefore, NDT has become a favorable solution in the pipeline maintenance industry.

The ILI process is a set of operations aiming for the exploration and examination of the target pipeline. The process is executed by an inspection gauge specialized in scanning and verifying the pipe condition. As a part of the maintenance process, the pipe condition is anticipated with a healthy, clean, flawed status. An influential inspection gauge apprehends any imperfection which potentially degrades the pipe quality. If the conventional pressure test weakly recognizes any defect approaching the safe threshold, the combination of non-destructive evaluation (NDE) and the ILI method has the capability to detect and quantify considerably significant and critical flaws. Besides, the ILI technique issues a warning for high potential threats even when the pipelines have not been started for operation.

Regular work of flaw detection relates to the design of pipe inspection gauge (PIG) tool. In general, the inspection robot is classified according to the inspection target. A comprehensive survey has been conducted on the trend of inspection robots in [34]. Although the research concentrated on tunnel discipline rather than pipe structure, similar approaches with slight modification are applicable to the pipe cases. Firstly, the design of the robot should be adapted to different material structures ranging from concrete and steel to masonry. Due to the limitation of space capacity, in our only essential sensors are selected for the operation. In consequence, data collection methodologies have been categorized into visual, strength-based, sonic/ultrasonic, magnetic, electrical, thermography, radar, radiography, and endoscopy groups. Typical sensors, such as optical, laser, or impact sensors, are defined to match the appropriate group. Among various studies on the combinations of constructing ILI robots, the difficulties are found as the lack of fully automated operation (since most of them execute through teleoperation), incomplete inspection data, and the limitation of communication (tether length, wireless area). The challenge emerges as implemented in a pipeline system.

Resemble as tunnel inspection; the pipeline inspection also shares the same approaches as well as the design of the robot. However, when operating inside a cylinder surface, the creation of an ILI robot has to adopt five essential factors 1) shape and size, 2) navigation mechanism, 3) steering mechanism, 4) control technology, and 5) detection mechanism [35]. Details of these factors will be elaborated in the following subsections.

2.1.1 Shape and size

When traveling inside a narrow environment, the robot design may adopt the following forms of appearance to support the locomotion effectively, which are a) pig type, b) wheel type, c) caterpillar type, d) wall-press type, e) walking type, f) inchworm type, g) screw type. Depending on the pipe diameter and its transportation material, a specific robot model has been designated. Practical experiments also clarify the high correlation between the design selection and the in-pipe movement of the robot [36].

Particularly, the PIG type is recognized as the design for inspection of fluid pressure. The PIG model is widely used in the industry to maintain a large diameter pipe system [37]. Most of the common commercial solutions prefer wheel type in the interest of flexibility and simplicity [38][39][40]. The replacement of wheels with caterpillars is also a favored solution for industrial mass production. By using caterpillars, the robot restricts the driving options to only differential drive. On the other hand, the wheel model can implement a synchro drive as an alternative solution. However, when traveling in a vertical or diagonal pipe, both solutions need to compromise the balancing issue.

A typical wall pressed design is adopted to overcome this issue. In this design, the robot sustains the position and controls the motion by clutching its limbs sequentially. The design requires careful measurement so that the height and width of the robot fit the pipe diameter appropriately. This drawback imposes a re-scale to operate in different pipe sizes [41].

The inchworm robot specializes in small, narrow pipes. The long shape of the robot splits into sub-coaches as a train. Due to the limitation of space capacity, each sub-component usually specifies only 1 task. The common design composes of 3 parts, a motor control, a data collection or cleaning unit, and a motion supporting unit. Due to the advancement of micro-technology, recent research proposed various hybrid models using wheels and inchworm design. Practical experiments indicate the adaption of those hybrid model in distinct pipe systems [42][43][44][45].

2.1.2 Propelling mechanism

A study categorized the locomotion mechanism inside a pipe into three forms [38]. The first form utilizes fluid pressure in a pipeline to acquire the propelling movement. This solution requires less input power and precomputed mechanism. However, the design of the robot has to consider the integration of a fluid pipeline.

The second form is often applied for snake-like robot architecture, whereas transforming the propulsion through an elastic rod. This mechanism is efficient when operating in slender pipes. By implementing this mechanism, the robot can be divided into rather small coaches but flexible in movement. Consequently, the carrying capacity is increased since sub-coaches allow attaching more components. Nevertheless, the engine coach has to provide an additional tractive force for carrying its tail, which induces an expensive power consumption.

The last form is commonly used as a specific drive mechanism in its body. The design appears as kinematical modeling, in which the primary objective is to allocate the instant position of the robot. By inputting information about the plan profile of the robot, the robot can understand the route in advance and use an angular speed sensor that measures the speed and time to calculate the distance so far. An alternative solution is dynamic modeling, in which the locomotion has to evaluate the interference of flow.

2.1.3 Steering mechanism

A decent standard of robot motion endorses differential drive, synchro drive, and articulated drive as the appropriate steering mechanisms. If the robot's degree of freedom is accessible in outdoor space, traverse inside the pipeline employs specific computation to adapt to typical routes. The standard pipeline consists of the most direct straight line, the elbow (L shape), the branches (T shape, Y shape), and the vertical pipe [46].

The differential drive associates the robot motions with the movement of a tank. The robot's rotation accomplishes by modulating the speed of the side wheels, depending on the desired direction. Since the wheels rotate in a reversed order, this mechanism requires the understanding of pipe geometry and potentially originates the slipping problem. Conversely, synchro drive allows each wheel to steer to its own degree. As a result, the vehicle is more flexible but also consumes more power as each wheel connects with one motor.

While the first two drive options keep the whole structure unattached, the articulated drive splits the body into front and back halves for acceleration [47][48]. The design adapts to the curving surface and also integrates with other steering mechanisms in sub-domain but requires complex movement computation [43].

2.1.4 Control mechanism

The control mechanism has been classified into three domains: tethered cable, wireless communication, and automatic drive. With tethered cable, the robot utilizes the tethered connection as a flexible power supply. This design guarantees the robot to operate the mission in durable time. By connecting through a cable, the robot can switch between automatic and manual modes. The mode-changing options reduce the complexity of the automatic navigation workload. Nevertheless, for pipeline missions, this cable can be considered as a track line which becomes useful for rescue missions in emergency cases.

However, the attached cable also induces irresistible drawbacks. The system highly depends on the length of the cable. Besides, an additional mechanism has to be concerned to avoid entangled problems while exploring complex structures, and efficiency is reduced due to the friction force. The alternative solution with wireless communication allows the robot to perform freely in various architects. Herein, the solution also supports manual navigation in need. Similar to using a cable, the length of connection is limited by the range of the signal. In a specific environment like metallic material, communication can be penetrated severely.

The automatic drive is the most complicated solution as it involves a specific mechanism to support navigation and localization fully. As a the result, more sensors with high accuracy can lead to the fast decay of supplied power. Despite the effects, the benefit of freedom from labor work and fully automatic operation in restrained conditions have favored this option [49].

2.1.5 Detection mechanism

A standard method of the detection mechanism, which is integrated with most industries, refers to magnetic flux leakage. The principle of MFL originated from the magnetic particle technique. The leakage of magnetized flow flux to the outside environment occurs when discontinuous ferromagnetic structure. The approach has been used extensively in various petrochemical, energy, and metal industry. Consequently, the embedding device to record the MFL for inspection pipeline pig was proposed to detect the corrosion and defect in gas and oil pipe [41]. In particular, a strong magnet emits the magnetic field surrounding the pipe while traveling. In a closed, leak-less environment, the recorded signal is stable. However, at the metal loss position, the flow is alternate. Circumferentially distributed sensors monitor the flux signal and store them in the data acquisition system. The transferred data is analyzed to reveal information about flaws' size, shape, and location. .

The MFL system requires a complex distribution among excitation fields, leakage flux, and material defects despite the simple concept. A comparison study indicated the remaining MFL solution [35]. Firstly, to capture the magnetic flux that occurs at the defect position, the level of excitation magnetic flux requires to be large and homogenous. Secondly, the sensors must be closed to the defect position to read the leakage data. Building a susceptible sensor that adapts to noise and differentiates the origin signal is another concern. Besides, developing an effective inversion method to identify the flawed characters by the recorded MFL signals is difficult since the defect is irregular.

Eddy-current is another nondestructive testing method developed from the EC principle. EC manipulates the concept of electromagnetism and magnetic induction. Application of EC technique has been used generally in flaw detection of the conductive material surface. For example, a pipe made of conductive material like metal generates an electrical current as an interaction with the device inspection signal. The predefined current is generated through a coil system that controls current amplitude and frequency. Subsequently, as the result of mutual inductance, an alternative magnetic field is produced, which modifies the flow of EC on the surface of the nearby pipe wall. Hence, the current in the pipe wall is defined as a secondary magnetic field that is opposed to the primary area inducing it.

Material inhomogeneity, caused by corrosion or metal loss, modified the flow of ECs by the influence of mutual inductance. The pipe defect is studied in this phenomenon, and it concentrates on the measurement and analysis of the amplitude and the phase shift between the input and output signal. The height of amplitude cannot be analyzed further when a narrow frequency band occurs. This limitation is solved by investigating a pulsed signal in a broad spectrum. The spectrum perspective converts the narrow space into a wide depth range that becomes feasible to analyze. The flaw in a deeper surface is allocated by enhancing the high excitation current. To overcome the heating issue of the probe while applying high current, Pulsed Eddy Current (PEC) the technique is presented.

On the other hand, the ultrasonic technique exploits the high speed of transmitting sound to measure the changes in the environment. The ultrasonic wave typically has a very high frequency that ensures accuracy in real-time application. Furthermore, the ultrasonic wave has become one of the preferred solutions for nondestructive testing methods. It can be applied for multi-purpose, from identifying the defects in material on the surface to supporting navigation.

The process of ultrasonic measurement contains sending its sound wave to release its energy. Once sending signal impacts the surface, it reflects the source. Due to the high frequency and traveling speed, reflected sound is captured without concern for the misplaced source. The signal is analyzed to determine the presence and allocate the defect. By using this design, different defects such as cracks, gaps, or other discontinuities can be discovered as long as they present the reflecting surface.

2.1.6 Defect detection robot

A large volume of published studies describe the role of using a camera to analyze the defects. In the study of Halfway, an SVM classifier was applied to the histogram of oriented gradient Histogram of Oriented Gradient (HOG) features to verify the defect [50]. The images were filtered through a low-cost segmentation process to extract the region of interest. The study overcomes the instability caused by background noise and non-uniform illumination conditions by refraining from color intensity analysis. However, the study only reported on the severe defect detected and were lack of observation in the case of minor defects. Since the study used only HOG features, the variety of faults with different orientations and shapes degraded the performance. In another approach, Myron tackled the multi-scale GIST features and supplied them to a random forest classifier [51]. The approach profits from the GIST features that reduce the dimensionality of footage while preserving the original features that describe a frame's state. The reported results also revealed eight fundamental orientations in the four extracted scales. However, to improve the speed of surveying pipe networks, additional footage is required to be analyzed in advance.

Recently, several attempts have integrated deep learning (DL) models to investigate defect patterns. Almost at the same time, Kumar [52] and Cheng [53] explore the variation of convolution neural network Convolutional Neural Network (CNN) models. The performance of using CNN dominates previous studies and also advances traditional computer vision techniques in terms of multiple types of defects detected. The discussed studies analyzed images captured from the pipe's front instead of the side surface. However, the drawback of using DL is the size of imported models and hardware computational capacity. Consequently, the robot locomotion must be adjusted at a slow speed so that the model analyzes the video frame correctly. Moreover, DL requires a considerable volume of images for training to improve accuracy. Collectively, these studies outline a critical role for machine learning approaches to solve the pipe defect detection problem. With the rapid development of various computer vision models, the research has tended to focus on cameras rather than signal devices in wielding inspection robots. Since signal vibration and analysis have been verified to be successfully applied in the pipe network in the previous section, this indicates a need to understand the various perceptions of other sensors' signals when attached to an inspection robot.

In terms of fusion detection sensors, recent research has been operated in an open environment to support civil infrastructure. However, most of the research compiles sensors in the built-in static system rather than wielding an inspection gauge [54][55]. Lately, the robot designed by Gibb accumulates ground-penetrating radar Ground-Penetrating Radar (GPR) sensor, Electrical Resistivity (ER) sensor, and camera sensor [56] for aiding the defect decision. Particularly GPR functioning in reinforced concrete to allocate subsurface steel, while the ER is sensitive to the corrosion of concrete material. The output decisions from sensors are visualized with a camera system. The robot has been tested in various indoor and outdoor environments, such as parking garages and bridges. The study indicates the efficiency of governing multiple sensors for defect detection purposes and encourages the use of fusion sensors in pipe systems.

2.2 SVM in Defect Detection

The concept of SVM was introduced in the early of 90'. A theory on the convergence of the perceptron algorithm, developed by Novikoff, is the cornerstone for the SVM approach[57] In general, the theorem indicates the ability to use the input vector of features space to generalize the classification in Vapnik-Chervonenkis (VC) space. The learning mechanism associates the process of constructing a hyperplane to separate different classes into specific regions.

2.2.1 Hard margin SVM

Mathematical induction indicates that the performance of hyperplane construction improves with a deliberate solution. As a consequence, the importance of separating training data and problem generalization becomes a remarkable concern. Instead of minimizing the error measurement as in other learning mechanisms, SVM exploits the structural pattern of observations to construct a separation hyperplane. To avoid the obsession with minimizing the training error, SVM is developed by analyzing the assumption of small-size samples. Initially, SVM explores the Structural Risk Minimization (SRM) principle rather than traditional Empirical Risk Minimization (ERM)[58].

Assuming w denotes the vector orthogonal to the separating hyperplane, $x \in R^n$ is the representative vector of any vector in the sample feature space. The dot product $w^T x$ indicates the projection of x onto w coordinate. Mathematically, the primary objective is to maximize the margin of separation hyperplane as large as possible.

$$maxL_{w,b} = ||w||^2 - \sum_{i=1}^{l} \alpha_i (y_i (w_i . x_i + b)) \tag{2.1}$$

subject to

$$\alpha_i \geq 0, \forall i = 1, 2..., N$$

where as α is the Lagrange multipliers.

Consequently, Lagrange optimization can be solved with duality property. Instead of solving the primal form, it is proved to be more efficient to examine the dual state with the expansion of Karush-Kuhn-Tucker (KKT) conditions. The optimum point obtained from the duality problem is fully described as

$$\alpha = argmax_\alpha (\sum_i^N \sum_j^N \alpha_i \alpha_j y_i y_j x_i x_j) \tag{2.2}$$

$$\begin{cases} 1-\xi_i-y_i(w.x+b)\geq 0 \\ \alpha_i(1-\xi_i-y_i(w.x+b))=0 \\ \xi_i,\alpha_i,\beta_i\geq 0 \\ \beta_i\xi_i=0 \end{cases} \tag{2.3}$$

whereas ξ_i is the additional insensitive variable, in soft-margin approach, to measure how significant an instance is misclassified.

The prestige of SVM commits to its generalization ability. SVM explores the VC space which defined as the number of spanning vectors to separate different classes from the indicator function. The problem of evaluating the largest bound to is successfully proved to be a convex problem. Consequently, it guarantees that the searching space is free from local optima, and the convergent solution is also the global optimum. SVM has a sturdy mechanism for adequate capacity control by using the kernel transformation technique. As a result, the use of SVM has been successfully adopted in lots of real-world problems. Many implementations of SVM as the main engine for defect detection have been reported. [59][60][61][62] [63][64].

$$w^T x+b=0 \tag{2.4}$$

Assume the expected margins have the length of 1, and y_i defines the class sign of i^{th} vector. The decision rule is bounded by constraints

$$y_i(w^T x_i+b)-1\geq 0 \tag{2.5}$$

The primary objective is to maximize the margin of separation hyperplane as large as possible. From assumptions, the width of margins can be derived to be equal $2/||w||$. With the respectively constraints, the optimization can be rewritten using Lagrange condition.

$$L_{w,b} = \frac{1}{2}||w||^2 - \sum_{i=1}^{l} \alpha_i (y_i (w_i . x_i + b)) \tag{2.6}$$

subject to

$$\alpha_i \geq 0, \forall i = 1, 2..., N$$

where as α is the Lagrange multiplier coefficient corresponds for the $i^t h$ constraint.

The status of above equation can be analyzed by evaluating the derivative with respect to each variable w, b. Solving the equation of each derivative at critical points, we returns the stationary value for the optimal objective of L. Substitute back the obtained conditions the Lagrange form is simplified to be

$$L = \sum_{i}^{N} \alpha_i - \frac{1}{2}(\sum_{i}^{N}\sum_{j}^{N} \alpha_i \alpha_j y_i y_j x_i x_j) \tag{2.7}$$

Hence the goal is to determine α that maximized L, subject to

$$\alpha_i \geq 0$$

$$\sum_{i} \alpha_i y_i = 0$$

2.2.2 Soft Margin SVM

The above solution refers to the hard margin method, whereas the separation hyperplane strictly classifies instances. However, this approach exposes to be sensitive to the noise issue. An alternated solution, known as soft margin, tackles the problem by introducing an additional insensitive variable ξ called slack. Each instance has a correspondent slack value that measures how much the instance is away from its correct side. The soft margin tries to adjust the number of misplaced instances to be as small as possible. This leads to the same minimizing purpose of the predefined loss function. Hence, the objective function becomes

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i \tag{2.8}$$

, whereas $\xi_i \geq 0, i = 1,...,N$. C is a positive constant.

