Machine learning based automated evaluation of piping design

Tan, Wei Chian

2019

Tan, W. C. (2019). Machine learning based automated evaluation of piping design. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/80906>

<https://doi.org/10.32657/10220/48128>

MACHINE LEARNING BASED AUTOMATED EVALUATION OF PIPING DESIGN

WEI CHIAN TAN

**SCHOOL OF MECHANICAL & AEROSPACE
ENGINEERING**

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

May 2019

.....

Date

.....

WEI CHIAN TAN

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

May 2019

.....

Date

.....

PROFESSOR I-MING CHEN

Authorship Attribution Statement

Please select one of the following; *delete as appropriate:

~~*(A) This thesis does not contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.~~

*(B) This thesis contains material from [3] paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 4 is published as W. C. Tan, I. Chen, S. J. Pan and H. K. Tan, "Analysis With Histogram of Connectivity: For Automated Evaluation of Piping Layout," in IEEE Transactions on Automation Science and Engineering (T-ASE), vol. 15, no. 1, pp. 381-392, Jan. 2018 and Wei Chian Tan, I-Ming Chen, Sinno Jialin Pan and Hoon Kiang Tan, "Automated design evaluation on layout of Piping and Instrumentation Diagram using Histogram of Connectivity," 2016 IEEE International Conference on Automation Science and Engineering (CASE), Fort Worth, TX, 2016, pp. 1295-1300.

The contributions of the co-authors are as follows:

- Under supervision from the three co-authors, I collected data and created the datasets, developed the methodology, implemented the idea and wrote the paper.
- Professor Chen I-Ming provided guidance from the perspective of automation on how the problem should be addressed systematically and pointed out relevant directions regarding this topic. He also reviewed the paper and pointed out the weaknesses.
- Assistant Professor Sinno Jialin Pan provided guidance and supervision from the perspective of machine learning, validated the approach, reviewed and edited the manuscripts.
- As an experienced engineer from the industry, Mr Hoon Kiang Tan provided guidance on piping designs of a ship and relevant rules.

Chapter 7 is published as W.C. Tan, I.M. Chen, D. Pantazis and S.J. Pan, "PipNet: Vision-based Analysis of Piping Design in Maritime Engineering," 2018 14th IEEE Conference on Automation Science and Engineering (CASE), Munich, 2018.

The contributions of the co-authors are as follows:

- Under supervision and guidance from the three co-authors, I collected data and created the dataset, developed the methodology, implemented the idea and wrote the paper.
- Professor Chen I-Ming provided guidance on the use of deep learning in automation, pointed out the difference compared to deep learning related research in the domain of computer vision and highlighted the need to go beyond a binary classification. He also reviewed and provided constructive comments to various versions of the paper.
- Dr. Dimitrios Pantazis was involved extensively in various discussions on the problem and an important improvement of the idea came up in a discussion with him.
- Assistant Professor Sinno Jialin Pan provided guidance as an expert on deep and transfer learning, validated the approach theoretically and practically, reviewed and edited various versions of the paper.

May 2019

.....
Date

.....
WEI CHIAN TAN

Acknowledgements

The author would like to thank his PhD advisor, Professor I-Ming Chen, and mentor, Associate Professor Sinno Jialin Pan, for the endless guidance, care and patience throughout the study. This project has been made possible with the support from the Nanyang Technological University and Lloyd's Register.

Next, a heartfelt gratitude also goes to Dr Dimitrios Pantazis from the Massachusetts Institute of Technology, who made an academic visit and a stay at the beautiful city of Cambridge in 2017 possible. It has given the author an amazing experience, both academically and personally.

A deep gratitude also goes to the colleagues in Robotics Research Centre, Nanyang Technological University and Lloyd's Register especially Mr Sobhith Hariharan, Dr Kie Hian Chua and Mr Tim Courtney for the industrial guidance on the project.

Last but not the least, the author would like to thank his family members for the continuous support spiritually throughout the study, including his mother (Yuet Ho), sister (Wei Ling), brother (Seng Chian), uncles (Kin Siong and Kin Keong) and auntie (Bee Yee).

Contents

Acknowledgements	xi
List of Figures	xvii
List of Tables	xxiii
Abstract	xxvii
1 Introduction	1
1.1 Challenges in Design Appraisal	1
1.2 Evaluation of a Piping Design	2
1.3 Problem Statement	3
1.4 Organisation	3
2 Literature Review	5
2.1 Design Automation in Engineering	5
2.1.1 Mechanical and Control	5
2.1.2 Civil and Environment	7
2.1.3 Electronics and Embedded Systems	8
2.1.3.1 Design Optimisation and Automation	9
2.1.3.2 Analysis, Simulation, Verification and Correction	10
2.1.3.3 Integrated Approach	10
2.1.4 Software	11
2.1.4.1 Automated Design	11
2.1.4.2 Analysis and Verification	12
2.1.5 Interdisciplinary	12
2.2 Relevant Developments in Machine Learning	14
2.3 Challenges and Future Directions	16
2.3.1 Knowledge Base and Expert System	16
2.3.2 Automated Modelling and Simulation	17
2.3.3 Possible Crossover among Domains	18
2.4 Summary	19
3 Piping Design	21

3.1	Introduction	21
3.2	Aspect or Dataset 0: MARPOL	21
3.3	Aspect or Dataset 1: Bilge System of Machinery Space	24
3.4	Aspect or Dataset 2: Fuel Oil Transfer System	27
3.5	Aspect or Dataset 3: Ballast system on Ship	31
3.6	Aspect or Dataset 4: Isolation of Bilge System	34
3.7	Summary	35
4	Histogram of Connectivity for Linear Classification	37
4.1	INTRODUCTION	37
4.2	Need of Interpretation Beyond Explicit Rules	38
4.3	Histogram of Connectivity (HoC)	38
4.3.1	Requirements	39
4.3.1.1	Independence to numbering or labelling sequence	39
4.3.1.2	Independence to number of connections	39
4.3.2	Assumptions and Limitations	39
4.3.3	Attributes of Each Component	40
4.3.4	Connectivity among Components	40
4.3.5	Histogram of Connectivity (HoC)	41
4.4	Properties of HoC	45
4.4.1	Segmentation and Analysis	45
4.4.2	Abstraction of Component Type	45
4.5	Learning from Examples	47
4.5.1	Linear Support Vector Machine (SVM)	48
4.6	Search for Non-Compliant Connections	48
4.7	Experimental Setup and Results	51
4.7.1	Dataset 0: Regulation 12, MARPOL	51
4.7.2	Dataset 1: Bilge System of the Machinery Space	52
4.7.3	Dataset 2: Fuel Oil Transfer System	53
4.7.4	Dataset 3: Ballast System	54
4.7.5	Dataset 4: Isolation of Bilge System	54
4.7.6	NCC	55
4.7.7	Less Data	56
4.8	Visualisation of Model	57
4.9	The performance on other classifiers	59
4.10	Discussion and Further Work	61
4.10.1	Directions to Investigate in the Future	63
4.11	Summary	63
5	Pyramid of Connectivity Histogram	65
5.1	Introduction	65
5.2	Pyramid of Connectivity Histogram	66
5.3	Learning by Linear Support Vector Machines	75

5.4	Experimental Setup and Results	76
5.4.1	Setup	76
5.4.2	Results	77
5.5	Visualisation of the Learnt Models	79
5.5.1	Dataset 0: Regulation 12, Annex I, MARPOL	79
5.5.2	Dataset 1: Bilge system of machinery space	80
5.5.3	Dataset 2: Fuel oil transfer system	81
5.5.4	Dataset 3: Ballast system	82
5.5.5	Dataset 4: Isolation of bilge system	84
5.6	Integrating Multi-Resolution into HoC	84
5.7	Discussion	87
5.7.1	Further Works	89
5.8	Summary	89
6	Integrated Learning	91
6.1	Introduction	91
6.2	HoC in Common Space	92
6.3	Integrated Learning	95
6.4	Learning with Limited Examples	96
6.5	Experimental Setup and Results	97
6.5.1	All Datasets	97
6.5.2	Dataset 0 and 3: MARPOL and Ballast System	99
6.5.3	Dataset 0 and 4: MARPOL and Isolation of Bilge	100
6.5.4	Dataset 3 and 4: Ballast System	101
6.5.5	Dataset 0, 3 and 4: Isolation of Bilge and Ballast System	103
6.6	Discussion and Further Work	105
6.6.1	Directions to Investigate in the Future	106
6.7	Summary	106
7	Transfer Learning	107
7.1	Introduction	107
7.2	Visual Understanding of Design	108
7.3	Methodology	108
7.3.1	Piping Net (PipNet)	108
7.3.2	Transfer Learning and Fine-Tuning	109
7.3.3	Data: Insufficiency and Imbalance	109
7.3.4	Flexibility in Representation	110
7.4	Experiments	111
7.4.1	Abbreviation	112
7.4.2	Symbol	121
7.5	Discussion and Future Work	130
7.6	Summary	131
8	Discussion and Conclusion	133

8.1	Summary of contributions	133
8.2	Discussion	135
8.2.1	Capturing existing knowledge via HoC, PoC and HoCC . . .	135
8.2.2	Learning by self-exploration: Transferring common visual features	136
8.2.3	Industrial adoption	136
8.2.4	Automated generation of compliant designs for industrial use	138
A	Rules Applicable from the Original Text	141
B	Legend for Symbols in Piping Designs	145
	List of Author's Awards, Patents, and Publications	147
	Bibliography	149