Apply the same process as in the hard margin; one returns the same conditions with a slight change in which the Lagrange multipliers have an upper bound is C. Hence, C indicates the trade-off between the insensitivity loss and the width of the margin (error and regularization). As C comes to a significantly large value, the objective function would be minimized, though the model becomes more complex with expensively computational cost, whereas a lower value would result in a simpler classifier. The first term represents the empirical risk or cost function and can be defined as the insensitive loss function. The regularization term defines fitness and is subject to constraints. Consequently, Lagrange optimization can be solved with duality property. Instead of solving the primal form, it is proved to be more efficient to examine the dual form with the expansion of KKT conditions. The duality problem is fully described as

$$\alpha = argmax_{\alpha}(\sum_{i}^{N}\sum_{j}^{N}\alpha_i\alpha_j y_i y_j x_i x_j) \tag{2.9}$$

$$\begin{cases} 1-\xi_i - y_i(w.x+b) \geq 0 \\ \alpha_i(1-\xi_i - y_i(w.x+b)) = 0 \\ \xi_i, \alpha_i, \beta_i \geq 0 \\ \beta_i\xi_i = 0 \end{cases} \tag{2.10}$$

When working with high-dimensional problems, SVM generalizes adequately by applying a pattern kernel which measures the similarity. A kernel is represented as

$$K(x,x_i) = \langle\phi(x), \phi(x_i)\rangle \tag{2.11}$$

in which ϕ is the transformation function that maps the input vector from vector space X into a higher dimensional space L. Depends on the objective function, common kernels such as linear and Gaussian will be effective to solve linear or non-linear problem respectively. In particular, Gaussian kernel and linear kernel of empirical data with variance σ and degree of freedom d were defined as below respectively

$$K(x,x_i) = e^{\frac{||x-x_i||^2}{2\sigma^2}} \tag{2.12}$$

$$K(x,x_i) = (x,x_i+k)^d \tag{2.13}$$

As replacing with kernel function and applying Lagrange techniques for the dot production calculation, former decision function becomes

$$F(x) = sgn\Big(\sum_{i=1}^{n} \alpha_i K(x,x_i) - \rho\Big) \tag{2.14}$$

The nature of SVM has been compared with various advanced techniques throughout decades in handling multiple pattern recognition problems. Unlike some algorithms like a neural network, the concept and the execution process of SVM were developed purely in math. The strength of SVM is based on its generalization ability. SVM explores the VC dimension, defined as the maximum number of vectors that can separate different classes of the indicator function. The sketching hyperplane is the security for the accuracy of classifying new input. Evaluating the largest bound to separate classes concerning the kernel transformation has successfully proved to be a convex problem. Consequently, it guarantees that the convergent solution is also the global optimum and free from local optima. In addition, SVM has a solid mechanism for well capacity control using the powerful techniques of kernel transformation.

Recent learning model architecture has been classified into four types, specializing in its depth. The first shallow architecture considers a fixed preprocessing step, followed by a single learning layer. The description refers to logistic regression and perceptrons techniques.

The second shallow architecture is composed of a template matcher layer that measures the similarity of inputs and one learning layer to adjust the coefficient. Vapnik version of SVM using kernel trick has been classified to this type. In particular, SVM with linear kernel would result in type-1 while Gaussian kernel that calculates the similarity results in type 2 architecture. The last shallow architecture is designed by a simple trainable basis function and linear predictor. SVM with Gaussian radial-basis-function (RBF) at which kernel function is learned to represent this type. Nevertheless, architecture consists of many layers representing non-linear computation which contains trainable parameters at all levels describing the deep architecture. However, in evaluating the two approaches, a variety of aspects should be put into consideration. Apart from the beautiful mathematics description, which restrains SVM would be its simplicity and abstraction. From an architectural perspective, SVM is similar to a small two-layer network. Particularly, the first layer identifies the similarity between inputs and stores the training samples as the prototype input. The second layer linearly integrates these similarities. An SVM with a "narrow" kernel function can always learn the training set perfectly, but its generalization error is controlled by the width of the kernel and the sparsity of the dual coefficients. As a result, SVM already embraces itself a limitation of exploiting the fancy features of preprocessing as compared to a deep architecture. Particularly, deep architecture can extract more abstract features at a higher level from the raw data easily, whereas applying kernel tricks would be exhaustive for SVM. Besides, it is an argument which is more important between the ability to execute a highly complex function with limited computation and the ability to control capacity.

Vapnik proposed another scheme to identify learning structure, in which he introduced the core mechanism, accelerating mechanism, and synergy mechanism. Although the trend today is building a deep convolution network for analysis and prediction, in the pipeline detection problem, SVM has the power to identify flaws by transforming to a higher dimensional features space. Many recent types of research implement SVM as the main engine for crack detection

In terms of model complexity, SVM keeps simplicity as a shallow architecture in comparison with other learning approaches such as neural networks or random forests. In particular, SVM with a linear kernel consists of a fixed preprocessing step, followed by a single learning layer. SVM with Gaussian kernel is composed of a template matcher layer that measures the similarity of inputs and one learning layer to adjust the coefficient. The last shallow architecture incorporates a simple trainable basis function and a linear predictor with kernel learning SVM as a representative.

2.2.3 Solving SVM in Defect Detection System

As superior as presented, the SVM consolidates being the legitimate model for solving in-pipe defect detection problems. The nature of this problem also concurs with SVM characteristics. The appearance of flaws inside the pipe is not trivial. Instead, the gauge will experience common signals frequently. Since working in a stable environment, the retrieved data is consistent most of the time, and hence, the detection task is closed to an outlier detection problem. In this case, an approach that interests in the generalization capacity is granted high priority. Besides, due to the limitation of the supplied power, the optical data will be restricted from operating as a supporting component in this system. The need for complicated architecture to extract high-level features neutralizes.

Optimizing the SVM problem is attempted in various directions. A comprehensive survey of J. Shawe-Taylor categorized the SVM optimization methods into seven assortments: interior-point; chunking, Sequential Minimal Optimization (SMO); coordinate descent; active set method (operation in online learning mechanism); Newton's method; stochastic sub-gradient; and cutting plane [65]. In many problems, selecting an appropriate optimization solution affects the training process. The study indicates that interior point algorithms are reliable and accurate in handling issues with thousands of samples. For large-scale problems, the sparsity of dual variables or compact representation must be adopted to manage the model capacity efficiently.

The common SVM optimizing approaches generally reflect solving Quadratic Programming (QP) problems. The interior-point approaches are associated with evaluating the Cholesky decomposition. Using linear algebra conventions, the process simplified the objective function and the constraints into solvable components. However, this technique calculates a matrix scaled by the number of training samples. The result of this model intensifies the resource capacity and consumes lots of time for training. Hence, the interior-point algorithm is practical with a small-scale problem. In SMO, the method exploits the equity constraints and the property of chunking approaches. The solution is preserved if the columns and rows corresponding to the zero entries coefficient are removed. The SMO tries to solve the sub-optimal problem by adjusting a pair of Lagrange coefficients in each sequence. Finally, the analytical bounding box constraint conserves the optimization process from the QP numerical calculation [66]. Hence, the performance of SMO has been reported to be approximately hundreds of times faster in some problems as opposed to the interior-point methods.

2.3 Expert System in Defect Detection Problem

The concept of case-based reasoning was introduced in the late 80s as a relatively new field of artificial intelligence [67]. The process of CBR is an upgrade version of the rule base system when applied to general areas of the hardware system. Conventional Rule-Based Reasoning (RBS) requires experts to extract powerful rules and conditional flows. In fact, the intended knowledge compels heavy time-consuming in a broad range of knowledge problems. However, it is difficult to deduce a thoughtful understanding that generalizes all problems, and hence shallow knowledge may be applied to the system. The RBS is feasible in a stable, well formalized, built-in problem. The limitation of this concept falls on the neglect of the new feature in system decision [68]. Although those limitations can be solved by improving the paradigm, developing an advanced ontology, and adopting better elicitation techniques and tools, an alternative solution adapting historical solution, known as Case-Based Reasoning, has received attention. Consequently, CBR does not have to identify the explicit domain model, and the process of elicitation refers to the collection of cases. Instead of applying rules to the problem, CBR solves problems by making use of solutions to the previous problems of the exact nature. The implementation is also simplified to address the essential features that control the case rather than generating an explicit model. Hence, it reduces the workload for data maintenance of large volume [69].

The evolution of CBR starts with the proposal of a memory-based expert that imitates the analysis of human experience in the same context. These experiences refer to the memory organization packages (MOPs) that are interconnected in a hierarchy structure. The MOPs connection is correctly mapped with sequential events. From a list of specific cases, the knowledge is firstly utilized. Whenever a novel case not related to any existing data is received, the system will evaluate the similarities with the stored experience. In some situations, the differences rather than similarities influence the decision. This mechanism is to prevent overlearning (i.e., when the system is fed with too many similar problems; it has a tendency to match any new one to the same cases with only a small level of similarities). As additional cases present, the knowledge will be adjusted and augmented. Kolodner indicates four assumptions representing the basis of the CBR approach. The same actions execute under the same conditions tend to have closed outcomes known as regularity [70]. Typical cases are built through regularly confront resembled patterns. Small changes in the situation require minor changes in the interpretation and solution. The compensation mechanism is applied to confirm the adaptability when repeated cases occur with minor differences. These assumptions are transformed into the CBR cycle as processing stages. Kolodner describes case retrieval processing as nominating a solution from the best matching case after assessing the problem. The retrieved solution is adjusted to adapt to the new problem in the case of adaption. Later, the solution's outcome is examined to determine its fitness. A non-adaptive solution may reverse the process back to the first stage to retrieve another closer solution. The reasoning system will be updated whenever a new case is correctly verified. The cycle has slightly changed in Aamodt and Plaza scheme [71]. In this description, the life cycle has been rearranged into four parts: 1) Retrieve the most similar case(s), 2) Reuse successfully solved case to solve the problem, 3) Revise the proposed solution if necessary, and 4) Retain the new solution as a part of a new case.

The four-cycle concept has been used widely in up-to-date CBR systems[72–77]. However, the lack of a standard transitional mechanism results in the combination of separated models used for each phase. Importantly, only information of cases is preserved during the cycling process, and models are switched after each phase. Consequently, their decision factors are eliminated. Several reported systems indicated this deficiency in their design. The early CBR systems were incorporated into a common flow. The list of similar cases is retrieved by measuring the feature-weighted distance. The other approaches in the family of feature computation, such as kNN, are also favored. Upon completing the retrieval cases, most systems define a specific adaption function provided by experts to determine the proposed solution. This solution is verified by either reassessing the performance of similar cases or calculating another expert predefined function. Such CBR systems are well reported in the past decade [32][22][78]. Up to now, the most common CBR designer frameworks are myCBR and jCOLIBRI. The myCBR framework was developed by the German Research Center for Artificial Intelligence. The framework has been used widely for many small and medium systems. The framework offers the designation for both hierarchical and feature-based case representation. In terms of the four-phase cycle, myCBR treats them separately. In each cycle, the framework suggests different approaches. For instance, the two most common options for the retrieval phase are weighted distance measurement and kNN clustering. On the other hand, the null adaption is mostly used with weighted distance, while the adaption solution, which takes the average solution, is applied if the group of k-neighbors is selected. Similarly, the jCOLIBRI framework, developed by the GAIA artificial intelligence group at Complutense University in Madrid, also tackled the same direction.

Apparently, the connection between phases is reflected through the case representation features only. Besides, the use of weighted distance in the first cycle results in a high dependence on expertise knowledge which restricts the system from extending or integrating with different domains [79–82]. This leads to the use of more powerful machine learning approaches in the recent CBR to reduce the overload of expertise input [83–88]. In the medical system reported on [30], the decision tree approach was implemented to extract similar cases. However, learning properties in decision trees such as retrieval cases entropy and their depth have not been concerned in the reuse phase. Instead, the author introduced a fuzzy method to trigger the case treatment based on the retrieval cases' features. Although the system reported proposing the replacement successfully, the fuzzy equations of this system are prone to changes of the case structure or the appearance of new features, especially when applied with a different design. Despite the success reported in this system, the neglect of valuable learning properties should be reconsidered. As the system is used for the treatment problem, the information from the diagnosis of the retrieval phase and the fuzzy equation in the reuse phase should be preserved. Since treatment is based on symptoms which is also the estimation of the similarity, this information connects with the revise or retain phases to keep it as a sample case. Similarly, in [22], important learning properties were omitted when integrating the retrieval cases and reuse strategy. Remarkably, the business failure system did not employ centroid information of selected cases as a feature and was consumed in later SVM decision models. In [29], the atmosphere-ocean evaluation system adopted a weighted distance computation instead of a machine learning approach to assess the case similarity. As a result, no information from the decision was transferred as a vital factor to the reuse phase. Despite applying SVR to evaluate atmosphere-ocean in the reuse phase, the integration of reuse and revise phases only commits to reassessing the SVR score. Since such a strategy implies a high error margin, an expert equation was needed to validate the proposed solution.

The need to fully utilize learning information and to exploit the data structure has been raised [89]. The study emphasized the limitation of the existing CBR to underlie the complex structure of data and was concerned about their efficiency as an extent to the problem with high-dimensional features. Another study that tackled the same problem as in our paper, perhaps, is the Bayesian Case Model as described in [28]. While the author solved the utilization of learning information by introducing an outer shell Bayesian model to connect the information deduction after each phase, our system targets the internal learning engine. It aims to deliver recalled functions used throughout the system.

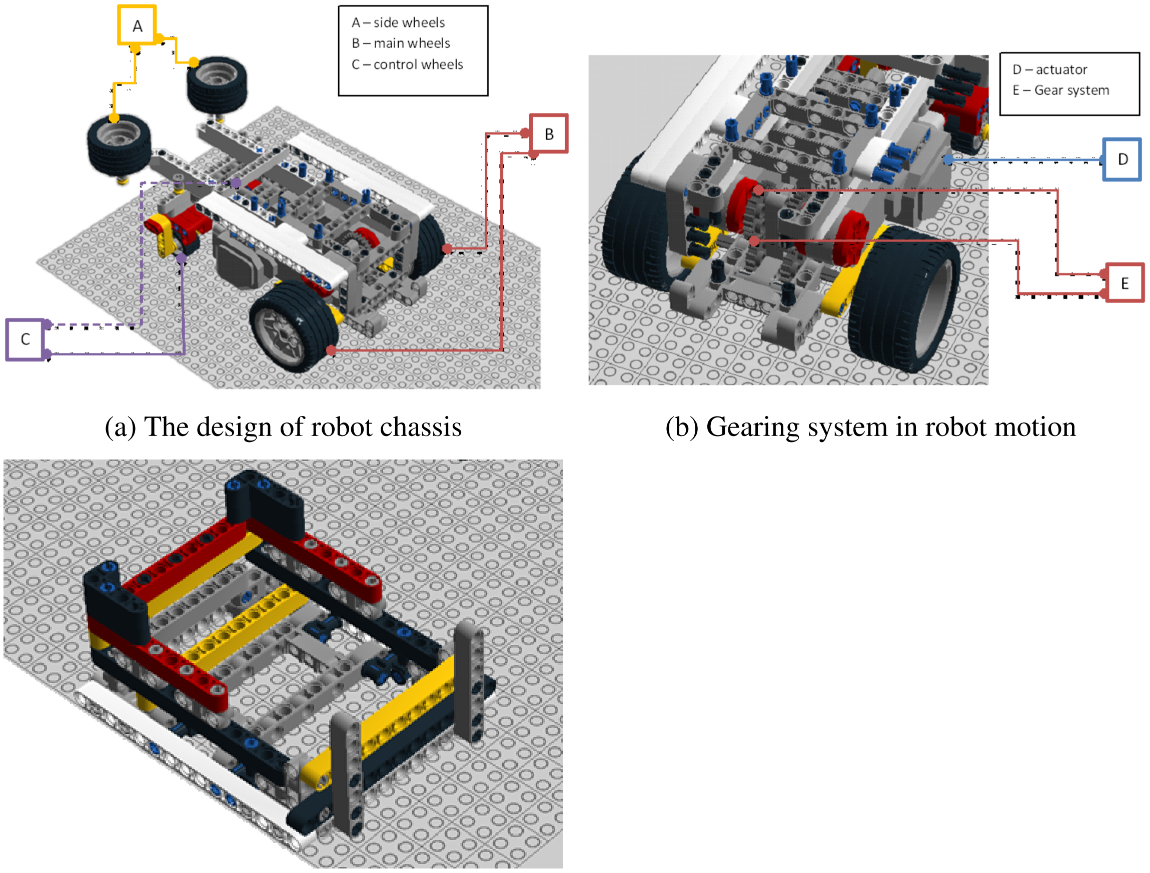
Chapter 3

System Design

3.1 The design of inspection robot

A review from Novak has classified sensors fusing methods from multiple modalities in a wearable robot into four categories [90]. A single fusion algorithm is the most common and direct approach. In this category, features from each separated modality are extracted and concatenated in the general data frame. In unimodal switching, a specific modality is served as a conditional verification to switch between different classifiers. The final sensor algorithm only uses the second modal as the input. In multimodal switching, still, one modality determines the transitional strategy. However, the sensors algorithm uses multiple modalities as input. Lastly, the mixing approach operates with multiple sensor fusions. Each is controlled by one or more modalities and executed in parallel. The output of these algorithms is appended with the corresponding contribution weights determined by a specific modality. In our work, the single fusion algorithm will be adopted as the dataset is not heavily complicated, and pre-processing steps have been successfully concatenated and separated features into a final data frame.

The pipe defect detection system operates through 2 major sections: an inspection robot that specializes in scanning data and a central data processing machine that concentrates



(a) The design of robot chassis

(b) Gearing system in robot motion

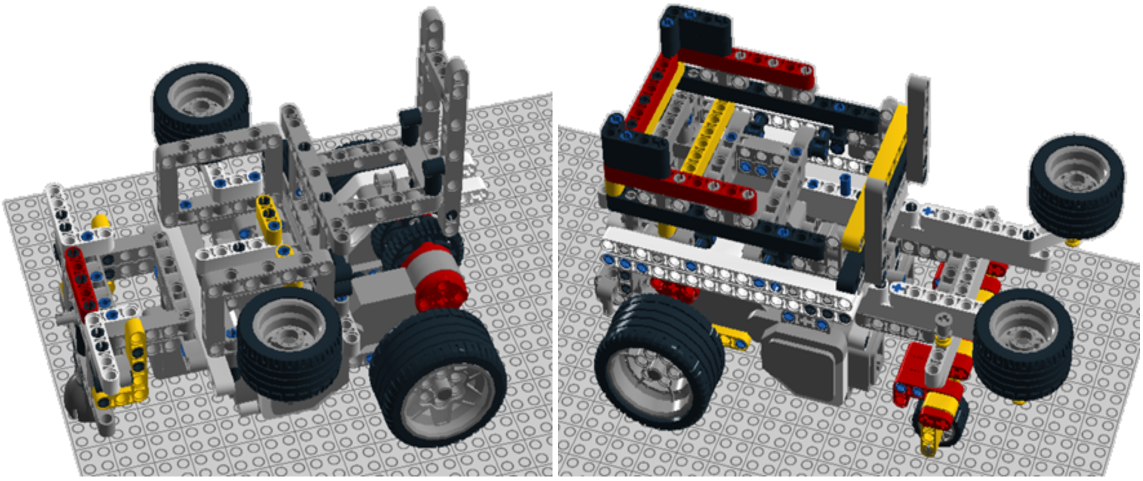
(c) Bracket design for holding SBC

Fig. 3.1 Decomposed structure of the robot components

on verifying the pipe flaws. The LEGO Mindstorm packages cooperated for the robot construction. This design has been selected due to the simplicity and robust support of the LEGO community. The modification can be easily customized with respect to the flexibility of LEGO components. Lastly, primary LEGO sensors and actuators are compatible with ordinary Single-Board Computer (SBC) such as Adruino or Raspberry Pi (RPi). Thus, the prototype is deliverable in the meantime. In our work, the robot design was conceptualized using CAD and Lego Digital Design. Several successful designs have been deliberately considered to determine the most suitable decision. Conclusively, the traditional sewer inspection robot, inspired by the KURT model, was favorable. The design is simple yet efficient to deliver an expeditious prototype. Practical implementation in both laboratory and practical pipeline systems certifies the ability to carry multiple sensors. Other strong candidates have not been selected due to different reasons. The disadvantage of multi-sensors calibration is the handicap of inchworm models like GRISLEE [91]. Wall-pressed or hybrid caterpillar, resembling FAMPER model, interferes with the scanning signals with the robot frame if multi-sensors are mounted [92].

3.1.1 The design of robot frame

As described in figure 3.1, a mixture of wheel and wall-pressed type is integrated. The chassis connects the actuators and holds the container. The robot motions function through two back actuators. The gearing system is designed to convert actuators' commands to the wheel motions. At the front, two control wheels are employed to maintain the robot's tension and trajectory. On top, a bracket assembles according to the size of the target SBC. The standard differential drive is implemented in the steering mechanism. The robot motion is also massively supported by two side wheels. These wheels adjust the robot directly to the relevant pipe flow in case of collision. They also function as the pressed arms, which retain the robot's balance status and prevent the flipping issue.



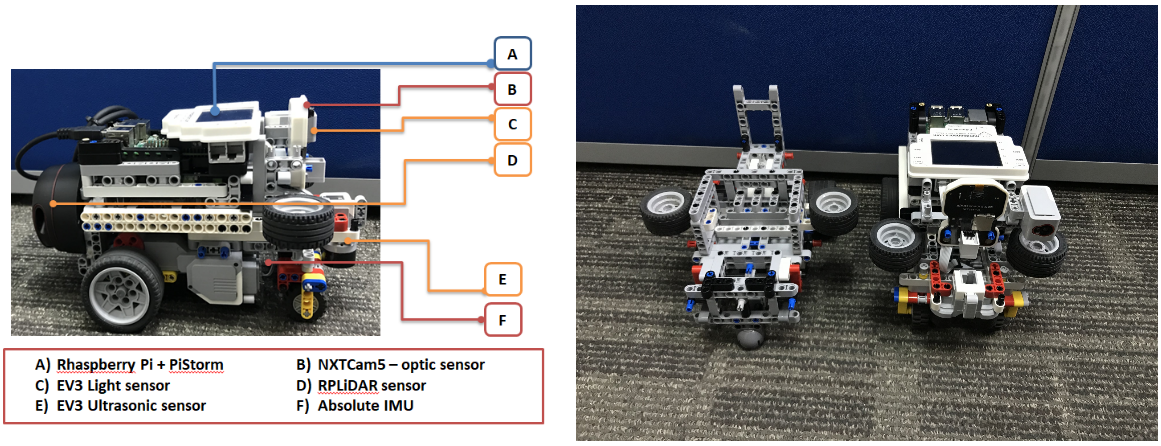
(a) First robot prototype with one control wheel (b) Final robot design using 2 balancing wheel

Fig. 3.2 Design of the proposed ILI with the decomposed structure of the robot components

Two versions are fabricated based on the above concept 3.2. In the first version, the robot uses only one sphere control wheel to support the locomotion. This design aims to reduce the friction force on the robot's motion at turning points. However, the contact area between the robot and the surface spans only three wheels. The robot's focal point sketches in a small tangent plane; it becomes unstable to maintain the robot's balance. When an impact occurs, the front wheel, which is much lighter, easily flip off caused by the strong traction force from the heavy back wheels. Therefore, small adjustments have been accomplished in the later model. An additional control wheel is appended to enlarge the tangent surface. The sphere wheel is also replaced by the regular round one to increase the contact area. Lastly, the bracket is designed to be adjustable so that the robot center regulates with the attached sensors.

3.1.2 The design of inspection sensors

Most pipe inspection robots favor only one detection method. In addition, the design of the robot also determines a specific type of detection. Unconventionally, exploring carbot usually monitors with optics sensor, whilst pipe fitting inspection gauge mounts with



(a) The implementation of robot wielding all sensors

(b) The implementation of the first prototype (left) and final design (right)

Fig. 3.3 The LEGO implementation of ILI design

ultrasonic. Specifically, the KURT industrial robot, developed by GMD, uses a camera for autonomously exploring and scanning defects in the sewer. The GRISLEE robot, designed as an inchworm, only attached an MFL module in its body to discover anomaly signals in the gas pipeline [91]. The PIG type capitalizes on the fitting with inspection pipe for efficiently scanning data. The detection part is designed with multiple ultrasonic sensors attached surrounding a ring, whereas a single sensor supervises only a set area. Although distinct signal determines the pipe defect problem efficiently, the necessity of merging different sorts of signals is considerable. By combining these signals in one inspection gauge, the strength in each detection method is utilized and also assists with the other's weaknesses. Thus, our work engages the process of integrating different sensor signals and the relevant performance.

Existing LEGO EV3 sensors are utilized for robot operations. Initially, LEGO sensors only present single refined output rather than the sequence of coarse signals. Since the detection procedure requires the pattern of serial data, these returned features are revised to adopt the interest. To assist the locomotion and capture data comprehensively, third-party sensors, including LiDAR, IMU, and NXTCam, are also attached.

As demonstrated in figure 3.3, the wielding of sensors specifies with the following description. The data collection is conducted using LEGO Ultrasonic, LEGO Color, and LiDAR sensors. In our design, the LiDAR is attached at the robot's end, perpendicular to the pipe cross-section. Since appending at the tail, the scanning space is ensured to be free from obstacles. The ultrasonic sensor is the changes in signal pulses pattern, and the color sensor is focused on the light stain, which is attached in the same orientation.

The remaining sensors mainly support navigation and improve data visualization. Notably, an Inertial Measurement Unit (IMU) is attached under the chassis to return the robot's six degrees of freedom status - the coordinates of the robot in 3D planes and the roll, yaw, and rotate angles. An optic sensor is attached at the front to avoid obstacle collision. It also captures photos during the operation for the verification process. The observed data is stored inside the robot memory and transferred to a connected computer using the conventional client-server solution.

3.1.3 The design of control system

According to the design, Brick's official LEGO Mindstorm processor is replaced by a more powerful SBC. However, the limitations arise from the dependence on the Brick processor. Specifically, the Brick compressed Linux OS eliminates important features to simplify the operation and maintain stability. Consequently, customization on the existing the sensors is restricted; while accessing external sensors is also restrained. The concern resolves by using an RPi mainboard with PiStorms add-in control unit. The replacement provides a stronger computational resource and is more flexible for multi-purpose tasks. In addition, RPi is also available for many robotics frameworks. As combined with a PiStorms control unit, external components are easily configured to be compatible.

To secure the accuracy of data location, the IMU sensor keeps track of the traveling distances and maps with the relevant scanning records. Therefore, the pipe architecture

IMU-milestone	x-axis	y-axis	z-axis	x-angle	y-angle	z-angle
LiDAR	-6.2 cm (behind the IMU)	4.5cm (above the IMU)	0cm	90 (face up)	0	0
Ultrasonic	5.6cm (in front of IMU)	3.8cm (above the IMU)	0cm	90 (face up)	0	0
Light	3cm (in front of IMU)	2cm (above the IMU)	2cm	0	0	0

Table 3.1 List of sensors attach locations as comparing with the IMU sensor

is simplified in this work to contain only a straight route. Other architecture such as Y-branch, diagonal turn, and vertical pipe will be examined further in the later industrial phase. Throughout this work, the data science toolkit sci-kit-learn has been imported to process the data frame, including cleaning, converting hierarchical structure to tabular structure, and metrics computation. The robot operating system is run through PiStorm, which is a distribution of Raspbian OS specialized for Raspberry Pi. The OS also consists of a compiled module to communicate with the LEGO actuator and sensors. It is also available for installing the Linux library to integrate with other sensor products. For detection model, popular toolkit LibSVM was adopted [93].

Since the robot is attached to multiple sensors, calibration steps are prepared to map the scanning result from the sensors. As mentioned in the JSON format of the data, the IMU sensor indicates the milestones for the references of other sensors. Remarkably, the distances from different sensors to the IMU are carefully measured and fixed. In this robot, the detailed mapping of each sensor as compared with the IMU is described in the table3.1.

The detection of any defects from one sensor will be interpreted as the appearance of defects within the radius of the scanning sensor from the center of the robot - which is the IMU. Additionally, sweeping sensors like ultrasonic and LiDAR setup the same angle in the beginning - by default is to face up - and configure the frequency to be multiples. This spinning rate selection eases the angle mapping of sensors as well as validates the scanned location.

3.2 The design of defect detection engine

Support vector machine has been recently reported as the most suitable solution in many signals processing problems [94] [95]. In terms of structural risk minimization, it is considered state-of-the-art due to its superior performance and solid mathematical background [96]. Optimizing the SVM problem has been attempted in various directions. A comprehensive survey of J. Shawe-Taylor categorized the SVM optimization methods into seven assortments: interior-point; chunking, sequential minimal optimization (SMO); coordinate descent; active set method (operation in online learning mechanism); Newton's method; stochastic sub-gradient; and cutting plane [65]. In many problems, the selection of an appropriate optimization solution affects the training process. The study indicates that interior point algorithms are reliable and accurate in handling problems with thousands of samples. For large-scale problems, the sparsity of dual variables or compact representation must be adopted to manage the model capacity efficiently [66].

The common SVM optimizing approaches reflect solving the quadratic programming (QP) problem. The interior-point strategies are associated with evaluating the Cholesky decomposition. Using linear algebra conventions, the process simplified the objective function and the constraints into solvable components. However, this technique calculates a matrix scaled by the number of training samples. The result of this model intensifies the resource capacity and consumes lots of time for training. Hence, the interior-point algorithm is only practical for a small-scale problem. On the other hand, an SMO method exploits the equity constraints and the property of chunking approaches. The solution is preserved if the columns and rows corresponding to the zero entries coefficient are removed. Finally, the SMO tries to solve the sub-optimal problem by adjusting a pair of Lagrange coefficients in each sequence. The analytical bounding box constraint conserves the optimization process from the QP numerical calculation [66]. Hence, the performance of SMO has been reported to be approximately hundreds of times faster in some problems as opposed to the interior point methods [97].

Since applying box constraint, the convergence rate of the SMO approach heavily depends on the trade-off parameter C between regularization and error. Therefore, if high accuracy is demanded, the approach requires more computations to reach the critical point [98]. In a problem with thousands of support vectors, the additional process concerning a more extensive working set or solving the QP sub-problem isis adopted instead of examining a pair of vectors.

3.2.1 Sensor CBR system – online or offline learning

As stated in the review, solving a SVM problem can be directed from different perspectives. The selection of an appropriate optimization method must content the performance accuracy as well as the operation time. In addition, the selected approach must sustain in increasing of feeding data in real-time. As the scanning frequency is very high, the core engine must learn effectively in a very short time to match with the fast streaming. The forgetting issue of incremental learning is also an important aspect for considering. Although, these constraints suggest an online solution for tackling the SVM problem, the solving mechanism which simulates the book keeprecords as in CBR system is the main motivation.

Different from other techniques, while the number of features heavily affects the model complexity, SVM concerns more about the number of training samples. When implementing reproduce kernel Hilbert space to the SVM dot production computation, samples covariance in higher dimensional space is accessible. However, while computing the kernel depends on the data size, it can be exaggerated quickly if the number of support vectors is large. As solving SVM loss function in dual forms, the time performance is scaled proportionally with the data size.

The classical off-line learning, or batch learning, does not fully examine the whole data given the time restriction. The off-line learning mechanism of kernel classifiers also prohibits their models to resume the training. The unprocessed data must be retrained from scratch which is time-consuming and inefficient if the model is required in a specific time interval. To overcome the drawback, training paradigm in streaming fashion, where data are input sequentially, is encouraged. Particularly, each sample is learned at an instant, and the supported kernel is evaluated in the meantime. The process of updating kernel is simplified with appending additional row and column to the existing matrix if it is classified as the support set. The mechanism is practical for the system that operates with high frequency signal sensors. As the sensor spins in a rapid rate, hundreds of new data are captured and overwhelmingly fed into the model in a few seconds. Hence, an online fashion would be appropriate to serve this rate of incoming data.

3.2.2 Current online SVM

Many online SVM approaches have been proposed and achieved comparable results with the conventional SVM [99]. The optimization methods of online SVM also tackle the same process of offline learning. In general, there are two common approaches. The first approach aims to adjust the element inside each set, including support set, error set and remainder set, while the second approach involves the study of changes of loss function. The first method relates to incremental and decremental SVM [100] while the second refers to the primal estimated sub-gradient solver(PeGaSOS) method [101]. In dual form, the problem is solved by adjusting a pair of support vectors while maintaining the Karush-Kuhn-Tucker (KKT) condition. The incremental technique adopts sequential minimizing optimization (SMO) process to determine if the insertion into support set violates the KKT condition. The other common solution is to apply stochastic gradient descent (SGD) that minimizes the loss function of structural risk minimization (SRM) using the representation of a predefined loss function such as Hinge loss. However, when using SVM as the core engine of CBR system, solving SVM with the first approach intuitively corresponds to the four phases of the CBR, especially the similarity estimation.

The most common incremental SVM that solves online SVM directly by considering the status of active sets, was introduced by Cauwenberghs and Poggio [102]. A new sample is inserted into the appropriate set by determining the closest marginal from the sample to each set. The optimal solution is updated by the process of adjusting the existing samples to the relevant sets after a new insertion. However, the approach is favored as a quadratic programming (QP) optimization problem rather than solving online classifier problem. Consequently, little of successful practical applications utilizing the approach have been reported.

Due to the distinct description during the optimal process, many extensions have been developed to enhance this approach's solution. Almost at the same time, Ma and Martin recognized the rules of removing or adding samples to appropriate set for unlearning process [100][103]. The defined rules justify the searching conditions and direct their path to the optimal solution efficiently. The method implements a bookkeeping procedure that records the actions of transferring samples among sets when a new sample is added to the training set. Practical implementation on time-series forecasting with cross-validation mechanism denotes the efficiency of bookkeeping as opposed to the traditional batch learning.

Similarly, the work of Martin extends the incremental SVM (ISVM) in classification tasks to function approximation. The searching rule relies on the KKT conditions and adjusts their multiplier β. The modification is applied with respect to reserve the constraint conditions of the remaining data. In general, the training sequence consists of three main processes, including incrementally adding new data to the training set, removing data from the support set, and updating target values for existing data. A comprehensive study indicates the increase in computation in quadratic time is the main drawbacks of incremental SVM. The complexity heavily depends on the balance of memory access and arithmetic operation [104]. Therefore, implementation of ISVM is not as favored as other powerful batch learning packages like LibSVM or SVM light.