List of Figures

1.1	An example of a simple piping design.	2
2.1	Interdisciplinary Overview (Red: The major engineering domains, Black: The sub-domains or applications, Blue: The similarity. . . .	14
3.1	An example of a compliant design under aspect 0.	22
3.2	The first example of a non-compliant design under aspect 0.	22
3.3	The second example of a non-compliant design under aspect 0.	22
3.4	An example of a compliant design for MARPOL case.	23
3.5	An example of a compliant design of aspect 1.	25
3.6	The first example of a non-compliant design of aspect 1.	25
3.7	The second example of a non-compliant design of aspect 1.	26
3.8	An example of a compliant design of aspect 1 in the form of abbrevi- ations.	27
3.9	An example of a compliant design.	28
3.10	An example of a non-compliant design.	29
3.11	An example of a compliant design.	30
3.12	An example of a non-compliant design.	30
3.13	An example of a compliant design of aspect 3.	32
3.14	An example of a non-compliant design of aspect 3.	32
3.15	Aspect 3: An example of a compliant design in the form of abbrevi- ations.	33
3.16	An example of a compliant design of aspect 4.	34
3.17	An example of a non-acceptable design of aspect 4.	34
4.1	Establishing the matrix of connectivity and the HoC of a P&ID. . . .	41
4.2	A comparison of the HoC of a compliant design (blue) and a non- compliant design (red) under dataset 0 in R^{144} (x axis: Dimension, y axis: Value or frequency of the connection seen).	43
4.3	A comparison of the HoC of a compliant design (blue) and a non- compliant design (red) under dataset 1 in R^{49} (x axis: Dimension, y axis: Value or frequency of the connection seen).	44
4.4	An illustration of abstraction.	47
4.5	An example of an NCC of dataset 0.	50
4.6	An example of an NCC of dataset 1.	50

4.7	The visualisation of a model in R^{144} for dataset 0. x -axis: Dimension of the learnt model, y -axis: Weight.	57
4.8	The visualisation of a model in R^{49} for dataset 1. x -axis: Dimension of the learnt model, y -axis: Weight.	58
5.1	Beyond immediate neighbours.	67
5.2	A compliant design of dataset 0.	68
5.3	A non-compliant design of dataset 0.	69
5.4	An illustration of Pyramid of Connectivity Histogram.	70
5.5	A comparison of PoCs for dataset 0, blue: compliant design in figure 3.4, red: non-compliant design in figure 3.3. x -axis: Dimension, y -axis: Value or frequency.	70
5.6	A comparison of PoCs for dataset 2, blue: compliant design in figure 3.11, red: non-compliant design in figure 3.12. x -axis: Dimension, y -axis: Value or frequency.	71
5.7	An example of a non-compliant design in dataset 0.	72
5.8	An HoC of a non-compliant design in dataset 0. x -axis: Dimension, y -axis: Value or frequency.	72
5.9	A PoC of a non-compliant design in dataset 0. x -axis: Dimension, y -axis: Value or frequency.	73
5.10	A difficult example of a non-compliant design in dataset 0.	73
5.11	A PoC of the difficult example of a non-compliant design in dataset 0.	74
5.12	An example of a non-compliant design in dataset 2.	74
5.13	A PoC of a non-compliant design in dataset 2. x -axis: Dimension, y -axis: Value or frequency.	75
5.14	The magnitude of the weights learnt for dataset 0 using HoCs.	79
5.15	The magnitude of the weights learnt for dataset 0 using PoCs. x -axis: Dimension of the learnt model, y -axis: Absolute weight.	80
5.16	The magnitude of the weights learnt for dataset 2 using HoC. x -axis: Dimension of the learnt model, y -axis: Absolute weight.	82
5.17	The magnitude of the weights learnt for dataset 2 using PoC. x -axis: Dimension of the learnt model, y -axis: Absolute weight.	82
5.18	The magnitude of the weights learnt for dataset 0 using PoC in R^{225}	85
5.19	The magnitude of the weights learnt for dataset 2 using PoC in R^{225}	86
5.20	An example of a PoC in R^{225} for dataset 0.	86
5.21	An example of a PoC in R^{225} for dataset 2.	86
6.1	A compliant design under aspect 0.	93
6.2	A non-compliant design under aspect 0.	93
6.3	A comparison of the HoCC for dataset 0. Blue: Compliant design, Red: Non-compliant design. x -axis: Dimension, y -axis: Value or frequency.	94
6.4	A comparison of the HoCC for dataset 3. Blue: Compliant design, Red: Non-compliant design. x -axis: Dimension, y -axis: Value or frequency.	95

6.5	An illustration of integrated learning.	96
7.1	An example of a non-compliant design in dataset 0.	110
7.2	An example of a non-compliant design in dataset 1.	111
7.3	Set 0: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	112
7.4	Set 0: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	113
7.5	Set 0: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	113
7.6	Set 0: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	113
7.7	Set 1: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	114
7.8	Set 1: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	114
7.9	Set 1: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	114
7.10	Set 1: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	115
7.11	Set 2: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	116
7.12	Set 2: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	116

7.13	Set 2: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	116
7.14	Set 2: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	117
7.15	Set 3: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	118
7.16	Set 3: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	118
7.17	Set 3: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	119
7.18	Set 3: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	119
7.19	Set 4: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	120
7.20	Set 4: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	120
7.21	Set 4: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	120
7.22	Set 4: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	121
7.23	Set 0: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	121

7.24	Set 0: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	122
7.25	Set 0: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	122
7.26	Set 0: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	122
7.27	Set 1: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	123
7.28	Set 1: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	123
7.29	Set 1: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	123
7.30	Set 1: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	124
7.31	Set 2: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	125
7.32	Set 2: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	125
7.33	Set 2: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	125
7.34	Set 2: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	126

7.35	Set 3: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	127
7.36	Set 3: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	127
7.37	Set 3: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	127
7.38	Set 3: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	128
7.39	Set 4: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	129
7.40	Set 4: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	129
7.41	Set 4: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	129
7.42	Set 4: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.	130

List of Tables

3.1	List of the components studied in R^{144} .	23
3.2	List of the components studied in R^{49} .	26
3.3	Aspect 2: Legend for the abbreviations in a P&ID.	31
3.4	List of the components studied in R^{196} .	33
4.1	The accuracies on dataset 0 in R^{144} ($\mu \pm \nu$).	52
4.2	The performance of the different types of descriptors on dataset 0 ($\mu \pm \nu$).	52
4.3	The accuracies on dataset 0 in R^{121} ($\mu \pm \nu$).	52
4.4	The list of the components studied in R^{121} .	52
4.5	The accuracies on dataset 1 in R^{49} ($\mu \pm \nu$).	53
4.6	The performance of the different types of descriptors on dataset 1 ($\mu \pm \nu$).	53
4.7	The accuracies on dataset 2 in R^{225} ($\mu \pm \nu$).	53
4.8	The performance of the different types of descriptors on dataset 2 ($\mu \pm \nu$).	53
4.9	The accuracies on dataset 3 in R^{196} ($\mu \pm \nu$).	54
4.10	The performance of the different types of descriptors on dataset 3 ($\mu \pm \nu$).	54
4.11	The accuracies on dataset 4 in R^{196} ($\mu \pm \nu$).	54
4.12	The performance of the different types of descriptors on dataset 4 ($\mu \pm \nu$).	55
4.13	The accuracies on dataset 0 in R^{144} ($\mu \pm \nu$).	55
4.14	The accuracies on dataset 1 in R^{49} ($\mu \pm \nu$).	55
4.15	The accuracies on dataset 2 in R^{225} ($\mu \pm \nu$).	55
4.16	The accuracies on dataset 3 in R^{196} ($\mu \pm \nu$).	55
4.17	The accuracies on dataset 4 in R^{196} ($\mu \pm \nu$).	56
4.18	The accuracies on dataset 0 in R^{144} ($\mu \pm \nu$).	56
4.19	The accuracies on dataset 1 in R^{49} ($\mu \pm \nu$).	56
4.20	The accuracies on dataset 2 in R^{225} ($\mu \pm \nu$).	56
4.21	The accuracies on dataset 3 in R^{196} ($\mu \pm \nu$).	56
4.22	The accuracies on dataset 4 in R^{196} ($\mu \pm \nu$).	56
4.23	Set 0: The dimensions in R^{144} with the highest weights.	59
4.24	Set 1: The dimensions in R^{49} with the highest weights.	59
4.25	Set 0: The dimensions in R^{121} with the highest weights.	60

4.26	Set 2: The dimensions in R^{225} with the highest weights.	60
4.27	Set 3: The dimensions in R^{196} with the highest weights.	60
4.28	Set 4: The dimensions in R^{196} with the highest weights.	61
4.29	The performance of random forest ($\mu \pm \nu$).	61
4.30	The performance of deep neural network ($\mu \pm \nu$).	61
5.1	The values of T	77
5.2	The prediction accuracies under dataset 0 ($\mu \pm \nu$).	77
5.3	Performance on dataset 1 ($\mu \pm \nu$).	78
5.4	The prediction accuracies under dataset 2 ($\mu \pm \nu$).	78
5.5	Performance on dataset 3 ($\mu \pm \nu$).	78
5.6	Performance on dataset 4 ($\mu \pm \nu$).	78
5.7	The effects of resolution on dataset 0 ($\mu \pm \nu$).	78
5.8	The effects of resolution on dataset 2 ($\mu \pm \nu$).	78
5.9	Top 10 dimensions with the highest absolute magnitude for dataset 0 using HoCs.	80
5.10	Top 10 dimensions with the highest absolute magnitude for dataset 0 using PoCs.	81
5.11	Top 10 dimensions with the highest absolute magnitude for dataset 1 using PoCs.	81
5.12	Top 10 dimensions with the highest absolute magnitude for dataset 2 using HoC.	83
5.13	Top 10 dimensions with the highest absolute magnitude for dataset 2 using PoC.	83
5.14	Top 10 dimensions with the highest absolute magnitude for dataset 3 using PoC.	83
5.15	Top 10 dimensions with the highest absolute magnitude for dataset 4 using PoC.	84
5.16	Performance on dataset 0 ($\mu \pm \nu$).	85
5.17	Performance on dataset 2 ($\mu \pm \nu$).	85
5.18	Top 10 dimensions with the highest absolute magnitude for dataset 0 using PoC in R^{225}	87
5.19	Top 10 dimensions with the highest absolute magnitude for dataset 2 using PoC in R^{225}	87
6.1	Total IL: The accuracies on dataset 0 in R^{1521} ($\mu \pm \nu$).	98
6.2	Total IL: The accuracies on dataset 1 in R^{1521} ($\mu \pm \nu$).	98
6.3	Total IL: The accuracies on dataset 2 in R^{1521} ($\mu \pm \nu$).	98
6.4	Total IL: The accuracies on dataset 3 in R^{1521} ($\mu \pm \nu$).	98
6.5	Total IL: The accuracies on dataset 4 in R^{1521} ($\mu \pm \nu$).	98
6.6	IL 0 and 3: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).	99
6.7	IL 0 and 3: The accuracies on dataset 3 in R^{576} ($\mu \pm \nu$).	99
6.8	IL 0 and 3 with 10% of the data: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).	99