3.2.3 Incremental SVM

The proposed solution of Laskov and co-authors directly restructures the accounting storage and reorganizes the computations [104]. The solution is considered as a lossless model since it maintains all of the observed data and arranges them in the appropriate sets. The approach exploits the KKT conditions defined as below

$$g_i = \alpha K_i + \mu y_i - 1 \begin{cases} \geq 0 & if\ \alpha_i = C \ (\ remainer\ set\ R\ ) \\ = 0 & if\ 0 < \alpha_i < C\ (\ support\ set\ S\ ) \\ \leq 0 & if\ \alpha_i = 0\ (\ error\ set\ E\ ) \end{cases} \tag{3.1}$$

When new streaming data c is input, the Lagrangian coefficients must be adjusted to satisfy the constraints. Instead of solving the minimax problem of SVM batch learning, ISVM considers minimizing the loss of new sample with previously observed data. To enhance the efficiency of computation, a compact matrix Q denotes the kernel representation of support sets and their sign has been introduced. The compact representation of changed variations are described as follow

$$\beta = - \begin{bmatrix} 0 & \alpha_s^T \\ \alpha_s & K_{ss} \end{bmatrix}^{-1} \begin{bmatrix} \alpha_c \\ K_{cs}^T \end{bmatrix} = -Q^1 \vec{\eta} \tag{3.2}$$

$$\gamma = \begin{bmatrix} y_c & K_{cs} \\ y_r & K_{rs} \end{bmatrix} \beta + \begin{bmatrix} K_{cc} \\ K_{cr}^T \end{bmatrix} \tag{3.3}$$

where β indicates the sensitivity of observed data from the support set with respects to the new sample c; and γ indicates the sensitivity of margin from the remainder set with respects to the new sample c. The largest possible increment of the new sample is determined by a bookkeeping procedure. The procedure accounts for the changing structure when a sample reaches its set variation. As presented in figure 3.4, four possible cases of constraints violation have been analyzed: a support coefficient reaches its bounding constraints; a remainder sample shifts to the margin when g_i closes to 0; the new sample belongs to support set which requires updating the other coefficients, and the new sample coefficient reaches the upper bound constraint. The moving sample that yields the minimum variation is transferred to the relevant set. Once the new sample is allocated to the correct set, the inverse matrix Q is expanded with an additional zero row and column, and its updated result is obtained by matrix multiplication.

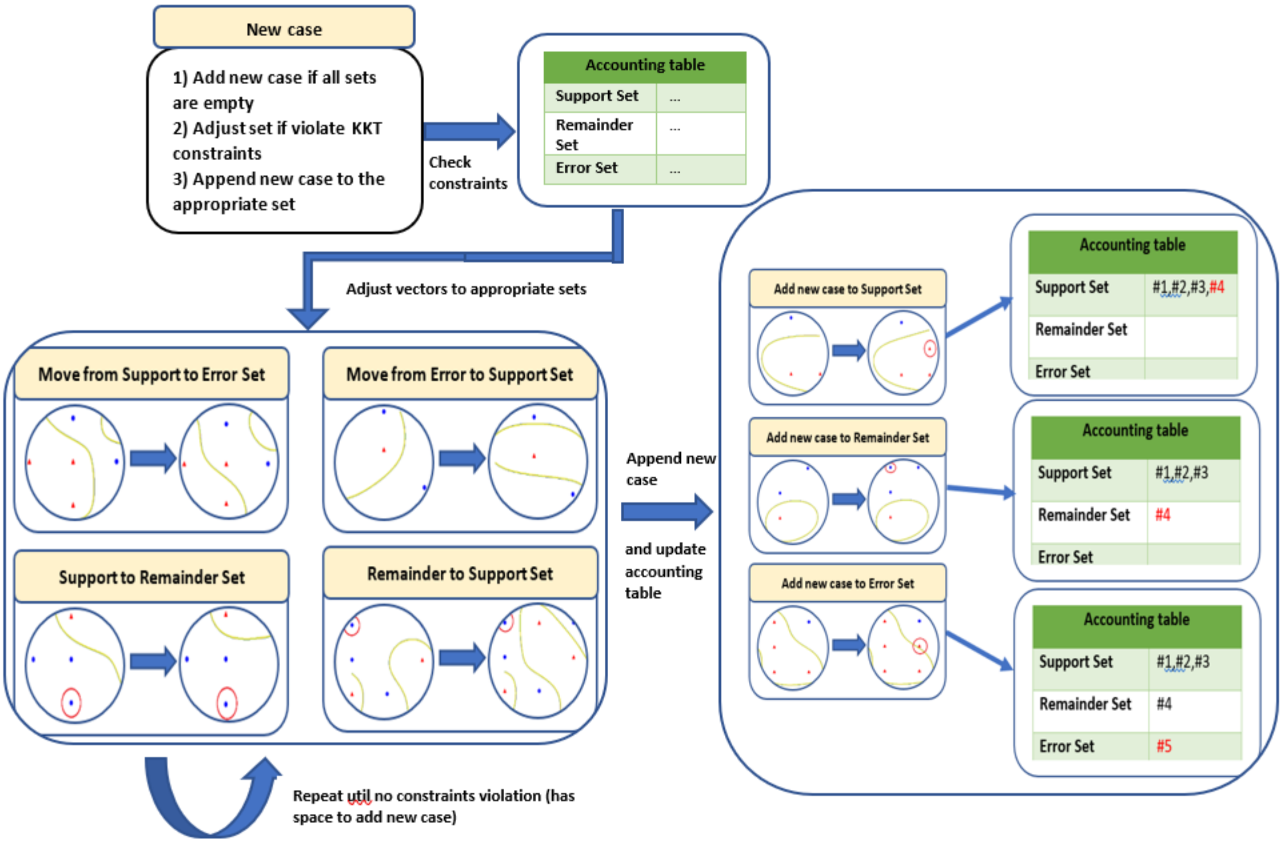
$$\tilde{Q} = \begin{bmatrix} Q^{-1} & \eta_k \\ \eta_k^T & K_{kk} \end{bmatrix}^{-1} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\kappa} \begin{bmatrix} \beta_k \\ 1 \end{bmatrix} \begin{bmatrix} \beta_k^T & 1 \end{bmatrix} \tag{3.4}$$

in which $\kappa = K_{kk} - \eta_k^T Q \eta_k$

Hence, the operation time for updating and removing is quadratic in the size of Q [104]. Although ISVM describes exactly the process of online-learning, the computation quickly escalates with the number of learned data. According to the learning mechanism, ISVM has to record the entire data and the belonging status. The operation time of ISVM is boosted rapidly at the very first samples and degrades linearly in later iterations.

3.2.4 Online LASVM

LASVM is a semi-online training mechanism that is also applicable to other kernel classifiers [105].The approach solves the large margin classifier problem by utilizing the sequential searching direction of SMO. The direction, called $\tau - violatingpair$, is determined by moving


New case
1) Add new case if all sets are empty
2) Adjust set if violate KKT constraints
3) Append new case to the appropriate set
Check constraints
Adjust vectors to appropriate sets
Move from Support to Error Set
Move from Error to Support Set
Support to Remainder Set
Remainder to Support Set
Repeat util no constraints violation (has space to add new case)
Append new case
and update accounting table
Add new case to Support Set
Add new case to Remainder Set
Add new case to Error Set

Accounting table	
Support Set	...
Remainder Set	...
Error Set	...

Accounting table	
Support Set	#1,#2,#3,#4
Remainder Set	
Error Set	

Accounting table	
Support Set	#1,#2,#3
Remainder Set	#4
Error Set	

Accounting table	
Support Set	#1,#2,#3
Remainder Set	#4
Error Set	#5

Fig. 3.4 The process of online SVM - the new cases is added after adjusting the observed vectors to the appropriate sets with respect to the constraints

along a pair of samples (i, j) as long as it expands the margin without violating any constraint.

$$\tau - violatingpair(i, j) \Leftrightarrow \begin{cases} \alpha_i < max(0, Cy_i) \\ \alpha_j > max(0, Cy_j) \\ g_i - g_j > \tau \end{cases} \tag{3.5}$$

The convergence of solution is achieved by alternating two phases of direction search, namely PROCESS, and REPROCESS. In PROCESS phase, a potential vector (i) is considered to be appended into the current kernel. Initially, the new sample is added to the support set. Then, the process identifies its second $\tau - violatingpair$ (j) from the support set S that has the greatest gradient. The searching directions of existing support vectors are shifted accordingly

$$\lambda = min(\frac{g_i - gj}{K_{ii} + K_{jj} - 2K_{ij}}, max(0, Cy_i - \alpha_i), \alpha_j - min(0, Cy_j)) \tag{3.6}$$

$$\alpha_i = \alpha_i + \lambda \ ; \ \alpha_j = \alpha_j - \lambda \tag{3.7}$$

$$g_s = g_s - \lambda(K_{is} - K_{js}) \quad \forall s \in S \tag{3.8}$$

On the contrary, instead of preserving all of the observed data in ISVM, LASVM adopts the removal mechanism to efficiently manage the storage capacity. The elimination procedure is achieved in the REPROCESS. This process, first, repeats the searching for $\tau - violatingpair$ (i, j) as in the previous description.

$i = argmax_{s \in S} \quad where \quad \alpha_s < max(0, Cy_s)$

$j = argmin_{s \in S} \quad where \quad \alpha_s < min(0, Cy_s)$

Upon completion of the adjustment, any support vectors that exceed the new bounding constraint –defined by the pair (i,j) -are pruned. At the end of elimination, the bias term of decision function and the gradient of $\tau - violatingpair\ (i, j)$ are recomputed.

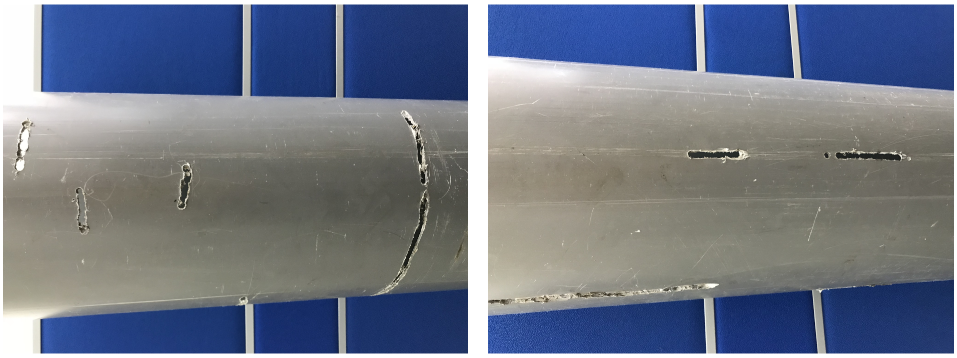
$b = \frac{g_i + g_j}{2}$

$\delta = g_i + g_j$

LASVM successfully combines online and offline-learning in its processes. In the online iterations, the adding and removing procedure are achieved consequently through PROCESS and REPROCESS. However, to reach a better solution, additional REPROCESS steps must be applied gradually until there is no further $\tau - violatingpair$, defined as $\lambda < \tau$. The finishing step is performed as offline-learning since it is achieved after the entire batch has been observed.

In the practical implementation, the online iteration of LASVM is learned through in epochs. An epoch is defined by a sequence of shuffle training example. Running one epoch involves the computation of online setup. To ensure the accuracy of the output model, the finishing step is applied after a predefined number of epochs. Multiple epochs are combined as a stochastic optimization approach [105]. The report from different benchmark dataset indicates the competitive accuracy with common offline SVM in a single training. Moreover, LASVM requires less memory and dominates the common SVM solvers in training time.

According to the result, it is safe to conclude that the learning mode did not have great affect in terms of accuracy given the dataset. On the other hand, there is a slightly advantage of the online mode on the time performance. Thus, at first the learning quality is reserved as replacing online for the traditional offline mode. Secondly, with respect to the solving SVM mechanism, the online mode produces the booking keeping procedure which is a valuable information when implements into a CBR system. If in the offline mode, only the list of support vector is stored, the solving procedure in online also extends the list to the remainder and the error list. This categorization benefits the case ranking process in CBR by estimating



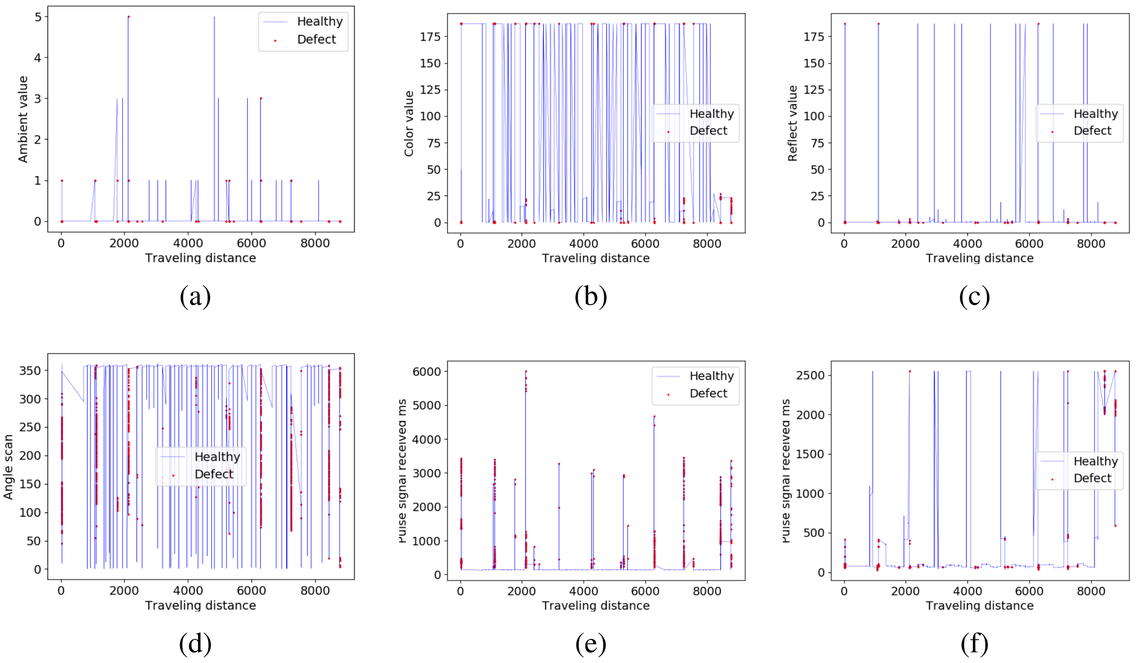
(a) circumferential defects s (b) axial defects

Fig. 3.5 Experimental defect pipe design for robot inspection mission

the case similarity. In addition, the set status also contribute to the reusable decision by evaluate the confidence. Furthermore, the procedure also includes the log of changing sets. Although this step consumes more capacity, by referencing the set movement, the trust factor is applicable to the revise and retain of the sample. Hence, as integrating in the CBR system, SVM with online mode not only preserves the quality but also guarantee the transferring information among phases.

3.3 The process of data collection

The experiment processes can be split into inspecting the pipe to collect fusion sensory data and sending results to the analysis center for detection. As the scope of this work concentrates on analyzing the efficiency of fusion sensors, the implementation of a real-time robot, in which the trained model has been embedded in the robot control unit and transmit an alarming signal whenever facing the flaw, would not be covered. Instead, the processes of data collection and data classification were separated. According to the pipe description figures 3.5, the pipeline route organizes into three distinct sections to assess the ability of capturing defects in different types. The first 1.5 meters consists of circumferential defects, while the next 1.5 meters comprehends axial defects 3.5. The defects positions are measured accurately for later labeling task. The last 1 meter is a flawless pipe which represents a healthy sample. Besides, an additional 1.5 meters pipe preserved for the test case. To evaluate the generalization of learning model, test pipe composes of both circumferential and axial defects, presented in random order. Furthermore, the test pipe also includes a junction which enlarges the diameter (17.7 cm) in compare with the one in training pipe (16.5 cm)

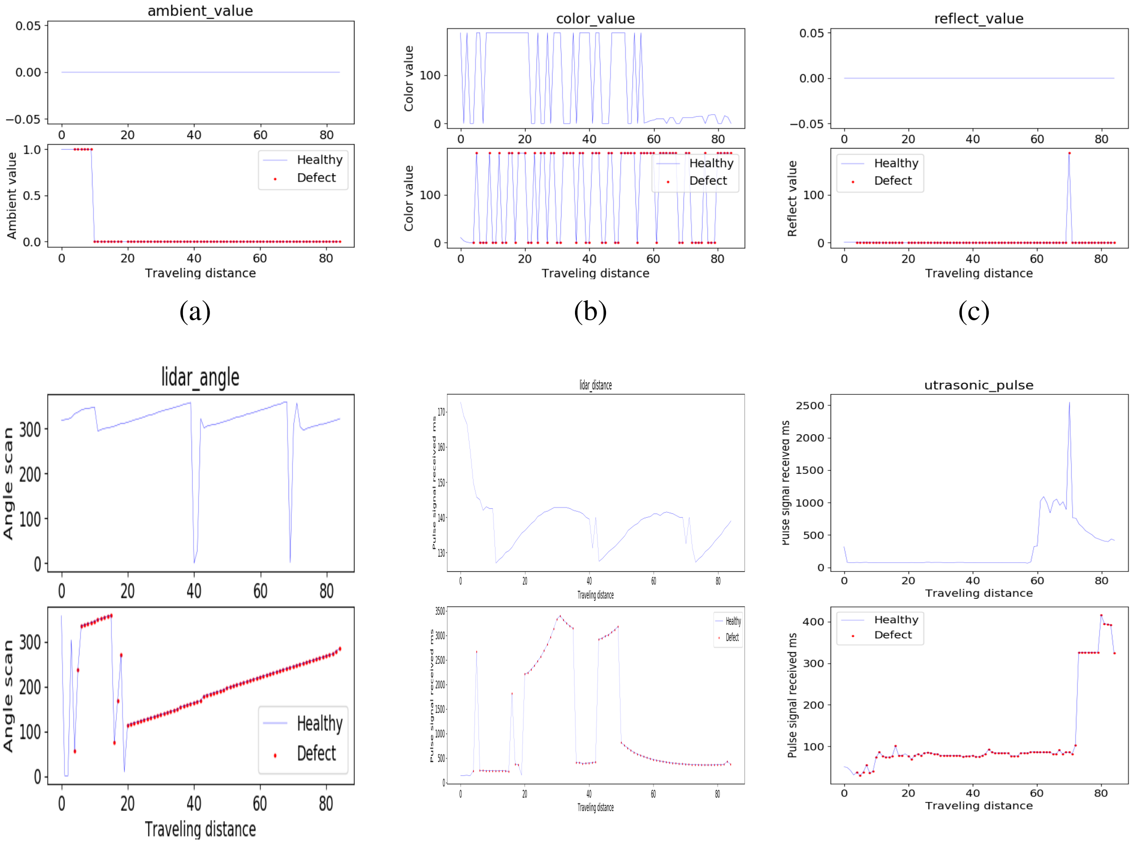


(a) (b) (c)

(d) (e) (f)

Fig. 3.6 The completed data after completing 1 voyage (8-meters) (a) ambient light, (b) color, (c) reflected light (d) lidar angle (e) lidar distance, (f) ultrasonic pulse

The figure 3.6 indicates the scanning patterns of attached sensors inside the pipe. Particularly, the x-axis indicates the traveling distance (in millimeter) while the y-axis indicates the scanning value. A complete voyage consists of a forward (4 meters) and a backward route (4 meters). According to the setup, the first three meters and the last three meters result in considerable defects. With the same set of sensors, the comparison is en The figure represents full scanning patterns after completing a voyage. To demonstrate the appearance of defects, small chunks of scanning data (approximately first 8-cm) were described in the second row. The blue line denotes normal signals while the red scatters mark the defects



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3.7 comparison on healthy signal (upper) and defect signal (lower) in a small chunk of 8cm (a) ambient light, (b) color, (c) travelling light (d) lidar angle (e) lidar distance, (f) ultrasonic pulse

cases. Soft margin C-SVM and ν SVM have been applied to classified the observed data. Both C-SVM and ν SVM were optimized by applying GridSearch. The choices of tuning hyper-parameters include kernel types and the related error control parameter C or ν. The kernel coefficient also optimized according to its type (degree for polynomial kernel, and gamma for Gaussian kernel). Comparison between healthy and flaw signal indicates interesting points. By utilizing ambient value, the lights sensors can recognize defect at the first 10 cases. As combining with the reflected value, it slightly captures more cases from the 70^{th} signal. On the other hand, the color sensor indicates the abnormal pattern from case 10^{th} - case 20^{th} and also case 60^{th}-80^{th}. Finally, the abnormal pattern was found by LiDAR sensor, or ultrasonic sensor at cases 20^{th}-60^{th}. This derivation encourages the use of sensor compilation which utilize the strength of all sensors to detect defects thoroughly.

3.4 The implementation in CBR system

In general, CBR system of our research consists of three main components: a case representation, an evaluation engine, and a cycle engine. The diagram in Figure 3.8 simplifies the design of the proposed CBR system.

3.4.1 Case representation

The case representation is described in either feature-based or object-based. The common learning model favors the feature-based representation due to its simplicity. Single sample maps the relevant features directly to the input. In addition, using feature-based eases the complication of the structure and lessen the storage capacity. However, in SVM-CBR, the case is represented in an object-based form. This representation successfully describes different structure with hierarchy grouping from each sensor. In particular, the signal of color, and ultrasonic sensor in each sample are denoted as a series, while the signal from LiDAR sensors is indicated as a single instance.

The generic CBR system is designed in a object-oriented structure. The abstract layer consists of the generic model representation and generic phases of CBR system. This layer keeps the system reusable for other problems. The models are kept at a case instance which denotes a specific case. The case base serves as the collector of cases. Using object-based representation, the problem, and solution instances described in case object are implemented from the generic CBRclass object.

The specific sensory data scanning is extended from the case abstract layer. Similarly, SensoryDataCaseBase extends the CaseBase layer. This design gains the following benefit

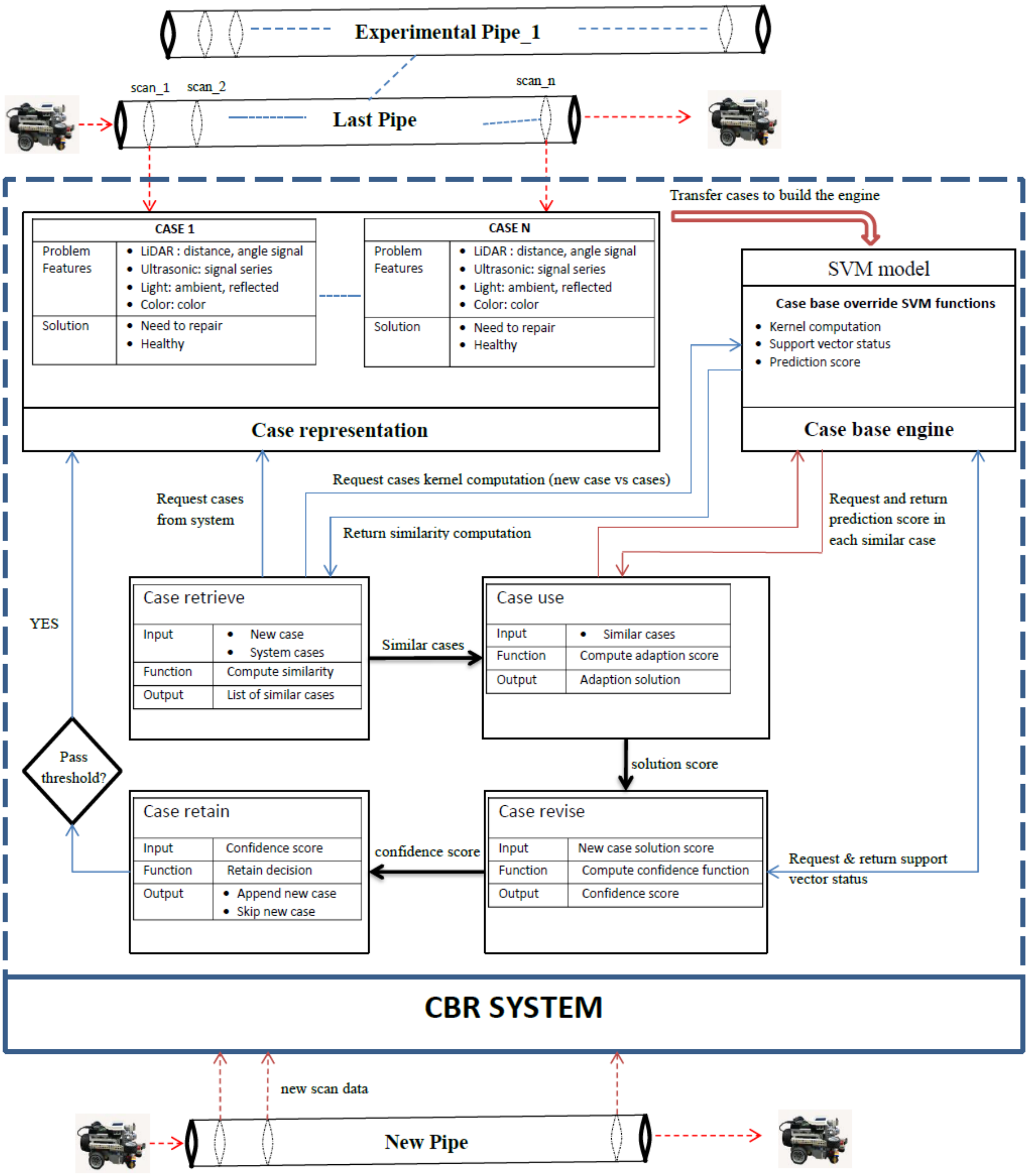


Fig. 3.8 The complete CBR diagram govern by online SVM for pipe defect detection system

- Sensory data can be split according to each sensor as long as they extend the Case and CaseBase layer.
- If the design required using different sensor for scanning, the system can build another sensory case without affecting the current case representation system.
- The design is sufficient to be applied to different datasets to determine the efficiency of CBR system.

3.4.2 CBR cycle

Case retrieval

Many existing systems favor two decisions for the case retrieval phase [106–108]. The first option is to get k nearest neighbor while the other option is to calculate cases distance. As the defined in this research was developed with SVM as the principal engine, all of the phases will utilize the computation of SVM. Instead of implementing the above techniques, this CBR system exploits the kernel calculation. In another point of view, the kernels mapping function conferred as the similarity function. Particularly, kernel mapping of a vector itself possesses the highest value (or the most similar). Therefore, the computed kernel matrix is employed to measure conditions concerning case retrieval. In this system, five types of kernels registered.

Since different similarity functions produce the different ranges of score, the similar outputs are scaled to preserve the consistency when applying the cut-off threshold. The system scaling alternatives incorporate min-max scale, quantile scale, and standard scale. Min-max scale and quantile scale use uniform distribution to control the similarity result within a range from zero to one. Meanwhile, the standard scale and quantile scale use normal distribution to keep the output in a range of Gaussian bell with a mean of zero and a standard deviation of one. The scaler supports expert in addressing the threshold that mostly accommodates with the experience. The list of predefined maximum similar cases that exceed the threshold will be chosen to nominate the solution.

At first, a new case is encoded according to its six-degree of freedom which annotates the position of the scanned points. In the default settings, similarity function directly uses SVM score with RBF kernel and cutoff at maximum nine cases. The similarity scores are normalized with quantile transformer.

Case reuse

The traditional ML models simply determine the output from its trained result. In order to increase the accuracy, multiple models have been ensembled to contribute to the solution. Although ensemble technique is considered as a good practice to boost up the performance, practical implementations into real problem require an exhaustive combination to determine the work.

In large and complex CBR system, the reuse phase is built with more adaption techniques. Generally, the adaption approaches may have the following designs. Direction adaption, or null adaption, simply applies the output of the most similar case as the output of the new problem. In substitutional adaption, the system uses a specific domain expert function that adjusts the prediction as closely as the retrieved cases.

In our SVM-CBR system, substitutional adaption adopts voting technique with regards to the level of similarity and confidence. Reversed Gaussian and absolute arcsine functions are adopted to estimate the prediction confidence. As implemented into the system, the expert is authorized to select the most suitable function. The level of confidence is controlled by the gamma parameter.

$$conf(x) = 1 - e^{-\gamma x^2} \qquad (reversed\ Gaussian) \tag{3.9}$$

$$conf(x) = -1 + \frac{e^{\gamma x} + e^{-\gamma x}}{2} \qquad (absolute\ arcsine) \tag{3.10}$$

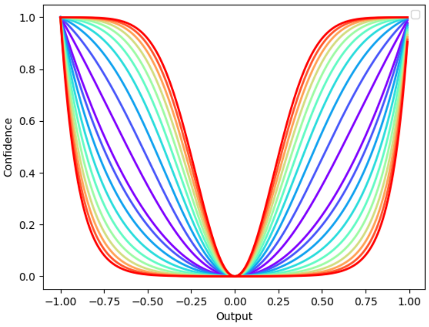


Fig. 3.9 Plotting of confidence functions using reversed Gaussian and absolute arcsine

In general, confidence is designed to satisfy the following requirements. The confidence decreases if the prediction result approaches zero and increases if the output of prediction reaches the boundary prediction $[\pm 1]$. The confidence conditions reflect the same property of SVM whereas a more certain vector determines if its prediction is far from the bound– fall to the remainder set. In reversed Gaussian function, the confidence accelerates as long as the prediction exceeds the ambiguous state at zero value while in absolute arc sine function, the confidence increases rapidly as reaching the boundary value. The Fig 3.9 describes the changes of confidence levels in different gamma levels.

The red band indicates high gamma and violet band denotes small gamma value. The reversed cone shapes illustrate the property of reversed Gaussian function in which the level of confidence increases as long as increasing the gamma parameter. On the contrary, the U-shapes define the behavior of absolute arcsine function whereas the level of confidence and gamma parameter are inversely proportional.

Case revise

Only a few CBR frameworks realize an automatic mechanism for the revised phase. Instead, most of the existing frameworks require manual verification from experts before registering a new case solution. In this system, a semi-auto revise approach is implemented. In particular, the system provides two options for the expert to decide. In the first option, the list of retrieval cases monitors according to their belonging status in the SVM model. As in SVM convention, the vectors are assigned to either one of the following sets: support set (directly define the margin), remainder set (correctly classify and far from the margin – less contribution), and error set (misclassify). A case is obligated to revise if the sum of vectors distances from the remainder set is considerably less than the sum from the other sets.

In the second option, all sets from the online SVR models are also exploited. However, instead of considering only the retrieval vectors, the new case will compare its distance with every vector in each set. Similarly, if the new case is not close to the remainder set, a revision is advised. The decision is also based on the level of confidence. Whenever a case approximates the remainder set, the decision easily concludes as the violation of KKT conditions problematically occurs if new samples attach. On the contrary, vectors adjacent to the support set should be examined further as they may affect the generalization.

Case retain

If the prediction exceeds a specific threshold level or the case is close to the remainder set, that case will certainly be added as a new case. The case based system is updated with the new case with the judgment given from the previous solution as the benchmark. This new case is also updated into the core engine of online SVM model. The model will run an updated training incrementally. While training incrementally, the CBR system still functions to predict new cases. On the contrary, vectors that are close to the support set should be examined further as they may affect the generalization.

3.4.3 Integrate CBR pipeline

As mentioned, the system need to be constructed in generic implementation of the CBR system to maximize the possibility of reusable components. At the same time, the system is provided with a graphical user interface as a web based service to be adaptability friendly and intuitive. This intent has resulted in a number of technical decisions that will be outlined in this section.

The object-based representation is designed as the following scheme: the first layer (located in core/internal repr) represents generic case structures (CBRclass, Case, and CaseBase)). These python classes are able to represent virtually any object hierarchy independent of the domain. In a way this can be seen as the redefined Object Class specific for CBR Systems. The second layer, which is more domain specific, builds on top of the created in core/internal representative. The classes are Sensor and SensorsCaseBase, located in the module wrapper.py. The Sensor class wraps the abstract Case structure whereas the SensorsCaseBase extends the CaseBase.

This way, The system is able to capitalize on the benefits that object oriented case representation brings about without significantly sacrificing the generality. The system deals specifically with the Sensor and SensorsCaseBase objects which in turn build on top of the generic classes. If need arises to build another CBR application and we choose to employ an object-oriented case representation, we would only need to provide alternative wrapper objects around the Case and CaseBase. The underlying infrastructure – constructing and managing the object hierarchy of the domain entities – could be used without any modifications.

Similarly, an effort has been made to approach the implementation of the individual CBR phases – retrieve, adapt, revise, and retain – in a generic way. This could best be illustrated with an example. Class Case, for instance defines a method phases.py retrieve(casebase, case, similarity function, thr, max cases) whose arguments are: 1. a collection of all cases, 2. a case for which similar cases should be found, 3. a similarity function, 4. a threshold value to filter out similar cases, and 5. the maximum number of cases to retrieve However, the most crucial argument, the similarity function, is not even defined in the generic representation of the case. Case only provides a functionality to apply the function, if one is passed, to the collection of cases and filters the results according to the threshold value passed thereafter. The similarity function is only defined in the wrapper.py module, that in addition to defining objects that are particular to our domain – Sensor or SensorsCaseBase – also provides implementations of the CBR phases in accordance to the prediction requirements.

Correspondingly, the other phases of our CBR systems also abstract away from the particular implementation of the relevant functions which are provided by the wrapper module. The generic functions and their respective implementations are listed below.

phases.py reuse(..., adaptation function, specific function): takes an adaptation function which can be any main adaptation technique as well as a specific function which is problem related. While we can pick one of the several adaptation functions provided in the generic module phases.py, the specific function is defined in the wrapper.py module.