6.9	IL 0 and 3 with 10% of the data: The accuracies on dataset 3 in $R^{576} (\mu \pm \nu)$.	99
6.10	IL 0 and 3 with 15% of the data: The accuracies on dataset 0 in $R^{576} (\mu \pm \nu)$.	100
6.11	IL 0 and 3 with 15% of the data: The accuracies on dataset 3 in $R^{576} (\mu \pm \nu)$.	100
6.12	IL 0 and 4: The accuracies on dataset 0 in $R^{576} (\mu \pm \nu)$.	100
6.13	IL 0 and 4: The accuracies on dataset 4 in $R^{576} (\mu \pm \nu)$.	100
6.14	IL 0 and 4 with 10% of the data: The accuracies on dataset 0 in $R^{576} (\mu \pm \nu)$.	101
6.15	IL 0 and 4 with 10% of the data: The accuracies on dataset 4 in $R^{576} (\mu \pm \nu)$.	101
6.16	IL 0 and 4 with 15% of the data: The accuracies on dataset 0 in $R^{576} (\mu \pm \nu)$.	101
6.17	IL 0 and 4 with 15% of the data: The accuracies on dataset 4 in $R^{576} (\mu \pm \nu)$.	101
6.18	IL 3 and 4: The accuracies on dataset 3 in $R^{576} (\mu \pm \nu)$.	102
6.19	IL 3 and 4: The accuracies on dataset 4 in $R^{576} (\mu \pm \nu)$.	102
6.20	IL 3 and 4 with 10% of the data: The accuracies on dataset 3 in $R^{576} (\mu \pm \nu)$.	102
6.21	IL 3 and 4 with 10% of the data: The accuracies on dataset 4 in $R^{576} (\mu \pm \nu)$.	102
6.22	IL 3 and 4 with 15% of the data: The accuracies on dataset 3 in $R^{576} (\mu \pm \nu)$.	103
6.23	IL 3 and 4 with 15% of the data: The accuracies on dataset 4 in $R^{576} (\mu \pm \nu)$.	103
6.24	IL 0, 3 and 4: The accuracies on dataset 0 in $R^{576} (\mu \pm \nu)$.	103
6.25	IL 0, 3 and 4: The accuracies on dataset 3 in $R^{576} (\mu \pm \nu)$.	103
6.26	IL 0, 3 and 4: The accuracies on dataset 4 in $R^{576} (\mu \pm \nu)$.	103
6.27	IL 0, 3 and 4 with 10% of the data: The accuracies on dataset 0 in $R^{576} (\mu \pm \nu)$.	104
6.28	IL 0, 3 and 4 with 10% of the data: The accuracies on dataset 3 in $R^{576} (\mu \pm \nu)$.	104
6.29	IL 0, 3 and 4 with 10% of the data: The accuracies on dataset 4 in $R^{576} (\mu \pm \nu)$.	104
6.30	IL 0, 3 and 4 with 15% of the data: The accuracies on dataset 0 in $R^{576} (\mu \pm \nu)$.	104
6.31	IL 0, 3 and 4 with 15% of the data: The accuracies on dataset 3 in $R^{576} (\mu \pm \nu)$.	105
6.32	IL 0, 3 and 4 with 15% of the data: The accuracies on dataset 4 in $R^{576} (\mu \pm \nu)$.	105
7.1	CNN architectures investigated for piping design appraisal.	109
B.1	List of symbols used in the piping designs.	145

Abstract

A methodology to automate the process of evaluating the layout of a piping design in the form of a Piping and Instrumentation Diagram (P&ID) according to a set of rules of the marine and offshore industry, mainly from the International Maritime Organisation and Lloyd’s Register, is proposed. The method starts with transforming a P&ID into a vector $\mathbf{x}$ in R^d . The transformation is done based on a concept introduced for piping known as Histogram of Connectivity (HoC). The proposed descriptor captures two essential properties of a P&ID: the attributes of each component and the connectivity among the components. Next, a linear Support Vector Machine (SVM) is used to learn a classifier from existing compliant and non-compliant designs. Subsequently, the linear classifier can be used to check if an unseen design complies with the standards. In addition, to enable a follow up on a non-compliant design including corrections or modifications, a method to analyse the reason of non-compliance prediction by the learnt SVM model is introduced.

Next, on top of the Histogram of Connectivity developed, the concept of multi-resolution is introduced to look at connectivity beyond components which are connected immediately for cues on non-compliance. This is done by extracting the connectivity information among the components connected via a series of intermediate components. Known as Pyramid of Connectivity Histogram (PoC), the feature vectors extracted are fed into a linear SVM for learning and prediction subsequently.

Following the HoC and PoC, a method to learn piping designs of different aspects of international rules in an integrated manner is introduced. Instead of learning designs of different aspects separately, learning using an SVM is done in a common feature space for all aspects. Based on the concept of HoC established previously, the method begins with mapping the mathematical representation of each piping layout into a common feature space for all different aspects or datasets. Next, learning by a linear SVM is performed.

Lastly, an end-to-end learning approach based on the latest Convolutional Neural Network (CNN) architectures and transfer learning to perform vision-based analysis of piping designs is introduced. Having a piping design in the form of an image, a framework known as Piping Net (PipNet) is introduced to understand the design and interpret if it complies with applicable engineering regulations. Designs and corresponding labels (compliant or non-compliant) are fed into an existing trained CNN in the form of images for transfer learning, with the subsequently obtained fine-tuned network called PipNet. The developed system has demonstrated outstanding performance on all five challenging datasets of piping designs introduced in this work.

Chapter 1

Introduction

Known as design appraisal in the marine and offshore industry, the evaluation of an engineering design has been one of the most important topics since hundreds of years ago. The designs of massive and expensive systems for offshore operations such as Floating Production Storage and Offloading vessels are required to be evaluated and approved by an authorised classification society before the construction. The evaluation is performed according to sets of rules developed by classification societies, regulatory bodies and engineering institutes worldwide based on knowledge and experience.

1.1 Challenges in Design Appraisal

However, this process relies on an engineer's experience heavily and could vary significantly from person to person. This brings a frequently seen but not well addressed problem of interpretation of the existing rules based on the actual situation and the constraints of each project. In addition, the recent problem of an aging workforce is leading to a situation of the current pool of experts in the industry retiring very soon. There is an outward flow of knowledge and experience of the industry and hence an urgent need to automate this process.

1.2 Evaluation of a Piping Design

Among the various aspects of designs on-board a ship, the design of piping system of a ship, also known as a piping design, is among the most fundamental and critical ones. It is necessary for the smooth operations like the transport of the raw oil extracted from the drilling operations and maintaining the balance of the ship through the water ballast system on-board. Sea water is taken in and out of the bottom part of the ship to achieve required balance during a cruise. Major incidents and accidents such as the ship losing balance and sinking can happen due to an inappropriate piping design. An effective and efficient evaluation of a piping design (usually given in the form of a Piping and Instrumentation Diagram (P&ID)) can lead to an optimised design, a smooth operation and a minimised risk of accidents happening. Figure 1.1 shows an example of a P&ID. While the symbols used are commonly used in the industry, a legend is provided at the Table B.1 at the appendix.

An automated system for design appraisal can be deployed as a tool for the early warning of non-compliances in a piping design. For example, it could be in the form of a plug-in to an engineering software to assist naval architects during the process of designing a piping network on-board a ship. By removing as many non-compliant parts as possible during the early stage of the design, the need of having the design to be circulated between the design engineers and the reviewing parties is minimised, hence increasing the efficiency.

FIGURE 1.1: An example of a simple piping design.

1.3 Problem Statement

Following the issues discussed in the previous sections, this project aims to address the issue described above by developing an automated framework for the evaluation of a piping design. Given a P&ID as the input, the developed framework is to tell if the layout of the design is compliant or non-compliant, according to a set of international rules or guidelines.

1.4 Organisation

The remaining of the report is organised as follows. A survey of the state-of-the-art in the relevant communities is presented in Chapter 2. Five different aspects of the piping design of a ship and offshore platform are presented in Chapter 3. Next, the main methodology for the automated evaluation of a piping design by a mathematical representation based on Histogram of Connectivity (HoC) and a classification mechanism using linear Support Vector Machine (SVM) is presented in Chapter 4. An extension of the main mathematical representation from HoC to capture the indirect connectivity through a Pyramid of Connectivity Histogram is presented in Chapter 5. For industrial adoption, Chapter 6 attempts to address the issue of learning different aspects of a piping design together, in a common space for HoC. Next, deep neural networks based transfer learning is used to augment the methodology for the automated evaluation of a piping design in Chapter 7. Lastly, discussions on future directions of the project and conclusion are presented in Chapter 8. A copy of the original text of each applicable rule used in this thesis is provided in the appendix.

Chapter 2

Literature Review

This chapter starts with a survey on the state-of-the-art in the automation of an engineering design in section 2.1. Next, a review of the relevant developments in machine learning and computer vision follows. Lastly, a discussion on the state-of-the-art, the existing challenges and the aim of the project is presented.

2.1 Design Automation in Engineering

A survey on the recent works (within five years from the beginning of the study) on design automation in the major disciplines of engineering is carried out in this section.

2.1.1 Mechanical and Control

Existing works in mechanical and control span across a wide range of engineering problems. Optimisation techniques have been used in applications like an on shore wind farm [1] and smart home environments [2].

In structural study and optimisation, a cuckoo search based formulation was used as a meta-heuristic approach [3]. The firefly algorithm for the case of mixed variable [4] and the great deluge algorithm for the optimisation in constrained mechanical design problems [5] were developed.

In manufacturing and related activities, a knowledge based system was developed

for complex forging die machining [6] and the planning of an assembly sequence [7]. A study on methods with intelligence for the layout design of flexible manufacturing systems has also been performed [8], highlighting the advantages and disadvantages of each approach. A hybrid artificial bee colony algorithm was introduced for a robust optimal design and manufacturing [9]. With standard parts which can be reused, the design of mold was automated [10].

In robotics, the automated design of various types of systems including soft [11], swarm [12] and control systems using bees algorithm [13] have been developed. A study on the state-of-the-art of selective pressure for evolutionary robotics [14] has also been performed. Relevant to this, a framework for computer aided conceptual design applied on mechatronics products [15] was developed.

In ship design, a knowledge based approach was developed for optimisations, mainly based on the existing expertise, rules and examples [16]. Case based and rule based reasoning were used in the approach. Similar approaches including document based for a basic ship design [17] and knowledge based engineering for the structural design in a ship [18] have been explored. An attempt to automate the simulation of a ship navigation [19] has also been made. In a study of operations of a construction fleet, approaches using knowledge based simulation and multi-modal process data mining was developed [20].

Besides, a method to automate the design of synchronous reluctance motors which gives the best output of torque was introduced [21]. In turbine, an automated framework was developed for the design of FDI system [22], the initial design and the performance study of a radial inflow turbine based on a brute force search algorithm [23].

In works that involve piping, attempts to automate have been carried out in the construction of models including a genetic algorithm based approach for a thermal model of active pipe embedded building envelopes [24]. On the other hand, the main task in pipe routing during a design stage is to search for the optimal route of piping network given specific requirements such as the existence of some obstacles and criteria defining optimal, the latest efforts include a projection and geodesic based algorithm [25].

Along the direction of automated modelling, a work on the 3D modelling of metal frame connections based on Light Detection and Ranging (LiDAR) data [26] was carried out. Dimension reduction and meshing of a stiffened thin wall structure were automated [27]. A modelling with a Fuzzy C-Means Clustering (FCM) based

neuro-fuzzy inference system and a genetic algorithm-polynomial neural network was used in the problem of thermal conductivity of alumina-water nanofluid [28]. By iterative calculations of thermal hydraulic, structural mechanics and minimised coupling to neutron transport codes, a module based computer code was written to automate the design of a reactor core [29]. A parametric genetic algorithm based method for large scale wind-solar coupling [30], multi-resource constraints and scatter search based optimisation on a sequence of disassembly [31] have been introduced.

Along the direction of control, a hybrid bacterial foraging algorithm was developed for the automated experimental control design of an electrical drive [32]. Methods have also been developed for problems including the bi-objective [33] and multi-objective [34] optimisation of permanent magnet (PM) machines by finite element analysis (FEA) and differential evolution. ACADO which is an open source platform for control and dynamic optimisation [35] has also been introduced.

2.1.2 Civil and Environment

In the construction industry, following optimisation based building designs [36], a rule based approach was applied on Building Information Modeling (BIM) for the automated evaluation on the safety aspect of a building design [37]. Similarly, a case study on the safety compliance checking automatically to protect users from a fall [38] was also carried out. As a necessary step to automated planning for the safety and the checking for constructability, an ontology based semantic modelling of the knowledge in construction safety [39] and a rule based approach [40] were developed respectively. Implementations of expert or knowledge based systems have been successful here as engineering knowledge and experience involved in this category of problems is relatively suitable to be represented by rules.

Relevant to the design of a building, a space database for an automated review [41] and a data mining based analysis of an energy efficient building design [42] were introduced. Extending the existing works on building design, simulation based optimisation methods were used for the analysis of building performance [43]. {ANFIS} was introduced to predict the needs of building energy during the early stage of a design [44]. In addition, carbon accounting and BIM were integrated for a sustainable design with intelligence [45].

In the planning of a water network, attempts in automating the design process

have been made, covering problems such as the sectorisation with genetic algorithms and minimum dissipated power paths [46] and the performance study of a hybrid evolutionary based algorithm [47]. The task of partitioning a water network was automated through an integration of hydraulic simulation, graph partitioning and a heuristic optimisation criterion [48]. A multi-objective algorithm was also developed for the optimal design of water networks with a large number of pipes and nodes [49]. In design automation, approaches to optimise a water distribution network have been developed. The direction of evolutionary algorithm has received significant attention in this problem, modifications were introduced and applied, including a hybrid multi-objective [50], a genetic algorithm based with domain knowledge [51] and a penalty-free multi-objective [52]. Similarly, various methods have been developed for the optimisation of water distribution systems including a combined non-linear programming (NLP)-differential for a looped system [53], a self-adaptive differential [54] and a coupled binary linear programming differential [55]. Extending the direction further, efforts on optimising pipe replacements and control valve installations to reduce the leakage [56] has been made. A modelling of pipe dimensioning based on heuristic [57] was also introduced.

In another similar domain, the design of a sewer network was optimised using an adaptive genetic algorithm [58]. Through the Steiner minimal tree and particle swarm optimisation, an approach was developed for a multi-terminal pipe routing [59].

A network design can be represented effectively and efficiently by a graph as the design basically consists of a set of nodes and connections among them. This type of design will also be discussed in other domains.

Related to environment, the effects of an expert's opinions in the modelling of the environment were studied and analysed [60]. An expert system was developed as a support in selecting a design [61] through a question and answer framework. A data mining based approach was developed for the formulation of geotechnical engineering systems [62] using genetic programming related approaches.

2.1.3 Electronics and Embedded Systems

Significant progress has been observed in Electronics Design Automation (EDA) [63] in recent years. In general, directions can be divided into the automated design and the analysis. Each direction will be discussed in detailed.

2.1.3.1 Design Optimisation and Automation

Tools are being developed to assist in an automated design and the corresponding optimisation of electronic systems. From net merging algorithms designed for channel routing [64], an automated tool for the sizing of a transistor and optimising designs was developed for Field-Programmable Gate Array (FPGAs) [65]. Along the direction of optimal sizing, particle swarm optimisation for an analog circuit [66] and an evolutionary programming based approach for a grid connected photovoltaic system [67] were developed. In the optimisation of the reliability, a multi-criteria based deployment of embedded systems was introduced [68]. In the problem of decomposing a cell based row structure layout, it was demonstrated that a polynomial time algorithm could lead to the optimal solution [69].

Besides, a semantic parsing and Verilog based attempt to extract the design information from specifications in natural language [70] has been made. Methods to automate a design have been developed for Complementary Metal-Oxide-Semiconductor (CMOS) analog integrated circuits by a hybrid shuffled frog leaping algorithm [71], IEEE P1687 [72], logic in memory blocks for a specific application [73], a high power and frequency Series Parallel Resonant Converter (LCC) for electrostatic precipitators [74] and hard real-time embedded streaming systems [75], input filters for direct AC/AC matrix converters using a genetic algorithm [76], geometric modelling algorithm based conformal cooling circuits for rapid tooling [77] and the analog IC design flow to layout from a circuit level [78]. A modelling of reconfigurable logic devices was established as the first step of design automation of all graphene IC [79]. The possibility of using EDA to address the issue of counterfeiting [80] has also been discussed.

Another major focus in electronic design automation is design synthesis. Given required specifications in high level, the system is expected to synthesise or generate a design that fulfills the requirements, automatically. The direction has seen successful developments including a method for a low power encoder using reversible circuit synthesis [81], OpenCL based synthesising of a platform architecture [82], reversible circuits [83], a high level approach for an FPGA [84] and a computer based synthesis [85]. An automated synthesis of a heterogeneous interconnect topology for an FPGA has also been established [86].

The development of hardware and software codesign including directions for investigations in future was studied [87]. Enhanced by a system on a programmable chip and a pipeline design method, a particle swarm optimisation based approach

was developed for the hardware software co-design of embedded systems [88].

In addition, methods have been developed for the exploration of a design space, including an approach for 3D-stacked DRAM [89] and a multi-objective based approach for multi-core systems [90].

The issue of manufacturability of IC under the situation of extreme scaling was discussed [91]. A component based design for future was studied [92]. Relevant to this, in the domain of network on chip design, existing works on application mapping strategies were studied [93].

2.1.3.2 Analysis, Simulation, Verification and Correction

A verification of a design is an important problem in the field, there are two major directions for verification, formal and simulation. Methods under the formal category rely on mathematical models and relevant tools for the analysis of a system, a guarantee on the performance of a system is obtained after the verification. Simulation based approaches use test patterns generated as the input and study the output of the system.

In formal verification, methods for hybrid system [94] have been developed and ad-hoc routing protocols for wireless sensor networks have been studied [95].

Along the direction of simulation based approaches, an approach for the rapid automated validation of an ultralow latency digital core [96] and polychrony based approach for the simulation at system level [97] has been developed. Along the direction of test generation, a swarm intelligence and bounded model checking based approach for functional testing [98], a coverage directed approach by machine learning [99], and an approach for test packet generation [100] were established.

A solver and circuit simulator was used to model and analyse the performance of Through-Strata-Vias technology in three dimensional integration [101]. Binary decision diagram based approaches for the symbolic analysis of analog IC [102] and combinatorial testing [103] have been studied.

2.1.3.3 Integrated Approach

Methods integrating both design and analysis are also being developed. Real time optical Orthogonal Frequency Division Multiplexing (OFDM) transceivers were optimised and analysed accordingly based on the synthesis for a 65nm standard

cell library [104]. An implementation flow for WCHB QDI asynchronous circuits was developed to make it compatible with common tools for EDA on synchronous circuits [105].