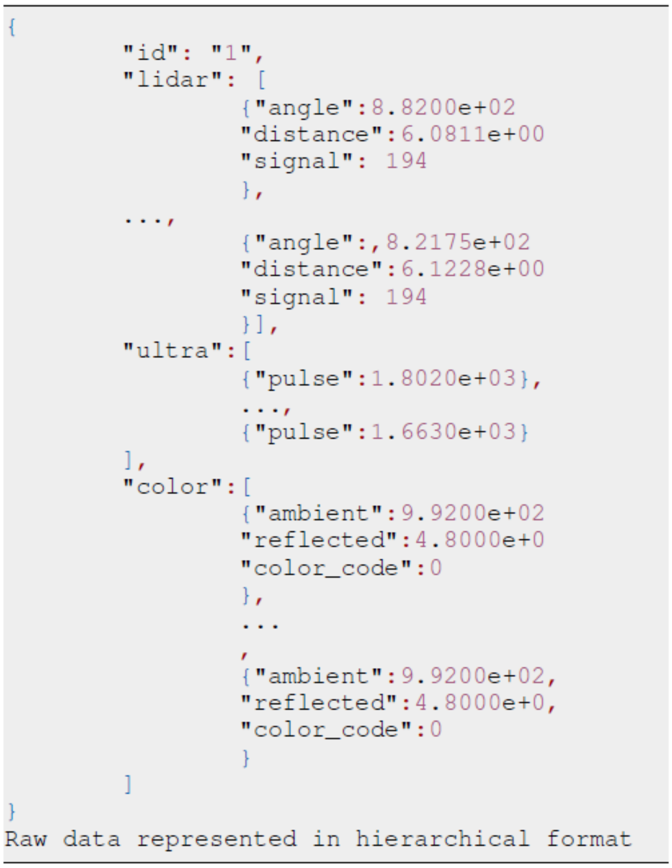
phases.py revise(..., expert function, ...): takes an expert function which is again defined in the wrapper.py and specific to our scenario. As can be seen, if we were to apply the CBR application developed to a different domain, we would be able to reuse the generic machinery and the code that commands the CBR cycle without any modification. The only task at hand would be to provide specific implementations of the similarity, adaptation and expert functions as well as devise sensible threshold values. These can thereafter be passed over to the CBR framework and expected to work properly.

Chapter 4

Results and Analysis

4.1 The experimental setup

The experiments are designed as follows. For all experiments, the core online SVM engine of CBR is only trained on a small sample size (10% of the dataset). This setup configuration tries to replicate the condition while real-time inspects pipeline. The performance of CBR is compared with other ensemble methods such as boosting techniques, random forest and the offline SVM which is trained on a larger data size. The model identifies its optimal hyper parameters through grid search. The list of all comparison approaches is organized as the following. The first approach CBR_full represents the performance of CBR system as described in chapter 4 with all learning functions being recalled in every phase. On the other hand, the CBR_half eliminates the use of confidence function and SVM output prediction in the reuse phase. The reuse phase of this CBR_half system engages with null adaption and major voting mechanism. Consequently, support vector status is also obligated to achieve the revised decision. Alternately, the revision is performed by examining the evaluation metric of SVM if the new case is appended. The two ensemble techniques, Random Forest and Model Voting, are nominated for comparison.



```
{
        "id": "1",
        "lidar": [
                {"angle":8.8200e+02
                "distance":6.0811e+00
                "signal": 194
                },
        ...,
                {"angle":,8.2175e+02
                "distance":6.1228e+00
                "signal": 194
                }],
        "ultra":[
                {"pulse":1.8020e+03},
                ...,
                {"pulse":1.6630e+03}
        ],
        "color":[
                {"ambient":9.9200e+02
                "reflected":4.8000e+0
                "color_code":0
                },
                ...
                ,
                {"ambient":9.9200e+02,
                "reflected":4.8000e+0,
                "color_code":0
                }
        ]
}
Raw data represented in hierarchical format
```

Fig. 4.1 The raw data represented in hierarchical format

The obtained data is presented in hierarchical structure as described in below JSON format 4.2. The parent node presents the 6-degree freedom to specify the inspection gauge postion. Three children contain the subsets of different signals. An additional feature preprocessing step has been implemented to convert signal into appropriate tabular format for later process. The lidar signals describes the scanning angle and the relevant distance. In a specific position, the dataset indicates the scanning distance from robot source to the pipe surface. As the number of received signals at different locations are not identical and also with the aim reduce the number of features, min, max, mean, and median values are selected to represent at a single location. Similarly, fundamental properties have been extracted to represent the data signal of the color sensor [109]. With ultrasonic signals, the received data at current positions are indicated by a series of pulses sent and received. This is an 11-pulse signal series as specified in the sensors announcement. To concatenate with the tabular format dataset, 11 pulses were converted as 11 distinct attributes. To control the balance of input from each sensor, weighted attributes are employed. After exploring data analysis on ultrasonic signals, mean and variance values have not been used as the changes are rarely small to recognize. However, information such as the signal trend by measuring the differences and slope among adjacent entry points are adopted.

The design of comparison models is aimed to clarify the following inferences. Through the comparison between CBR_full and CBR_half, the experiment validates the improvement when exploiting the learning information in all phases. The efficiency of using online-learning mechanism in CBR system to augment the new case can be achieved by comparing the CBR_full with the original offline SVM. The comparison between ensemble approaches and CBR_full is to determine if the experience achieves from the retained cases can fairly compete with the compilation of various models.

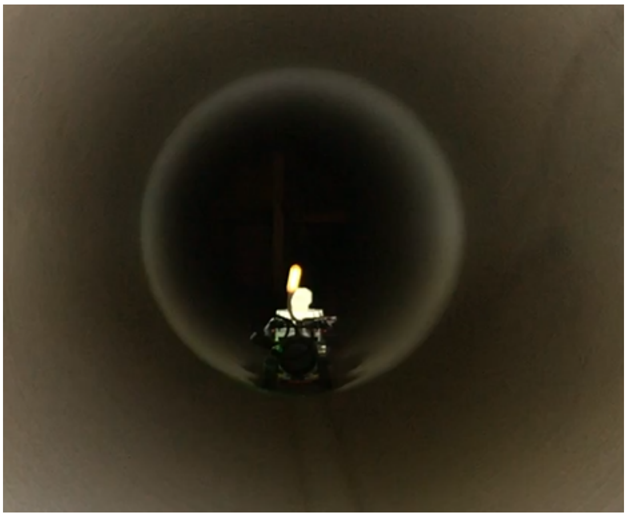
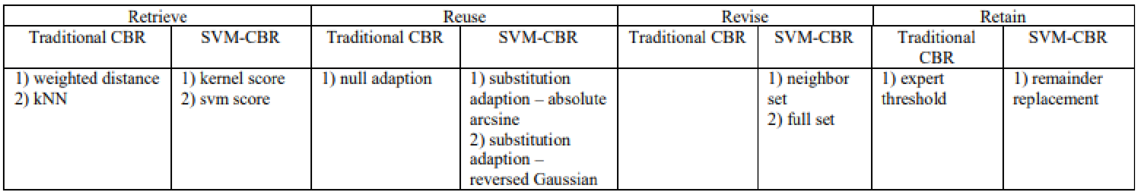


Fig. 4.2 The inspection gauge travels inside the pipe to transmit scanning signal for CBR system

4.2 Design of Comparison Analysis

In order to justify the remarkable performance, phases in conventional CBR system is decomposed to compare with relevant phases in the proposed CBR. As different approaches are applicable in each phase of SVM-CBR, all available solutions have been investigated with the conventional CBR. Finally, the whole pipeline is analyzed with the conventional-CBR. The table 4.3 reports the result of comparison. Specifically, in CBR-reuse phase, the popular approaches, kNN and weighted distanced are compared with the proposed estimations, which computes the kernel scores and svm score. The family of Minkowski's distance of level-1 Manhattan and level-2 Euclidean are used during the computation of conventional model. Regarding the reuse phase, conventional threshold filtering is competed with the confidence score evaluate through reverse Gaussian and absolute arcsine function. The observation of different functions is set in context of various adaptions - null and substitute adaption. As stated in the review of existing CBR system, revise phase normally assigns for domain expert to confirm the decision. To obtain the decision automatically, the confidence interval score is



Retrieve		Reuse		Revise		Retain	
Traditional CBR	SVM-CBR	Traditional CBR	SVM-CBR	Traditional CBR	SVM-CBR	Traditional CBR	SVM-CBR
1) weighted distance 2) kNN	1) kernel score 2) svm score	1) null adaption	1) substitution adaption – absolute arcsine 2) substitution adaption – reversed Gaussian		1) neighbor set 2) full set	1) expert threshold	1) remainder replacement

Fig. 4.3 The comparison of phase to phase between conventional CBR and SVM-CBR

used as an alternative solution. Considering as a fundamental technique, comparison with this settings is arguable.

In the first phase of comparison, Silhouette analysis is calculated as the selection quality index. Particularly, the list of selected cases in each approach is appeared a cluster, Silhouette is refers to examine the cohesion and separation scores. A new case is considered as another cluster. Silhouette scores is computed within these clusters. The process is repeated with the list of all cases and extract its means and covariance for analysis. The metric is bounded by the [-1, 1] interval. A score, which engages to this bound, reflects a completed distinguishable separation. On the contrary, the clustering is considered as overlapping once the Silhouette score progress to zero. In this case, the distances between among different clusters is minor.

To assess the efficiency of reuse solution, the classification metrics are used to compare with the ground truth. Since the defect detection CBR favors the identification of all defects, the elected metrics are composed to match this perception. In addition, the metrics must be effective in the case of imbalance data set. Therefore, due to the focus on the important of discovering positive case, the F Scores and PR AUC are preferred to compare the performance in the reuse phase. Since the comparison uses the final output, the later phases is combined to evaluate with other ensemble methods.

Although above phases performance are measurable independently, comparison in the revise and retain phase need to be established in a recurrent process. GridSearch is also executed on five hyper parameters: number of max cases, similarity function, confidence threshold; confidence function gamma, and retain threshold.

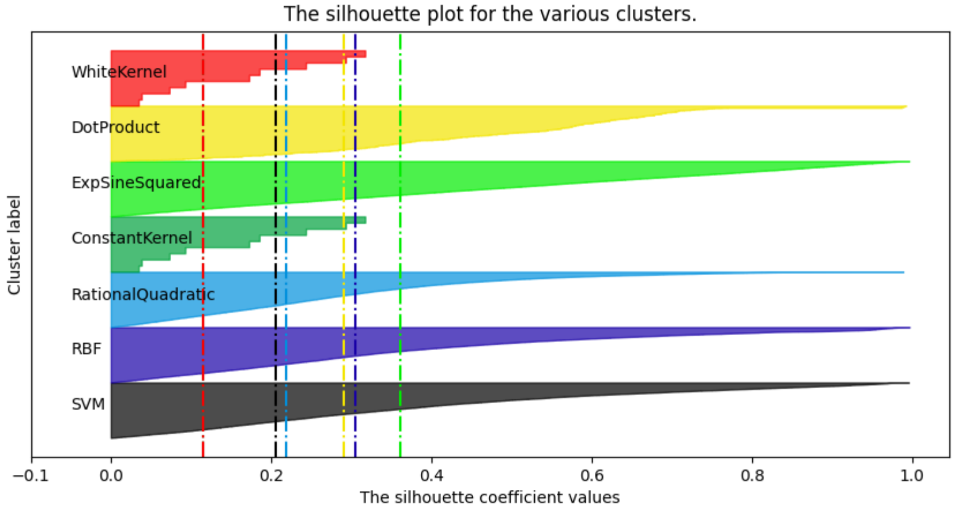


Fig. 4.4 Silhouette score with the use of different kernels for the proposed retrieval phase

4.3 Discussion on single phase

The experimental results on the efficiency of available clusters in the proposed retrieval phase is shown in figure 4.4. According to the results, six types of kernel together with the svm regression score are assessed with the Silhouette coefficient. In the figure, a cluster represents the sample silhouette scores of a specific kernel computation. Each new case retrieves nine cases from the case base and the process is applied for all new cases to achieve the final silhouette score. The dashed line is the mean silhouette score of each cluster. To achieve a apparent visualization, absolution values is also computed to convert the interval from[-1, 1] to [0, 1].

The obtained results can be summarized into 4 groups. The first group includes white kernel and constant kernel that produces similar result as white kernel is transformed into constant kernel according to the kernel formula with this dataset. This group has achieved the best result according to the perception of Silhouette concept. Particularly, this group score closes to zero rather than the other groups at around 0.1. As the score indicates the cohesion between new case and selected cases, the small score implies the proximity of the

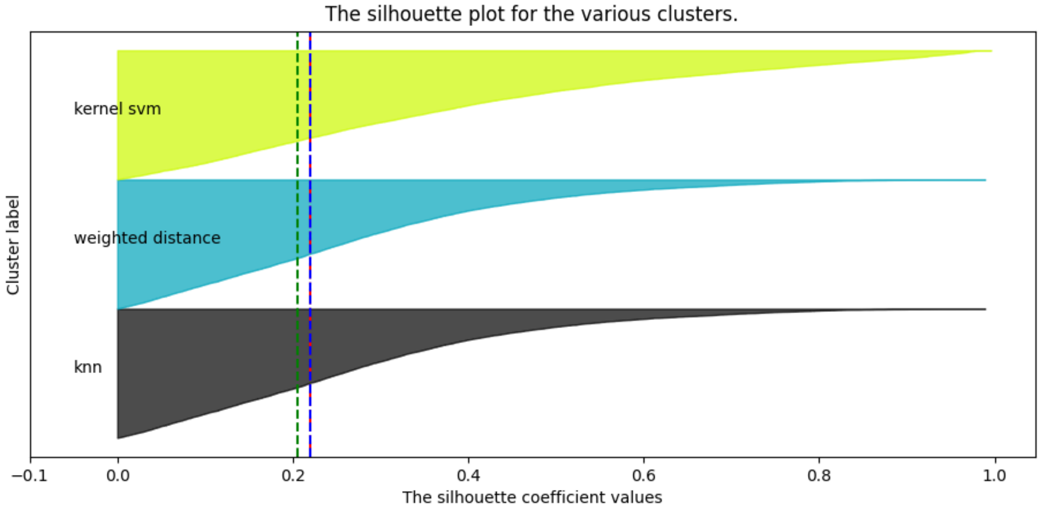


Fig. 4.5 Silhouette score when using SVM and traditional retrieval approaches

selected cases. The second group covers SVM score as well as rational quadratic kernel with silhouette score approximately 0.2. The last two groups are categorized into dot product + RBF kernel and exponential sine squared kernel. The figure suggests the dominance of the first and second group against the others as the scores are almost double. As rational quadratic kernel is the combination of various squared exponential kernel with different length scale, the output scores reflect a more distinguishable observation and close to the SVM score. Despite a better result, white kernel and constant kernel is not selected for further comparison due to the limitation of application in general. In specific, constant kernel heavily depend on the configuration of selected threshold and becomes inefficient if applied in other problems. Hence, in the later comparison, SVM score is endorsed to compete with traditional retrieval approaches.

The figure 4.5 illustrates the distribution of sample Silhouette scores. The Silhouette scores is defined as the following formula

	KNN	Weighted Distance	SVM
mean	0.219535	0.219509	0.206052
standard deviation	0.079098	0.079029	0.141730
minimum	0.000194	0.000194	0.206052
first quartile	0.174	0.174	0.090175
second quartile	0.238	0.238	0.196
third quartile	0.278	0.278	0.29425
maximum	0.440000	0.440000	0.820000

Table 4.1 The result of different retrieval phase approaches

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ \frac{b(i)}{i(i)} - 1, & \text{if } a(i) > b(i) > \end{cases} \tag{4.1}$$

where $a(i)$: the average distance between *i* and all other data within the same cluster

$b(i)$: the lowest average distance of *i* to all points in any other cluster, of which *i* is not a member

using kernel svm, weighted distance and KNN in the retrieval phases. According to this figure, the mean scores of the two common retrieval approaches - KNN and weighted distance - are almost similar (rounded by 4 division). Consequently, these scores are slightly prevailed by the SVM scores. Despite the difference between the propose retrieval and the common solutions is insignificant according to the figure, further statistics analysis indicates the overwhelming of the SVM score. Table 4.1 summarize the details in the comparison in different aspects. Although the mean silhouette are trivial - only 0.01 difference, examining the distribution of each quartile denotes a better selection of similar cases. Specifically, SVM reaches better milestones the first two quartile, 0.09 as compare with 0.17 in the first quartile and 0.196 with 0.238 in the second quartile. In the third quartile, common approaches have better scores at 0.278, whereas SVM score is around 0.294. However, the variation is smaller than the previous two quartiles. Remarkably, the maximum statistics indicates a moment when selected case from SVM are distinguishable from the input new case. However, examining standard deviation and observing the figure plot, this occurrence arises occasionally as the maximum value and standard deviation is almost double while the mean and the first three quartiles are similar. Hence, the conclusion of a slightly advantage of SVM in retrieval phase is achieved.

Gamma value	1	2	3
Arcsine	0.73	0.79	0.85
Reverse-Gaussian	0.67	0.73	0.81
Threshold	0.78	0.56	0.22

Table 4.2 Comparison of the percentage of reusable cases when applying confidence function at different gamma value and the traditional cutoff threshold at the third quartile

Table 4.2 describe the comparison of the proposed reused phase and the traditional reuse method. According to the result, The confidence function becomes more flexible as comparing with the traditional threshold method by the percentage of filtered case. If the traditional threshold serve as a single cutoff line, the confidence functions implicate as the fuzzy function which becomes more robust. In this experiment, the threshold cutoff is defined at the first, the second and the third quartile of the similarity scores. In addition, the output of the system also indicates the level of confidence to support the expert decision.

Since the traditional CBR requires an expert domain to defines the revise function, the comparison is a bit unfair as this work only apply a random selection for the revised. Subsequently, a random function is also applied in the retain phase to achieve the decision as in traditional CBR. Hence, the result is not reported as the outperform of the proposed CBR is trivial with this setting.

4.4 Discussion on SVM-CBR system

In this sections, the entire SVM-CBR is compared instead of single-phase. The table 4.3 described eight settings of the SVM-CBR. In particular, the retrieval phases includes of kernel estimation and SVM estimation to compute the similarity. The null adaption and substitutional adaption are queried. In the revise phase, the use of all support vector or n-closest vector are employed. The mean result of these settings are used to compare with other methods in table 4.4 The experiments indicate that only similarity function, and confidence function gamma have been involved in the output decision. On the other hand, the number of cases, confidence threshold and acceptance threshold do not have effects on the justification. Intensive examination of these three hyper parameters indicates the experience of saturation.