In relevant areas, the issue of separability for memetic computing [106], design time and run time mapping methodologies [107] were studied. The state-of-the-art iterative and core guided maxsat solving which is an important fundamental for many problems in the domain [108] has also been discussed. The focus on design automation in electronics so far has been to generate a design or layout according to the user's specifications, based on the developed mathematical models. The use of intelligence to interpret a design is relatively limited in these works.

2.1.4 Software

A tremendous amount of algorithms was developed for automation in various aspects of a software design, ranging from the automated generation of control software [109] to non-cryptographic hash function [110].

2.1.4.1 Automated Design

Following the recent development in design optimisation [111], methods for an automated design were developed for multidimensional schemas from object oriented databases [112], a decision tree with an evolution algorithm [113], neural networks with L systems and genetic algorithms [114], decision tree induction algorithms for flexible receptor docking data [115], decision tree induction algorithms [116], a lightweight method for the convergence in network protocols [117], a multi-objective ant colony optimisation algorithm [118], the generation of a GUI model for mobile applications [119], hybrid self configuring evolutionary algorithm based design of fuzzy classifier [120], partitioning in parallel database systems [121] and assertions from system level design based on data mining [122]. A methodology for the design of self-adaptive software with formal guarantees was also established [123].

In the exploration of a design space, flow based methods for taking run time resource management into consideration during a design [124] and a regression based GPU design [125] have been developed.

A declarative language was proposed for the specifications of visual contracts and the verification of transformations specified by a transformation language [126].

Relevant to design automation, the role of dark knowledge and graph grammar was discussed [127].

2.1.4.2 Analysis and Verification

Along the direction of analysis, attempts have been made for scenarios including the detection of inconsistencies and the follow up in design models [128], a rule based analysis [129], a reduction to Boolean Satisfiability problem based verification of software power analysis countermeasures [130], a plug-in based testing of a GUI driven software [131] and a GUI ripping based testing of an Android application [132]. A validation of the security of mobile applications in the application market based on the dynamic information flow and a systematic exploration of the execution path [133] was also introduced.

Besides, a parallel symbolic execution based testing [134], a SLAM based model checking [135] and a model based size estimation of embedded software components [136] were developed. Relevant to this, statistical model inference based techniques were used to design experiments for the testing of software [137].

A computing repair tree based method to resolve the inconsistencies in designs [138] was developed. Similarly, an example based approach was developed to detect and correct defects [139].

Similar to electronics, an automated software test case generation has been a focus for software verification [140]. The advantages and disadvantages of an automated testing of software [141] were discussed. Besides, a condition based algorithm to automate the refactoring to strategy design pattern was developed [142].

Along the direction of automating simulation, a system called SAFE was introduced to provide guidance to users of ns-3 network simulator [143].

2.1.5 Interdisciplinary

Along the direction of optimisation, the artificial bee colony algorithm was used for large scale design optimisation problems in engineering [144]. A hybrid genetic algorithm with a flexible allowance technique for a constrained optimisation [145] and an approach for the optimisation of the parameter design of a multi-response process based on Taguchi Method [146] have been developed.

Flexible and robust CAD models developed in [147] enable Multidisciplinary Design Optimisation [148] [149] based on the concept of high level CAD templates which were reusable. Relevant to this, a procedure based method to automate a mixed dimension modelling from two and three dimensional CAD models [150] was introduced.

After modelling, along the direction of the automation of a simulation, a model-driven architecture based approach was developed for SysML models [151]. A procedure based method for the analysis of the output from a simulation called AutoSimOA [152] was developed.

Borrowing from electronic design automation, an evolutionary algorithm was used for the automated design of the X-Band antenna for NASA's Space Technology 5 spacecraft in avionics [153]. When a certain part of provided specifications change, the possibility to focus on regenerating the test cases involved and not all has been made possible based on a Petri Net [154]. An online model for the fault detection on the cloud using an SVM (Support Vector Machine)-grid has been introduced [155].

The current state of knowledge based engineering [156], the automated capture and representation of knowledge [157] have been studied. Similarly, an FCBS (function-cell-behaviour-structure) model based representation of the functional knowledge in a conceptual design [158] was developed. For biologically inspired designs, a knowledge based system was introduced with four characteristics defined, including cognitive, collaborative, conceptual and creative [159]. A principle solution and knowledge based methodology was proposed for an interdisciplinary creative design [160].

Besides, an algorithm for the concurrent consideration of product design and assembly sequence planning was established [161]. A preference selection index based methodology for a multiple attribute decision making was developed to search for the optimal facility layout design from existing selections [162].

Based on Taguchi and a multivariate statistical method, a generic method was developed for a process design of multi-response [163].

In some relevant surveys, studies on biological intelligence applied in the optimisation of supply chain management systems [164], intelligent systems for product design and future [165], spatial computing in design [166] and model based testing approaches [167] have been performed. Structured representation and retrieval of design rationale were also studied [168].

FIGURE 2.1: Interdisciplinary Overview (Red: The major engineering domains, Black: The sub-domains or applications, Blue: The similarity).

Based on the randomisation of parameters, a method to explore in a design space was proposed [169]. An argumentation framework was also developed as the support for design debates in the form of issue based information system [170].

The relevance of each domain in design automation is summarised in Figure 2.1.

2.2 Relevant Developments in Machine Learning

Support Vector Machines (SVM) [171] [172] and subsequent developments including Core Vector Machines [173] and latent SVM [174] have been applied very successfully in a wide range of applications, with or without theoretical modifications, ranging from computer vision, natural language processing, financial analysis, to medical and engineering applications, as listed below. A linear SVM has an additional advantage of being relatively fast compared to other machine learning methods during the training. Using photovoltaic power data and meteorological data, a method to do predict weather based on a multi-class SVM has been developed [175].

1. Face detection and recognition [176].
2. Image classification [177].
3. Object detection [178].
4. Image retrieval [179].
5. Image annotation [180].
6. Natural language processing [181].
7. Financial analysis and forecasting [182].
8. Medical [183].
9. Engineering [184].

Deep Learning [185] [186] and the relevant Neural Network based methods [187] have received much attention lately. In natural language processing, Recurrent Neural Network (RNN) and the relevant methods were used for the parsing of natural scenes and language [188], translation [189], knowledge transfer [190] and general understanding [191] [192]. The recurrent structure of an RNN allows the adaptation to the temporal nature of the domain and hence is a more suitable choice relatively.

On the other hand, Convolutional Neural Networks (CNN) have achieved the state-of-the-art performance in visual understanding such as face recognition [193] [194], pedestrian detection [195], three-dimensional object recognition [196], scene understanding [197], image classification [198] [199], super-resolution [200] and general feature learning for recognition [201] [202]. The convolutional structure of a CNN makes the architecture very suitable for image based application, as demonstrated by the state-of-the-art performance reported worldwide.

Starting from the deep architecture introduced in VGGNet [203] and AlexNet for classification on ImageNet [204], improved models are rapidly developed, including deep residual networks [205] and densely connected one [206]. Algorithms to understand the internal representations of CNNs have also been developed [207]. Finally, transfer learning [208] [209] has been successfully applied to a broad number of tasks. Recent efforts include visual categorisation [210] and machine aided medical diagnosis [211] [212].

In addition, using a combination of deep and reinforcement learning [213], AlphaGo [214] has achieved the state-of-the-art performance which is beyond human in the traditional Chinese game Go. Using a linear SVM to replace the softmax classifier in the framework of deep learning has shown an improvement in the performance [215]. Along the direction of reinforcement learning, the idea of parallel based on a Markov Chain has been explored [216].

While deep learning based techniques demonstrate promising performances, it typically requires a huge amount of data for learning or reasoning, this may be infeasible in some domains like piping designs. In addition, the computational cost is very high even with the support from the recent advancements in Graphics Processing Unit (GPU) [217].

In terms of classification frameworks, a cascade structure of classifiers [218] may give a performance improvement to a linear SVM by transforming the problem from the search for a hyperplane in R^d space into a series of weak classifiers (or a set of hyperplanes in the same space, each hyperplane takes care of the separation of some points), leading to a higher chance of a successful separation.

Transfer Learning [208] that allows the training (learning) and the testing (prediction) to be done in different domains has attracted significant interests in the community recently. This may be useful to the project as it leads to the possibility of learning in the form of text and examples while predicting in the form of an example only. However, it is unclear on how this can be done currently.

2.3 Challenges and Future Directions

After a survey on the recent developments in the individual disciplines in previous sections, this section highlights the existing challenges and the directions to investigate further. The section ends with a discussion on a few theoretical themes that may be of interest to multiple engineering domains.

2.3.1 Knowledge Base and Expert System

While the knowledge base (KB) remains one of the most significant focuses of the community lately, there is still a lack of a standard and systematic representation

of the engineering knowledge or experience. The forms of knowledge and experience differ among a wide range of sub-domains in the field, some suitable to be represented as pre-determined rules while some require the intelligence to interpret based on the experience. This poses a challenge in identifying a standard way or approach to search for the best representation for each application to represent the knowledge in a KB or similar system. The ability of a KB in dealing with capturing, representing and reusing the engineering experience across multiple engineering domains remains unclear too. Relevant to this, most expert systems developed so far for engineering design have been built using a rule base and fuzzy system for design optimisation in various aspects of engineering. However, engineering knowledge and experience are often hard to be represented comprehensively in the form of explicit rules. This does not include the expert's experience which is accumulated over the years and is even harder to capture.

In particular, the majority of current approaches are still lack of an ability to do generalised learning from examples by themselves and hence requires system designers or operators to assist in transforming the raw knowledge into a representation that the system can understand. This poses a big limitation on the capabilities of these methods and brings up the urgent need to incorporate more intelligence in engineering design with significantly less human intervention.

Looking from the perspective of intelligence, earlier works on incorporating intelligence into design automation tend to focus on very specific step by step instructions specified in design procedures or software programs. However, this requires a significant amount of human intervention in transforming the existing knowledge or experience into the system. Next, the development on KB, expert or rule base systems which allows more flexibility in interpreting existing intelligence in an engineering domain comes but this is still not flexible enough for knowledge retention in most areas in engineering.

As discussed earlier, significant advancements have been observed in machine learning and big data recently such as visual recognition and AlphaGo. Borrowing the concept, automated reasoning from a huge amount of examples without human intervention may be an interesting direction to investigate.

2.3.2 Automated Modelling and Simulation

Another potential direction is the automation in modelling and simulation. This can be further divided into two. The first is the automation in a simulation process, the preparation of models is still mainly manual currently, this has become a limitation on the efficiency. Using intelligence to generate engineering models and parameters automatically for the subsequent simulation and analysis is definitely of interest. While a simulation based analysis produces promising results as discussed, the meaning of those simulation results still requires an interpretation by experts. The question to ask here is if there is a way to use intelligence to interpret these results automatically, possibly for design refinements later.

If an automated simulation with intelligence can be achieved, it becomes possible to have a machine to serve as a virtual expert for design in specific areas. The computer can learn and explore in the design space by itself with as many simulations as possible without stopping, requiring minimal assistance from human experts, hence being able to accumulate experience in a much faster way, compared to an engineer who can be affected by fatigue. With this massive knowledge about a model simulated in different ways, the database of simulation results and intelligence of the system becomes a valuable reference to design engineers. It also saves time in the process of design by doing simulations in advance and minimising the time design engineers need to spend waiting for results from simulation.

2.3.3 Possible Crossover among Domains

A graph based representation is very suitable for representation of circuit type of engineering designs, including but not limited to piping, water network or distribution system, electronic circuit and computer network. As discussed earlier, each of these topics has been studied extensively. It is observed that due to the similarity to a graph based representation, works done in each of the topics may benefit others. In particular, automated design including routing between two points and synthesis based on specific requirements is of significant interest as it reduces an engineer's workload.

Knowledge based engineering has been successful in design automation in mechanical engineering, it may benefit electronics by being able to incorporate known

principles and rules into an automated system for design and subsequent analysis. Besides, a combination of knowledge based and model based approach would be interesting to look at as it combines the strength of both, into a model which contains established knowledge and knowledge base from experience. However, for knowledge that is not suitable to be represented by rules in a KB, a further study is needed.

The optimisation and automation methods in automotive, ship and robotics for both static and dynamic analysis may be of interest to each other as they are based on fundamental mechanical analysis, including but not limited to structural, motion and fluid.

While design synthesis has been studied for a long time in electronic design automation, this could also be of interest to other engineering disciplines. As highlighted earlier, the existence or development of a standard language for the description of specific design requirements across most domains would be advantageous. There are two alternatives, domain specific or natural language. A domain specific language is expected to work in a subset of engineering domains. Under natural language based approaches, design descriptions given in a natural language need to be translated in a form that a machine can understand, before a further analysis. Besides electronics and software, being able to verify and validate a design before the construction or mass production would allow a minimisation of potential failure cost. Similar to design synthesis, significant achievements in electronics and software in design verification can be beneficial to many other domains. However, as the representation of a design is likely to be different across domains, modification to the theoretical foundations of the existing work before use is expected.

While evolutionary and genetic algorithms have been applied successfully in a range of engineering problems including the design of a water network, software and modelling of thermal properties, a similar approach may also benefit directions like product design and electronics, especially in the search for the optimal design.

2.4 Summary

As discussed in this chapter, after a broad survey across multiple domains, closely similar works on the automated evaluation of a piping layout are hardly seen. While methods have been proposed for the automation of graph based and piping

design, there is still a gap. The focuses of the community has been more on the (semi-) automated analysis of various engineering properties of a pipe design and the generation of pipe routing between two points. From the state-of-the-art, it remains unclear on how an interpretation if a P&ID complies with the specific parts of the international rules can be done. This requires intelligence in both the understanding of a P&ID and the international rules, which is to be addressed in this project.

Chapter 3

Piping Design

3.1 Introduction

Five aspects of the piping design on-board a ship or an offshore platform have been studied in this work. Five datasets of thousands of designs have been created for each aspect, as shown below.

3.2 Aspect or Dataset 0: MARPOL

The Regulation 12, Annex I, Regulations for the Prevention of Pollution by Oil of the International Convention for the Prevention of Pollution from Ships (MARPOL) [219] [220] [221] is used as the first aspect for the performance study.

In order to prevent pollution, the primary requirement in this section of the regulation is that the tank storing oil residue (or sludge) is not to be connected directly to the oily bilge water holding tank, the filtering system, the bilge system, the overboard and the other relevant systems. However, the use of a common standard discharge connection for both the oil residue tank and the oily bilge water holding tank is acceptable, with the condition that screw-down non-return valves are provided at suitable locations to prevent oil residue from entering the oily bilge water holding tank and the other systems mentioned above. Figure 3.1 shows a compliant design.

FIGURE 3.1: An example of a compliant design under aspect 0.

FIGURE 3.2: The first example of a non-compliant design under aspect 0.

FIGURE 3.3: The second example of a non-compliant design under aspect 0.

Two examples of a non-compliant design are shown in Figures 3.2 and 3.3. The reason of non-compliance for the first example is that the globe valves highlighted allow the possibility of oil residue entering the oily bilge water holding tank, the filter, the oily water separator and the bilge system. The reason for the second is

similar, the globe valve highlighted allows the possibility of oil residue to flow into the oily bilge water holding tank through the branch connection.

Piping designs in this work are assumed to be in vector form such as CAD (Computer-Aided Design) where information of components and connectivity can be extracted directly. For easier processing, symbols in the diagram are replaced using abbreviations in the design shown (such as SNV for screw-down non-return valve, ORT for oil residue tank, OBW for oily bilge water holding tank), the complete legend is provided in Table 3.1. Figure 3.4 shows an example of a compliant design in the form of abbreviations.

TABLE 3.1: List of the components studied in R^{144} .

P	Pump
SNV	Screw-Down Non-Return Valve
GV	Globe Valve
ORT	Oil Residue (Sludge) Tank
OBW	Oily Bilge Water Holding Tank
SDC	Standard Discharge Connection
FOB	Filter/Oily Water Separator/Bilge System
Inc	Incinerator
ORS	Oil Residue (Sludge) Service Tank
NV	Non-Return Valve
J	Joint
BV	Ball Valve

FIGURE 3.4: An example of a compliant design for MARPOL case.

3.3 Aspect or Dataset 1: Bilge System of Machinery Space

The bilge system of a ship takes care of most draining activities on a ship, from removing additional sea water in a ship to the discharge of dirty water after a proper treatment. Section 4, Chapter 13, Part 5 of the Rules and Regulations for the Classification of Ships of Lloyd's Register (LR) [222] provides the requirements for the bilge draining of the machinery space on a ship, covering the engine room. In particular, in [222], Section 4.1 highlights the need of two suctions for the machinery space where one can be through the main bilge line and the other through a direct suction (a pump with independent power). Besides, Section 4.6 highlights the need of an emergency bilge drainage for the main machinery space. If both suctions provided fail, an emergency suction is needed to ensure smooth basic operations.

Relevant to this, Section 7.2 in [222] requires screw-down non-return valves to be installed at suitable locations to prevent communication between compartments. In other words, compartments are supposed to be independent of each other. In the unlikely event of flooding, the loss of one compartment does not lead to a more severe situation which could cause the whole ship or platform to sink. This is of ultimate importance.

A compliant design is shown in Figure 3.5. As demonstrated above, searching for a list of applicable rules and checking a design against it can be a complex task for human as rules may not always stay in the same section of a massive book of rules. A small thing to note here is that compared to dataset 0, a slightly different symbol of globe valve is used in the drawings of this dataset.

FIGURE 3.5: An example of a compliant design of aspect 1.

FIGURE 3.6: The first example of a non-compliant design of aspect 1.

FIGURE 3.7: The second example of a non-compliant design of aspect 1.

Similar to the case of MARPOL, two examples of a non-compliant design are shown in Figures 3.6 and 3.7. The reason of non-compliance for the first design is the lack of a provision of an emergency and a direct suction, as compared to the compliant example in Figure 3.5. As highlighted, the reason for the second example is that a globe valve which allows bi-directional flow is used and it introduces the possibility of communication among bilge wells, hence violating the rule of no communication between compartments.

Table 3.2 provides the legend for the abbreviations used. For this dataset, based on Section 4, Chapter 13, Part 5 of the LR Rules, 254 compliant designs and 1,534 non-compliant designs (total of 1,788) were obtained. An example of a compliant design in this group is shown in Figure 3.8.

TABLE 3.2: List of the components studied in R^{49} .

P	Pump
SNV	Screw-Down Non-Return Valve
GV	Globe Valve
BuV	Butterfly Valve
BW	Bilge Well
J	Joint
OB	Overboard

FIGURE 3.8: An example of a compliant design of aspect 1 in the form of abbreviations.

3.4 Aspect or Dataset 2: Fuel Oil Transfer System

The fuel oil transfer system is a critical system on-board ships and offshore platforms as it supplies fuel oil to systems such as the engine for operations. Fuel oil is taken to the purifier for processing if necessary.

Based on Part 5, Chapter 13 and 14 in LR rules [222], the main requirements on this part of piping are listed below.

Chapter 13

1. 3.5.1: A forepeak tank is not to be used for the fuel oil.

Chapter 14

1. 4.3.1: A relief valve is to be provided for a pump system which can develop pressure beyond the designed capacity.

2. 4.4.1: A valve is to be provided between a pump and a pipe during the suction and the discharge.
3. 4.7.3: A valve is to be installed on the suction pipe from each double bottom tank.
4. 4.8.1: A valve is to be installed on the tank for the suction pipe coming from a double bottom tank.
5. 4.11.2 and 12.4.6: An overflow is to be provided for settling tanks.
6. 4.18.2: Two service tanks are to be provided.