1. Confidence threshold: The threshold was designed to search for an optimum value inside the interval [0.5-0.9] with offset 0.1. However, the level of confidence output after applying confidence function is extremely high, almost close to 1 - 1e-3. The main reasons are the dominant of the number of closely certain cases, and the high output from past cases prediction. As a result, any decision outputs from the reuse phase are accepted.

2. Retain threshold: Similarly, the reason for not making any changes when searching this parameter is the dominance of the output value above the threshold. As a result, to observe the changes, the searching grid has to be set very high and is scaled with considerably small offset.

3. Number of retrieved cases: Since the confidence level is substantially high with the error of 1e-4, the number of cases should be in thousands in order to make changes.

It is apparent from Table 4.4 that the performance of using full CBR pipeline dominates the common CBR practices. In most of the experiments, the improvement is significant with almost 10% in accuracy from 87% to 98%. Particularly, we observed that the tuning of confidence function gamma lead to the increasing of 2% in the accuracy. The appropriate choice of retrieval function also boosted the performance dramatically. For this specific dataset, SVM hypothesis score and RBF kernel function are reported as the highest score, while other simple dot product functions such as Dot Product and, Polynomial kernel produce poor results.

The next comparison of the metrics was is with the advantage of using online-learning mechanism. Although the original offline SVM has shown better results, online learning is still able to fairly compete with offline learning due to the following facts. The number of the initial learning samples in online CBR is much smaller than the training size of offline learning. The success of using small sample size as experience in section 4 encourages the handicap of online CBR setup. Furthermore, CBR mechanism allows expanding the case-based repository through testing, though the number of retained cases are small. Since the experiments are conducted on a medium size dataset, the improvement of around 1% does not strongly imply the overwhelming of offline method. Eventually, it is interesting to note that the output result of CBR pipeline is close to and also surpasses the original offline SVM in the case of small learning size (28% of the total dataset) and much greater training size (73% of the total dataset). However, contrary to expectations, the proposed pipeline scheme is failed to attain the accuracy of ensemble techniques. This result leaves an open opportunity for designing an advanced system that replaces the powerful ensemble procedure in future work. Regardless of the under performance, the complete SVM-CBR has accomplished the research questions. Firstly,the system has overwhelmed any combination setup of CBR. Especially in the three latter phases, the proposed system slightly increase the accuracy thanks to the following advantages. The confidence function serves as a filter to eliminate uncertainty cases which diverse the decision. Secondly, the system can automatically defined the reexamined procedure without the need of using expensive domain expert function. Particularly, even if the case gain has strong confidence, the system still has a level of uncertainty by comparing with the support vector set status. Thus, the system advance traditional CBR since the support vector set status already contains the correction information in the VC space. Ultimately, the system has successfully solve a common CBR problem which is defect detection with practical data set.

Setup	Retrieve	Reuse	Revise	Hyper parameter tuning		
ST1	Kernel comparison	null adaption	Neighbor sets	threshold	max cases	confidence function
ST2	Kernel comparison	null adaption	Full sets	threshold	max cases	confidence function
ST3	Kernel comparison	substitution adaption	Neighbor sets	threshold	max cases	confidence function
ST4	Kernel comparison	substitution adaption	Full sets	threshold	max cases	confidence function
ST5	SVM comparison	null adaption	Neighbor sets	threshold	max cases	confidence function
ST6	SVM comparison	null adaption	Full sets	threshold	max cases	confidence function
ST7	SVM comparison	substitution adaption	Neighbor sets	threshold	max cases	confidence function
ST8	SVM comparison	substitution adaption	Full sets	threshold	max cases	confidence function

Table 4.3 The result of hyper-parameters searching

Train size CBR: 0.1; Train size: 0.28; Test size: 0.72							
	Accuracy	RMSE	F1 score	Log-loss	Precision	Recall	ROC_AUC
CBR_full	0.9895	0.0104	0.9912	0.3624	0.9986	0.984	0.9909
SVM	0.9898	0.0101	0.9886	0.3499	0.9774	1	0.991
CBR_half	0.8711	0.1289	0.8837	4.4505	0.9709	0.8108	0.8869
Voting	0.9922	0.0078	0.9936	0.2682	0.9873	1	0.9902
RF	0.9931	0.0069	0.9942	0.2392	0.9996	0.9889	0.9942
Train size CBR: 0.1; Train size: 0.37; Test size: 0.63							
	Accuracy	RMSE	F1 score	Log-loss	Precision	Recall	ROC_AUC
CBR_full	0.9889	0.0111	0.9906	0.3839	0.9988	0.9825	0.9904
SVM	0.9941	0.0059	0.9934	0.2046	0.987	1.0	0.9946
CBR_half	0.8247	0.1753	0.8283	6.0554	0.9914	0.7112	0.8512
Voting	0.9948	0.0052	0.9956	0.1802	0.9913	1.0	0.9935
RF	0.995	0.005	0.9958	0.1723	0.9992	0.9924	0.9956
Train size CBR: 0.1; Train size: 0.46; Test size: 0.54							
	Accuracy	RMSE	F1 score	Log-loss	Precision	Recall	ROC_AUC
CBR_full	0.9879	0.0121	0.9894	0.4185	0.9977	0.9812	0.9891
SVM	0.9961	0.0039	0.9956	0.1356	0.9911	1.0	0.9965
CBR_half	0.885	0.115	0.8917	3.971	0.9745	0.8219	0.8964
Voting	0.9956	0..41	0.9964	0.1425	0.9929	1.0	0.9951
RF	0.9946	0.0054	0.9953	0.187	0.996	0.9946	0.9946
Train size CBR: 0.1; Train size: 0.55; Test size: 0.45							
	Accuracy	RMSE	F1 score	Log-loss	Precision	Recall	ROC_AUC
CBR_full	0.9896	0.0104	0.9905	0.36	0.9968	0.9842	0.9902
SVM	0.9949	0.0051	0.9942	0.1758	0.9885	1.0	0.9955
CBR_half	0.8978	0.1022	0.8986	3.5299	0.988	0.824	0.9059
Voting	0.9968	0.0031	0.9971	0.11	0.9942	1.0	0.9965
RF	0.9951	0.0049	0.9955	0.17	0.9989	0.9921	0.9954
Train size CBR: 0.1; Train size: 0.64; Test size: 0.36							
	Accuracy	RMSE	F1 score	Log-loss	Precision	Recall	ROC_AUC
CBR_full	0.9888	0.0112	0.9894	0.3869	0.9981	0.9809	0.9894
SVM	0.9972	0.0028	0.9968	0.0977	0.9936	1.0	0.9975
CBR_half	0.8798	0.1202	0.8795	4.1503	0.9472	0.8208	0.8842
Voting	0.9983	0.0017	0.9984	0.0586	0.9968	1.0	0.9982
RF	0.998	0.002	0.9981	0.0703	1.0	0.9962	0.9981
Train size CBR: 0.1; Train size: 0.73; Test size: 0.27							
	Accuracy	RMSE	F1 score	Log-loss	Precision	Recall	ROC_AUC
CBR_full	0.9889	0.0111	0.989	0.3846	0.9983	0.9798	0.989
SVM	0.9868	0.0132	0.9852	0.4559	0.9709	1.0	0.9882
CBR_half	0.8921	0.1079	0.886	3.7275	0.9589	0.8234	0.8934
Voting	0.9987	0.0013	0.9987	0.0444	0.9975	1.0	0.9987
RF	0.9966	0.0034	0.9966	0.1183	1.0	0.9933	0.9966
Train size CBR: 0.1; Train size: 0.82; Test size: 0.18							
	Accuracy	RMSE	F1 score	Log-loss	Precision	Recall	ROC_AUC
CBR_full	0.9899	0.0101	0.9894	0.3485	0.9974	0.9815	0.9896
SVM	0.9991	0.0009	0.9989	0.0326	0.9978	1.0	0.9992
CBR_half	0.9045	0.0955	0.8932	3.3002	0.9656	0.8309	0.9017
Voting	0.9999	1e-5	0.9999	9e-6	0.9999	1.0	0.9999
RF	0.9976	0.0024	0.9975	0.082	0.9988	0.9963	0.9976

Table 4.4 The performance of full pipeline CBR system against common CBR template, original offline SVM, and ensemble methods

Chapter 5

Conclusion

In this chaper, we summarize our research and provide insights into the future work. The thesis has introduced a complete pipeline solution in non destructive testing by using multiple sensor inspection gauge. The proposed system in this research proposed combination sensors of LIDAR, ultrasonic, light sensors and a full description of integration. In terms of data processing and analysis, the thesis denoted a versatile CBR system inspired SVM engine. Experimental results from pipe defect data set has indicated the overwhelming of the propose SVM CBR against traditional CBR system. This SVM-CBR utilized learning information from the beginning phase and successfully inherit it in the later steps. The comparison result in chapter 6 has proved the prevail of SVM-CBR in all phases. Since the thesis has provided a end to end solution, it is mandatory to review it in chapter wise.

We firstly introduced the current problem of existing non destructive testing methods, the problem of designing an effective inspection gauge and the ability to utilize all available sensory information. The later part raise the problem of existing expert system, specialize in the incoherent between phases which lead to the mislead of important features. Subsequently, the thesis initiates the objectives of the study to establish an enhanced CBR system concentrating on the pipe defect detection problem from multiple sensors.

Chapter 2 familiarize readers with a comprehensive review in the related studies. The literature of this chapter has categorized into the hardware and software aspects. In the first section, concepts and criteria of the design of inspection gauge is focused. The section introduced the advantages and limitation of existing gauge model. The research also mentions on the used of each specific sensor in each gauge. The mechanism of movement and detection is familiarized in separated with the inspection model. In terms of software, the chapter allocates 2 sections to described the availability of expert system. The chapter is not narrow the perception in only pipe defect detection problem but aiming on the general of existing CBR system. A section is used to summarize the research on SVM which is the core of the proposed system while the other section abstract the branch of expert system. Literature of SVM focuses on the approach of solving the SVM objective function in offline and online mode.

Chapter 3 encapsulate the studies of previous chapter to accomplish the design of both inspection gauge and decision system. The first section is granted the review in chapter 2 to construct the frame of inspection gauge. In general, the pipeline detection gauge with multiple sensors is built on LEGO Mindstorm toolkit with the supplementary components. The data was captured regularly during the operation process and described the status of the inspected pipe. The proposed sensors were also well integrated as attaching to the robot. An custom Raspberry Pi OS is adopted to control the gauge operation. The next section inspired finding from the research of solving SVM to conclude the relevant engine core. In term of learning model, SVM attempts SRM approach which minimizes an upper bound on the expected risk, while ERM minimizes the error on the training data. As a result, SVM is favored as the core detection mechanism due to the effective performance and excellent generalization ability in high dimensions.

This chapter also carefully described the entire SVM-CBR system which reviewed in the last section of chapter 2. The chapter emphasize how SVM core is reusable in each phase according to the proposed design. The first section of the chapter demonstrate the final form of inspection gauge from the blueprint design. The later section includes technical system architecture of the CBR. The implementation is built to ensure the extension for different problem.

Chapter 4 portrays the experimental design to evaluate the success of this system. This chapter defined the configuration of running inspection gauge in lab scale environment. The configuration emphasized on the requirement of pipe shape and size, the type of defect and experimental running times. In the next section, the thesis composed a detail comparison process for each phase between SVM-CBR and traditional CBR. At the end, the chapter depicts the procedure to evaluate the whole system in competing with other advance system.

Lastly, the chapter provides an insight analysis on the result obtained from the experiment. Overall, the proposed system opens for further extension and improvement from the robot design to the detection approach. In future work, an advanced robot trajectory system must implement and integrate into the operation in a complex pipeline system As this work only examines the operation in a lab-scale environment, field test experiments, in different shapes, diameters and types of material, are required to verify the efficiency of the detection strategy. Other types of data that exploit different features of pipe status, especially images, should incorporate to expand the richness of features. Although conventional SVM produced high accuracy result, the choice of kernel remains manually. As shifting to a complex pipe system, the reliable performance has not been verified. Since the data contains meaningful geometry information, constructing a schema for using various kernels are experimented to increase the performance.

This work was undertaken to design a consistent CBR system which is fully integrated with SVM. The experimental result consolidates the effectiveness of extracting the learning factors from machine learning model and applying into the relevant phase. In addition, the investigation also suggests a consideration to operate CBR system with online learning mechanism. However, the current study has only examined in 1 dataset. In order to be extended as a standard procedure, more experiments should be conducted with various dataset. Besides manipulates the core learning function as in the proposed system, a combination between our proposed pipeline and an outer shell decision support mechanism like Bayesian case model as in [28] to achieve a durable consolidation should be taken into account .

References

[1] M.S. Safizadeh and T. Azizzadeh. Corrosion detection of internal pipeline using ndt optical inspection system. *NDT & E International*, 52:144–148, 2012.

[2] Sunil K. Sinha and Fakhri Karray. Classification of underground pipe scanned images using feature extraction and neuro-fuzzy algorithm. *IEEE Transactions on Neural Networks*, 13(2):393–401, 2002.

[3] Timur Chis and Andrei Saguna. Pipeline Leak Detection Techniques. *Annals. Computer Science Series, 5th Tome 1st Fasc.*, pages 25–34, 2007.

[4] Sunil K Sinha and Paul W Fieguth. Automated detection of cracks in buried concrete pipe images. 15:58–72, 2006.

[5] W. Al-Rafai and R. J. Barnes. Underlying the performance of real-time software-based pipeline leak-detection systems. *Pipes & Pipelines International*, 44(December):44–51, 1999.

[6] A V Deokar and V D Wakchaure. Experimental Investigation of Crack Detection in Cantilever Beam Using Natural Frequency as Basic Criterion. *International Conference on Current Trends in Technology (NUiCONE)*, pages 8–10, 2011.

[7] Mohd Zamzuri Ab Rashid, Mohd Fitri Mohd Yakub, Sheikh Ahmad Zaki bin Shaikh Salim, Normaisharah Mamat, Sharifah Munawwarah Syed Mohd Putra, and Shairatul Akma Roslan. Modeling of the in-pipe inspection robot: A comprehensive review. *Ocean Engineering*, 203:107206, 2020.

[8] S. Mirzoev, S. Mashurov, and J. Sibila. A Comprehensive Approach to Integrity of Non-Piggable Pipeline Based on Combined DCVG/CIPS/MTM Survey. *Proceedings of the Pipeline Technology Conference, Berlin, Germany*, 2015.

[9] Lee Vuen Nee, I. Elamvazuthi, Timothy Ganesan, M.K.A. Ahamed Khan, and S. Parasuraman. Development of a laboratory-scale pipeline inspection robot. *Procedia Computer Science*, 76:9–14, 2015. 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IEEE IRIS2015).

[10] Ahmad Bala Alhassan, Xiaodong Zhang, Haiming Shen, and Haibo Xu. Power transmission line inspection robots: A review, trends and challenges for future research. *International Journal of Electrical Power & Energy Systems*, 118:105862, 2020.

[11] Zheng Liu and Yehuda Kleiner. State of the art review of inspection technologies for condition assessment of water pipes. *Measurement*, 46(1):1–15, 2013.

[12] Phat Huynh, Robert Ross, Andrew Martchenko, and John Devlin. 3d anomaly inspection system for sewer pipes using stereo vision and novel image processing. pages 988–993, 2016.

[13] P.A. Torrione, C.S. Throckmorton, and L.M. Collins. Performance of an adaptive feature-based processor for a wideband ground penetrating radar system. *IEEE Transactions on Aerospace and Electronic Systems*, 42(2):644–658, 2006.

[14] A. Srivani and M. Anthony Xavior. Investigation of surface texture using image processing techniques. *Procedia Engineering*, 97:1943–1947, 2014. "12th Global Congress on Manufacturing and Management" GCMM - 2014.

[15] S.B. Costello, D.N. Chapman, C.D.F. Rogers, and N. Metje. Underground asset location and condition assessment technologies. *Tunnelling and Underground Space Technology*, 22(5):524–542, 2007. Trenchless Technology.

[16] Qiuping Ma, Guiyun Tian, Yanli Zeng, Rui Li, Huadong Song, Zhen Wang, Bin Gao, and Kun Zeng. Pipeline in-line inspection method, instrumentation and data management. *Sensors*, 21(11), 2021.

[17] Heidar Hashemi, Awaluddin Mohamed Shaharoun, and Izman Sudin. A case-based reasoning approach for design of machining fixture. *International Journal of Advanced Manufacturing Technology*, 74(1-4):113–124, 2014.

[18] Dongxiao Gu, Changyong Liang, and Huimin Zhao. A case-based reasoning system based on weighted heterogeneous value distance metric for breast cancer diagnosis. *Artificial Intelligence in Medicine*, 77:31–47, 2017.

[19] Mohammad Reza Khosravani, Sara Nasiri, and Kerstin Weinberg. Application of case-based reasoning in a fault detection system on production of drippers. *Applied Soft Computing Journal*, 75:227–232, 2019.

[20] Cindy Marling, Stefania Montani, Isabelle Bichindaritz, and Peter Funk. Synergistic case-based reasoning in medical domains. *Expert Systems with Applications*, 41(2):249–259, 2014.