Figure 3.9 and 3.10 show an example of a compliant and a non-compliant design respectively. The reason for the design in Figure 3.10 to be non-compliant is that it violates the rule 4.18.2 of Chapter 14 mentioned above that two service tanks are needed, as highlighted with the red circle in the figure.

FIGURE 3.9: An example of a compliant design.

FIGURE 3.10: An example of a non-compliant design.

802 compliant and 2404 non-compliant designs have been obtained for aspect 2. Following the practice in previous datasets, for convenience of storage and processing, symbols in P&ID are replaced by abbreviations, as shown in the compliant design in Figure 3.11 and another non-compliant design in Figure 3.12. A complete legend is provided in Table 3.3.

The reason of the design in Figure 3.12 being non-compliant is that a forepeak tank is used for the storage of the fuel oil, as highlighted in the drawing. This violates the rule 3.5.1 of Chapter 13 described above.

FIGURE 3.11: An example of a compliant design.

FIGURE 3.12: An example of a non-compliant design.

TABLE 3.3: Aspect 2: Legend for the abbreviations in a P&ID.

P	Pump
ReV	Relief valve
MV	Metal seated valve
OV	Oil fuel tank with attached valve
EV	Settling tank with attached valve
VV	Service tank with attached valve
Pur	Purifier
Ovf	Overflow
J	Joint
RV	Rubber seated valve
Sys	System
FT	Forepeak tank
OT	Oil fuel tank
ET	Settling tank
VT	Service Tank

3.5 Aspect or Dataset 3: Ballast system on Ship

The ballast system of a ship ensures a proper balance is achieved via the suction or discharge of sea water according to the weight of the ship. In other words, sea water is taken in if the ship is light to increase the stability and removed if the ship is too heavy. An inappropriate design of the ballast can lead to an improper suction or discharge of sea water and may cause the sinking of the ship ultimately. As shown in the compliant design in Figure 3.13, starting from the suction of the sea water from the sea chest, the ballast water is pumped to the ultraviolet ballast water treatment system before being moved to the ballast tanks. When necessary, the ballast water in the tanks can be pumped through the same circuit to be discharged overboard, by opening and closing a set of valves according to the requirements.

LR rules [222] are used as the reference for compliance here. Chapter 13, Part 5 of the rules specifies the requirements for the piping layout.

In particular, the sub-section 6.2 specifies that a pump for general service (G.S.) including the bilge can be used for the ballast, with a requirement that it must be able to return to the service for the bilge when it is needed. This requirement applies to the fire system too, which means that the G.S. pump must be able to attend to the fire system once it is needed.

Section 11.1 of the same part also highlights the requirement of a standby arrangement for the ballast pumping. An arrangement of two pumps with a ballast pump and another ballast or G.S. pump is acceptable. However, a provision of two G.S. pumps without a ballast pump is not acceptable, as shown in Figure 3.14.

Besides, non-acceptable practices in the design of a ballast system also include using the bilge or fuel oil tank for the purpose of ballast. Linking bilge or fuel oil tank to the ballast system could cause pollution to the ballast water.

568 compliant and 4544 non-compliant designs have been obtained. Similarly, for an easier storage and retrieval, components in the designs are also stored and represented as abbreviations, the legend for this dataset is available in Table 3.4. An example is shown in Figure 3.15 for the case of compliant. All designs handled in this work are assumed to be in a vector form where the connectivity information can be extracted without going through any image processing.

FIGURE 3.13: An example of a compliant design of aspect 3.

FIGURE 3.14: An example of a non-compliant design of aspect 3.

TABLE 3.4: List of the components studied in R^{196} .

GsP	General service pump
BlP	Ballast Pump
SNV	Screw-Down Non-Return Valve
BuV	Butterfly Valve
BT	Bilge Tank
BIT	Ballast Tank
Fr	Fire system
UTS	Ultraviolet Ballast Water Treatment System
OvB	Overboard
J	Joint
SC	Sea Chest
Flt	Filter
UFr	Use for Fire System
FoT	Fuel Oil Tank

FIGURE 3.15: Aspect 3: An example of a compliant design in the form of abbreviations.

3.6 Aspect or Dataset 4: Isolation of Bilge System

On top of the existing regulations on the ballast system, Section 7.3 of Chapter 13, Part 5 requires that the bilge to be isolated from the ballast system. However, a suitable connection to a common ejection is acceptable, provided that it does not cause pollution to the ballast water.

A compliant design and a non-acceptable design is shown in figure 3.16 and 3.17. The reason for the design in Figure 3.16 to be compliant is that while the bilge is handled by a common ejection, valves are provided at suitable locations to prevent a communication with the ballast system. The reason of non-acceptance for figure 3.17 is that a screw-down non-return valve is expected at the location of the butterfly valve highlighted in the diagram for safety and environmental purposes.

FIGURE 3.16: An example of a compliant design of aspect 4.

FIGURE 3.17: An example of a non-acceptable design of aspect 4.

256 compliant and 1342 non-compliant designs have been collected for this aspect.

3.7 Summary

In summary, this chapter has discussed the five important aspects of the piping designs on-board a ship or offshore platform, each of which leads to a dataset consisting of thousands of compliant and non-compliant piping designs. These will be used for the performance study of the methodology developed, which will be discussed in the next chapter.

Chapter 4

Histogram of Connectivity for Linear Classification

4.1 INTRODUCTION

This chapter introduces a machine learning based approach to perform an automated evaluation of a piping design. Given a P&ID as the input, the developed framework is to tell if the layout of the design is compliant or non-compliant, according to a set of international rules or guidelines. There are four main components in the framework:

1. The input to the system is a set of P&IDs, each of which is assigned a label (compliant or non-compliant) by experienced engineers.
2. Each P&ID i is represented by a point $\mathbf{x}_i \in R^d$ and has an assigned binary value y_i (1 for compliant and -1 for non-compliant).
3. A machine learning algorithm is used for the training using the provided set of $\mathbf{x}_i$ and corresponding y_i where $0 \leq i < N$. In other words, the algorithm tries to learn or capture the relationship between $\mathbf{x}_i$ and y_i .
4. A new or unseen P&ID is given to the machine or computer for a prediction on the compliance based on the same mathematical representation, the output is 1 if it is compliant and -1 if non-compliant.

The methodology starts with the transformation of a piping design into the corresponding mathematical representation, known as a Histogram of Connectivity. Next, the approach is extended to analyse the reason of non-compliance beyond a binary decision on the compliance of the design. The introduction of the ability to determine the reason of non-compliance is expected to bring the industrial adoption of this work one step closer. This framework can be extended to other closely related domains such as electrical or graph based designs.

The remaining of the chapter is organised as follows. Section 4.2 discusses why the ability to interpret beyond explicit rules is needed. The introductions to the idea of a Histogram of Connectivity and performing a linear classification using SVM are presented in Section 4.3, 4.4 and 4.5. A method to analyse the reason of a non-compliance is presented in Section 4.6. Lastly, experimental results, a visualisation of models obtained, the performance of HoC with other classifiers, discussions and summary are presented.

4.2 Need of Interpretation Beyond Explicit Rules

As shown through the examples in the five piping datasets, while some non-compliant designs can be identified in a relatively straightforward manner such as the existence of a connection between the oil residue tank and the oily bilge water holding tank, some situations require an interpretation such as the existence of a bi-directional valve at certain part of a layout is not acceptable because it violates certain requirements.

It is extremely hard to encode all possibilities around a specific part of a design into explicit rules. Further difficulties appear when engineers need or expect a method or system that can generalise and work on many different aspects of a piping design.

4.3 Histogram of Connectivity (HoC)

Before the mathematical analysis of a design using machine learning, given a P&ID, it is transformed into a descriptor (a vector $\mathbf{x}$ in R^d space) which captures the essential engineering properties of the drawing as explained in the following sections.

4.3.1 Requirements

In terms of layout, a P&ID has two main properties to be captured, the attributes of each component in the design and how these components are connected. The descriptor developed should be able to capture these two pieces of information well while giving some flexibility for some non-critical changes among the compliant designs.

There are two general requirements that apply in this process of designing a descriptor:

4.3.1.1 Independence to numbering or labelling sequence

The final descriptor is independent of the numbering order of the components, if there is any. In other words, if each of n components inside a P&ID is assigned a unique identity such as a number or name, which is quite common in a process design, the descriptor should be independent of how each component inside the design is numbered or named.

4.3.1.2 Independence to number of connections

A certain level of independence to the number of connections is also expected. This is to enable the developed descriptor to take both large and small P&IDs. It is not desirable to favour a design because of its large or small size. A normalisation by the total number of connections can help to achieve this, by dividing the value of each dimension of the descriptor by the total number of connections.

4.3.2 Assumptions and Limitations

Following the requirements established above, an assumption is needed to develop a description mechanism. It is assumed that there are at most T types of components in every P&ID, this list of T types is the same for every P&ID in the dataset. Hence the total number of the type of connections is T^2 . The direction of the connection between two components is taken into consideration, i.e., component m connected to component n is different from component n connected to component m .

In addition, as the focus of this work is to evaluate if the general layout of a P&ID is compliant, the positional information in the design such as the actual position of a pump or a pipeline is not considered in this work. Note that the number of components in a design is denoted by N .

Algorithm 1 Establishing the matrix of connectivity.

```

1: Initialise a matrix  $M$  of size  $T \times T$ 
2: for  $0 \leq i < T$  do
3:   for  $0 \leq j < T$  do
4:      $M_{ij} = 0$ 
5:   end for
6: end for
7: for each component  $i = 0, \dots, N - 1$  do
8:   for each component  $j = 0, \dots, N - 1$  do
9:     if there is a connection from component  $i$  to  $j$  then
10:       $k = \text{Type of component } i$ 
11:       $l = \text{Type of component } j$ 
12:       $M_{kl} = M_{kl} + 1$ 
13:    end if
14:   end for
15: end for

```

4.3.3 Attributes of Each Component

Component specific information considered here is limited to the type of component. For each component in the design, its type is determined and registered for further processing.