[21] Shahina Begum, Shaibal Barua, and Mobyen Uddin Ahmed. Physiological sensor signals classification for healthcare using sensor data fusion and case-based reasoning. *Sensors (Switzerland)*, 14(7):11770–11785, 2014.

[22] Hui Li and Jie Sun. Predicting business failure using multiple case-based reasoning combined with support vector machine. *Expert Systems with Applications*, 36(6):10085–10096, 2009.

[23] Colin Fyfe and Juan Corchado. A comparison of Kernel methods for instantiating case based reasoning systems. *Advanced Engineering Informatics*, 16(3):165–178, 2002.

[24] Dina A. Sharaf-El-Deen, Ibrahim F. Moawad, and M. E. Khalifa. A new hybrid case-based reasoning approach for medical diagnosis systems. *Journal of Medical Systems*, 38(2), 2014.

[25] Pau Herrero, Peter Pesl, Monika Reddy, Nick Oliver, Pantelis Georgiou, and Christofer Toumazou. Advanced insulin bolus advisor based on run-to-run control and case-based reasoning. *IEEE Journal of Biomedical and Health Informatics*, 19(3):1087–1096, 2015.

[26] Mazin Abed Mohammed, Mohd Khanapi Abd Ghani, N. Arunkumar, Omar Ibrahim Obaid, Salama A. Mostafa, Mustafa Musa Jaber, M. A. Burhanuddin, Bilal Mohammed Matar, Saif khalid Abdullatif, and Dheyaa Ahmed Ibrahim. Genetic case-based reasoning for improved mobile phone faults diagnosis. *Computers and Electrical Engineering*, 71(July):212–222, 2018.

[27] Mirjam Minor, Ralph Bergmann, and Sebastian Görg. Case-based adaptation of workflows. *Information Systems*, 40:142–152, 2014.

[28] Been Kim, Cynthia Rudin, and Julie Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 1952–1960, Cambridge, MA, USA, 2014. MIT Press.

[29] Juan F. de Paz, Javier Bajo, Angélica González, Sara Rodríguez, and Juan M. Corchado. Combining case-based reasoning systems and support vector regression to evaluate the atmosphere-ocean interaction. *Knowledge and Information Systems*, 30(1):155–177, 2012.

[30] Chin Yuan Fan, Pei Chann Chang, Jyun Jie Lin, and J. C. Hsieh. A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification. *Applied Soft Computing Journal*, 11(1):632–644, 2011.

[31] Pei Chann Chang, Chin Yuan Fan, and Wei Yuan Dzan. A CBR-based fuzzy decision tree approach for database classification. *Expert Systems with Applications*, 37(1):214–225, 2010.

[32] Claudio A. Policastro, André C.P.L.F. Carvalho, and Alexandre C.B. Delbem. A hybrid case adaptation approach for case-based reasoning. *Applied Intelligence*, 28(2):101–119, 2008.

[33] Xin Li, Wuyi Yu, Xiao Lin, and S. S. Iyengar. On optimizing autonomous pipeline inspection. *IEEE Transactions on Robotics*, 28(1):223–233, 2012.

[34] R. Montero, J. G. Victores, S. Martinez, A. Jarden, and C. Balaguer. Past, present and future of robotic tunnel inspection. *Automation in Construction*, 59:99–112, 2015.

[35] Amit Shukla and Hamad Karki. A review of robotics in onshore oil-gas industry. *2013 IEEE International Conference on Mechatronics and Automation, IEEE ICMA 2013*, pages 1153–1160, 2013.

[36] Se Gon Roh and Hyouk Ryeol Choi. Differential-drive in-pipe robot for moving inside urban gas pipelines. *IEEE Transactions on Robotics*, 21(1):1–17, 2005.

[37] Jun Okamoto, Julio C Adamowski, Marcos S.G Tsuzuki, Flávio Buiochi, and Claudio S Camerini. Autonomous system for oil pipelines inspection. *Mechatronics*, 9(7):731–743, 1999.

[38] S. Hirose, H. Ohno, T. Mitsui, and K. Suyama. Design of in-pipe inspection vehicles for ϕ25, ϕ50, ϕ150 pipes. *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 3(May):2309–2314, 1999.

[39] Koichi Suzumori, Kohei Hori, and Toyomi Miyagawa. A direct-drive pneumatic stepping motor for robots: designs for pipe-inspection microrobots and for human-care robots. *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 4(May):3047–3052, 1998.

[40] M. Muramatsu, N. Namiki, R. Koyama, and Y. Suga. Autonomous mobile robot in pipe for piping operations. *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, 3:2166–2171, 2000.

[41] H. R. Choi and S. M. Ryew. Robotic system with active steering capability for internal inspection of urban gas pipelines. *Mechatronics*, 12(5):713–736, 2002.

[42] Koichi Suzumori, Toyomi Miyagawa, Masanobu Kimura, and Yukihisa Hasegawa. Micro inspection robot for 1-in pipes. *IEEE/ASME Transactions on Mechatronics*, 4(3):286–292, 1999.

[43] Dongwoo Lee, Jungwan Park, Dongjun Hyun, Gyunghwan Yook, and Hyun Seok Yang. Novel mechanisms and simple locomotion strategies for an in-pipe robot that can inspect various pipe types. *Mechanism and Machine Theory*, 56:52–68, 2012.

[44] Zhelong Wang and Hong Gu. A bristle-based pipeline robot for Ill-constraint pipes. *IEEE/ASME Transactions on Mechatronics*, 13(3):383–392, 2008.

[45] L. Pfotzer, S. Klemm, A. Roennau, J. M. Z?llner, and R. Dillmann. Autonomous navigation for reconfigurable snake-like robots in challenging, unknown environments. *Robotics and Autonomous Systems*, 89:123–135, 2017.

[46] I. Hayashi, N. Iwatsuki, and S. Iwashina. The running characteristics of a screw-principle microrobot in a\nsmall bent pipe. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 152(1):225–228, 1995.

[47] Zheng Hu and Ernest Appleton. Dynamic characteristics of a novel self-drive pipeline pig. *IEEE Transactions on Robotics*, 21(5):781–789, 2005.

[48] a.M. Bertetto and M. Ruggiu. In-pipe inch-worm pneumatic flexible robot. *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No.01TH8556)*, 2(July):1226–1231, 2001.

[49] Han Pang Huang, Jiu Lou Yan, and Teng Hu Cheng. Development and fuzzy control of a pipe inspection robot. *IEEE Transactions on Industrial Electronics*, 57(3):1088–1095, 2010.

[50] Mahmoud R. Halfawy and Jantira Hengmeechai. Automated defect detection in sewer closed circuit television images using histograms of oriented gradients and support vector machine. *Automation in Construction*, 38:1–13, 2014.

[51] Joshua Myrans, Zoran Kapelan, and Richard Everson. Automated Detection of Faults in Wastewater Pipes from CCTV Footage by Using Random Forests. *Procedia Engineering*, 154:36–41, 2016.

[52] Srinath S. Kumar, Dulcy M. Abraham, Mohammad R. Jahanshahi, Tom Iseley, and Justin Starr. Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks. *Automation in Construction*, 91(October 2017):273–283, 2018.

[53] Jack C.P. Cheng and Mingzhu Wang. Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. *Automation in Construction*, 95(June):155–171, 2018.

[54] S. Dierks and A. Kroll. Quantification of methane gas leakages using remote sensing and sensor data fusion. *2017 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, March 2017.

[55] Min Meng, Yiting Jacqueline Chua, Erwin Wouterson, and Chin Peng Kelvin Ong. Ultrasonic signal classification and imaging system for composite materials via deep convolutional neural networks. *Neurocomputing*, 257:128–135, 2017.

[56] Spencer Gibb, Hung Manh La, Tuan Le, Luan Nguyen, Ryan Schmid, and Hu Pham. Nondestructive evaluation sensor fusion with autonomous robotic system for civil infrastructure inspection. *Journal of Field Robotics*, 35(September):988–1004, 2018.

[57] A. Novikoff. On convergence proofs for perceptrons. 1963.

[58] V.N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

[59] John Mashford, Mike Rahilly, Paul Davis, and Stewart Burn. A morphological approach to pipe image interpretation based on segmentation by support vector machine. *Automation in Construction*, 19(7):875–883, 2010.

[60] John Mashford, Dhammika De Silva, Stewart Burn, and Donavan Marney. Leak detection in simulated water pipe networks using SVM. *Applied Artificial Intelligence*, 26(5):429–444, 2012.

[61] Jian Ji, Chunshun Zhang, Jayantha Kodikara, and Sheng Qi Yang. Prediction of stress concentration factor of corrosion pits on buried pipes by least squares support vector machine. *Engineering Failure Analysis*, 55:131–138, 2015.

[62] Lam Hong Lee, Rajprasad Rajkumar, Lai Hung Lo, Chin Heng Wan, and Dino Isa. Oil and gas pipeline failure prediction system using long range ultrasonic transducers and Euclidean-Support Vector Machines classification approach. *Expert Systems with Applications*, 40(6):1925–1934, 2013.

[63] Nik Ahmad Akram, Dino Isa, Rajprasad Rajkumar, and Lam Hong Lee. Active incremental Support Vector Machine for oil and gas pipeline defects prediction system using long range ultrasonic transducers. *Ultrasonics*, 54(6):1534–1544, 2014.

[64] Rafael Amaya-Gómez, Mauricio Sánchez-Silva, and Felipe Muñoz. Pattern recognition techniques implementation on data from In-Line Inspection (ILI). *Journal of Loss Prevention in the Process Industries*, 44:735–747, 2016.

[65] John Shawe-Taylor and Shiliang Sun. A review of optimization methodologies in support vector machines. *Neurocomputing*, 74(17):3609–3618, 2011.

[66] John Shawe-Taylor and Shiliang Sun. *Kernel Methods and Support Vector Machines*. Number July 2012. 2014.

[67] Janet L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 1992.

[68] Roger C. Schank. *Memory-Based Expert Systems*. Yale University, 1989.

[69] Ian Watson Marir and Farhi. Case-based reasoning: A review. *The Knowledge Engineering Review*, 9(4):327–354, 1994.

[70] J. Kolodner. Making the implicit explicit: Clarifying the principles of case-based reasoning. In *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, pages 349–370. AAAI Press, USA, 1996.

[71] Agnar Aamodt and Enric Plaza. Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Commun.*, 7(1):39–59, mar 1994.

[72] Pin Chan Lee, Tzu Ping Lo, Ming Yang Tian, and Danbing Long. An Efficient Design Support System based on Automatic Rule Checking and Case-based Reasoning. *KSCE Journal of Civil Engineering*, pages 1–11, 2019.

[73] Kuo Sui Lin. A case-based reasoning system for interior design using a new cosine similarity retrieval algorithm. *Journal of Information and Telecommunication*, 4(1):91–104, 2020.

[74] Saroj Biswas, Debashree Devi, and Manomita Chakraborty. A hybrid case based reasoning model for classification in internet of things (IoT) environment. *Journal of Organizational and End User Computing*, 30(4):104–122, 2018.

[75] Mohammed Ahmed Jubair, Salama A. Mostafa, Aida Mustapha, and Hanayanti Hafit. A Survey of Multi-Agent Systems and Case-Based Reasoning Integration. *International Symposium on Agents, Multi-Agent Systems and Robotics 2018, ISAMSR 2018*, pages 1–6, 2018.

[76] Mengqi Chen, Rong Qu, and Weiguo Fang. Case-based reasoning system for fault diagnosis of aero-engines. *Expert Systems With Applications*, 202(April):117350, 2022.

[77] El Ghouch Nihad, Kouissi Mohamed, and En Naimi El Mokhtar. Designing and modeling of a multi-agent adaptive learning system (MAALS) using incremental hybrid case-based reasoning (IHCBR). *International Journal of Electrical and Computer Engineering*, 10(2):1980–1992, 2020.

[78] H. C. Chang, L. Dong, F. X. Liu, and W. F. Lu. Indexing and retrieval in machining process planning using case-based reasoning. *Artificial Intelligence in Engineering*, 14(1):1–13, 2000.

[79] Maximiliano Miranda, Antonio A. Sánchez-Ruiz, and Federico Peinado. Towards human-like bots using online interactive case-based reasoning. In Kerstin Bach and Cindy Marling, editors, *Case-Based Reasoning Research and Development*, pages 314–328, Cham, 2019. Springer International Publishing.

[80] Habib Hadj-Mabrouk. Application of case-based reasoning to the safety assessment of critical software used in rail transport. *Safety Science*, 131:104928, 2020.

[81] Mazin Abed Mohammed, Mohd Khanapi Abd Ghani, N. Arunkumar, Omar Ibrahim Obaid, Salama A. Mostafa, Mustafa Musa Jaber, M.A. Burhanuddin, Bilal Mohammed Matar, Saif khalid abdullatif, and Dheyaa Ahmed Ibrahim. Genetic case-based reasoning for improved mobile phone faults diagnosis. *Computers and Electrical Engineering*, 71:212–222, 2018.

[82] Kuo-Sui Lin. A case-based reasoning system for interior design using a new cosine similarity retrieval algorithm. *Journal of Information and Telecommunication*, 4(1):91–104, 2020.

[83] David Burca, Manuel Sch, and Johannes Zlabinger. Case-Based Reasoning and Machine Learning. (May):0–22, 2018.

[84] Rahman Ali, Aasad Masood Khatak, Francis Chow, and Sungyoung Lee. A case-based meta-learning and reasoning framework for classifiers selection. *ACM International Conference Proceeding Series*, pages 1–6, 2018.

[85] Yuan Guo, Bing Zhang, Y. Sun, K. Jiang, and K. Wu. Machine learning based feature selection and knowledge reasoning for CBR system under big data. *Pattern Recognition*, 112, 2021.

[86] Malik Jahan Khan, Hussain Hayat, and Irfan Awan. Hybrid case-base maintenance approach for modeling large scale case-based reasoning systems. *Human-centric Computing and Information Sciences*, 9(1), 2019.

[87] Carlo Mehli, Knut Hinkelmann, and Stephan Jüngling. Decision support combining machine learning, knowledge representation and case-based reasoning. *CEUR Workshop Proceedings*, 2846:0–2, 2021.

[88] Petr Berka. Sentiment analysis using rule-based and case-based reasoning. *Journal of Intelligent Information Systems*, 55(1):51–66, 2020.

[89] Frode Sørmo, Jörg Cassens, and Agnar Aamodt. *Explanation in case-based reasoning-perspectives and goals*, volume 24. 2005.

[90] Domen Novak and Robert Riener. A survey of sensor fusion methods in wearable robotics. *Robotics and Autonomous Systems*, 73:155–170, 2015.

[91] Hagen Schempf, Edward Mutschler, Vitaly Goltsbergm, and William Crowley. Grislee: Gasmain repair and inspection system for live entry environments. *The International Journal of Robotics Research*, 22(7-8):603–616, 2003.

[92] Jong Hoon Kim, Gokarna Sharma, and S. Sitharama Iyengar. FAMPER: A fully autonomous mobile robot for pipeline exploration. *Proceedings of the IEEE International Conference on Industrial Technology*, pages 517–523, 2010.

[93] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[94] Yunqian Ma and Guodong Guo. Support vector machines applications. *Support Vector Machines Applications*, 9783319023:1–302, 2014.

[95] Heesung Lee and Euntai Kim. Genetic outlier detection for a robust support vector machine. *Int. J. Fuzzy Logic and Intelligent Systems*, 15:96–101, 2015.

[96] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, 2008.

[97] Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 1982.

[98] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. pages 1–21, 1998.

[99] Vinod Kumar Chauhan and Kalpana Dahiya. Problem formulations and solvers in linear svm: a review. *Artificial Intelligence Review*, 2019.

[100] Mario Martin. On-line support vector machines for function approximation. *Techn. report, Universitat Politècnica de Catalunya, Departament de Llengatges i Sistemes Informàtics*, pages 1–11, 2002.

[101] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: primal estimated sub-gradient solver for svm. *Mathematical Programming*, 127(1):3–30, Mar 2011.

[102] G. Cauwenberghs and T. Poggio. Incremental and Decremental Support Vector Machine Learning. *Advances in Neural Information Processing Systems*, 13:409–415, 2001.

[103] Junshui Ma and Simon Perkins. Accurate on-line support vector regression. *Neural computation*, 15(11):2683–2703, 2003.

[104] Pavel Laskov, C Gehl, S Krueger, and Klaus-Robert Müller. Incremental Support Vector Learning: Analysis, Implementation and Applications. *Journal of Machine Learning Research*, 7:1909–1936, 2006.

[105] Antoine Bordes and Jason Weston. Fast Kernel Classifiers with Online and Active Learning Seyda Ertekin Léon Bottou. *Journal of Machine Learning Research*, 6:1579–1619, 2005.

[106] Mohammad Reza Khosravani, Sara Nasiri, and Kerstin Weinberg. Application of case-based reasoning in a fault detection system on production of drippers. *Applied Soft Computing*, 75:227–232, 2019.

[107] Hui Zhao, Hua Chen, Wei Dong, Xinya Sun, and Yindong Ji. Fault diagnosis of rail turnout system based on case-based reasoning with compound distance methods. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 4205–4210, 2017.

[108] Buseung Cho, Kuinam J. Kim, and Jin-Wook Chung. Cbr-based network performance management with multi-agent approach. *Cluster Computing*, 20:757–767, 2017.

[109] D. Van-Khoa Le, Zhiyuan Chen, and Rajprasad Rajkumar. Multi-sensors in-line inspection robot for pipe flaws detection. *IET Science, Measurement & Technology*, 14:71–82, January 2020.