4.3.4 Connectivity among Components

The connectivity information among the components is essentially captured in the form of a matrix of connectivity M of a size $T \times T$, where M_{kl} is increased by 1 if there is a connection from a component of type k to a component of type l . Algorithm 1 provides the details about this. It is important to note that the order of the component type in the matrix of connectivity and the HoC should remain consistent so that each dimension in the HoC corresponds to one specific type of connection. The recording of frequency of each connection type can be done subsequently. Figure 4.1 provides an example to illustrate this process.

FIGURE 4.1: Establishing the matrix of connectivity and the HoC of a P&ID.

4.3.5 Histogram of Connectivity (HoC)

The method proposed in this work is to develop a final descriptor using the concept of Histogram of Connectivity (HoC) for a P&ID. This is motivated by the idea that the existence of certain types of connection can serve as hints that the design is non-compliant.

As an example, a connection from the oil residue tank to the oily bilge water holding tank indicates a non-compliance. The HoC checks this by looking into the frequency of each type of connection appearing in the design. If such connections exist, a conclusion that it is non-compliant can be made.

In addition to looking at each connection separately, the existence of certain combinations of connection such as the oily bilge water holding tank to a globe valve

and the globe valve to the oil residue tank could mean that these two tanks are connected via a globe valve, leading to a non-compliance.

The HoC captures the overall distribution of the frequency of the connectivity types for the learning and the prediction by an SVM later. As highlighted earlier, a normalisation is performed to prevent the linear SVM from preferring larger or smaller design.

Similarly, this can also be applied to the other parts of the design such as checking the connections between the oil residue tank and the filter, the oily water separator and the bilge system.

Figure 4.2 and 4.3 show an example of a comparison of HoC signatures of compliant and non-compliant designs (from examples in Figure 3.4, 3.2, 3.8 and 3.6) for each aspect respectively. Clear distinctions between the compliant and non-compliant ones can be seen under very challenging conditions. As discussed earlier, the designs are actually very similar except small variations at certain parts, the mathematical description process is expected to capture the small but key differences for subsequent learning.

FIGURE 4.2: A comparison of the HoC of a compliant design (blue) and a non-compliant design (red) under dataset 0 in R^{144} (x axis: Dimension, y axis: Value or frequency of the connection seen).

FIGURE 4.3: A comparison of the HoC of a compliant design (blue) and a non-compliant design (red) under dataset 1 in R^{49} (x axis: Dimension, y axis: Value or frequency of the connection seen).

As a point in T^2 -dimensional space, the final descriptor of a P&ID $\mathbf{x} \in R^{T^2}$ is obtained by concatenating each row of the matrix of connectivity M which is obtained in the previous section, a normalisation of $\mathbf{x}$ is also carried out. Details are presented in Algorithm 2.

Algorithm 2 The HoC for a P&ID.

```

1:  $N_c = 0$ 
2: Initialise  $\mathbf{x} \in R^{T^2}$ 
3:  $\mathbf{x}_i = 0$  for  $0 \leq i < T^2$ 
4: for each row  $i = 0, \dots, T - 1$  of  $M$  do
5:   for each column  $j = 0, \dots, T - 1$  of  $M$  do
6:      $\mathbf{x}_{T \times i + j} = M_{ij}$ 
7:      $N_c = N_c + M_{ij}$ 
8:   end for
9: end for
10: for  $0 \leq i < T^2$  do
11:    $\mathbf{x}_i = \frac{\mathbf{x}_i}{N_c}$ 
12: end for

```

4.4 Properties of HoC

The two main properties of the HoC are discussed further in this section before the learning through a linear SVM is carried out.

4.4.1 Segmentation and Analysis

As demonstrated through the five aspects of piping, a huge and complex platform or ship design is typically separated (or segmented) into different parts before it is analysed according to the applicable rules. The usual segmentation is done by dividing the design according to functional descriptions or responsibilities of each part, such as the bilge, the ballast or the cargo.

The automated approach proposed also follows this. Hence the bilge system of the machinery space and the oil residue handling system are studied separately. This also brings down the possible number of component types that need to be studied at a time, as discussed in the next section.

4.4.2 Abstraction of Component Type

An important aspect to consider in forming an HoC is that there can be many different types for each component like a valve and a tank. For example, valves mentioned in this work have already covered butterfly, screw-down non-return, globe

and ball valves, there may be more than 50 types [223] which are not covered, similarly for tanks. This leads to a direct issue of feature vectors extracted being in a very high dimensional space which can increase the computational complexity significantly. While a linear SVM can deal with learning in a high-dimensional space [224] [177] which means that the HoC can deal with many types of components in one go, it is preferred to keep the number of component types studied in each case, T , as low as possible.

The idea of abstraction is introduced to address this issue. Many types of components may not appear at all in the specific part of a design or do not affect the compliance. These components can either be neglected or grouped into one category.

While there are many types of tanks for different purposes in the marine or chemical industry, it is not necessary to include all of them into each case study. Depending on the requirement of each part of a ship or an offshore platform, some kind of abstraction can be done. For example, the main information an HoC needs for the case of MARPOL is that if the tank is used for the storage of the oil residue or the oily bilge water holding tank, other information is not needed at this moment. This reduces the number of types to be recorded to two.

For a valve, the choice between a ball valve and a globe valve does not affect the compliance on the case of MARPOL, hence it is possible to combine these two types into a common type. Section 4.7 will present additional experimental details of abstracting these two types into a common type.

However, manual intervention may be needed to deal with exceptional scenarios such as an unseen type of component appearing in a test design and the machine is unable to include this in the original feature space learnt. With the current version of HoC, a possible solution is to map this unknown type of component to a suitable type in the existing list of the components for that space. Using the case of the bilge system as an example, if an unseen type of valve connected immediately to a bilge well is seen, depending on whether the use of this type is accepted for bilge, it can be taken as either a screw-down non-return (accepted) or globe valve (not accepted), as illustrated in Figure 4.4. The key thing to note here is that during the formation of feature space for the HoC, the separation of components into individual types does not consider the physical type only, functional abstraction is also taken into consideration.

Looking at this from another angle, since the machine has not learnt about this

FIGURE 4.4: An illustration of abstraction.

new type of component, it is unable to predict. An engineer's input on the relevance of this new type to an existing type in the list of learnt types of component would allow the machine to generalise and hence perform a prediction. This piece of information is necessary. In other words, a certain level of domain knowledge is needed to establish the feature space of each aspect.

4.5 Learning from Examples

Rules or guidelines in word form can be hard to understand for someone without much experience in the field, sometimes the contents are designed to be more generic so that all scenarios could be covered. When a specific case or design comes in, experts will have to start from a high level to check on the compliance. As demonstrated through the examples earlier, this makes it hard to incorporate the meaning of the rules or guidelines for design into systems and apply them to a P&ID directly. The idea here is to collect a large number of examples (or designs in this work) and perform reasoning using machine learning algorithms. In this work, a design with bad practice is put into the class of non-compliant.

A linear SVM based learning is used to perform the linear separation of the compliant and non-compliant designs in T^2 -dimensional space or R^{T^2} subsequently.

4.5.1 Linear Support Vector Machine (SVM)

Using instances in the dataset created, each P&ID is transformed to a point in R^{T^2} space using the concept of HoC. Each point belongs to either the group of compliant or non-compliant designs, a label is determined manually based on the existing rules and guidelines beforehand.

During the training, a linear SVM [171] is used to determine the hyperplane $\mathbf{w}^T \mathbf{x} + b = 0$ (where $\mathbf{w} \in R^{T^2}$ and $b \in R$) that best separates the two classes of points. Algorithm 3 provides a summary of the training.

After the training, when a new P&ID which the system has not seen before is given, an evaluation is performed using Algorithm 4. The algorithm gives a final output of 1 or -1 , 1 indicates a compliant design and -1 indicates a non-compliant design.

Algorithm 3 Summary of the training stage of a linear SVM.

- 1: **for** each P&ID in training set **do**
 - 2: Determine the matrix of connectivity M
 - 3: Determine the HoC representation $\mathbf{x}$
 - 4: Store the $\mathbf{x}$ and the corresponding label y
 - 5: **end for**
 - 6: Perform training on a linear SVM using all $\mathbf{x}$ and y
-

Algorithm 4 Classification of a P&ID.

- 1: Determine the matrix of connectivity M of given P&ID
 - 2: Determine the HoC representation $\mathbf{x}$
 - 3: $y = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$
 - 4: **return** y
-

4.6 Search for Non-Compliant Connections

Beyond a binary prediction (if a design is compliant or non-compliant), it would be beneficial to the industry if the machine can provide further information or highlight parts relevant to the non-compliance in a design.

Models learnt using linear SVMs (in the primal form) could be used to highlight possible reasons of non-compliance. This is achieved by determining possible connections that lead to the non-compliance in a design.

Mathematically, the most influential dimensions in the dot product of an HoC and

the weight vector of a learnt SVM are to be identified. This list of dimensions or type of connections is known as Non-Compliant Connections (NCC) in this work. More details are provided in Algorithm 5. The list L_c returned contains the dimensions (or types of connection) which may have caused the non-compliance and this could be used for further investigations.

Algorithm 5 Determining the NCC.

```

1: Extract the HoC  $\mathbf{x}$  of given P&ID
2: Initialise  $L_c$  as an empty list
3: Initialise  $\mathbf{y}_c \in R^{T^2}$ 
4: for  $0 \leq i < T^2$  do
5:    $\mathbf{y}_{c_i} = \mathbf{w}_i \mathbf{x}_i$ 
6:   if  $\mathbf{y}_{c_i} < 0$  then
7:     Insert  $i$  into  $L_c$ 
8:   end if
9: end for
10: return  $L_c$ 

```

Figure 4.5 and 4.6 show non-compliant examples of datasets 0 and 1 with the connections causing non-compliance highlighted respectively, the connections near and parallel to the red horizontal or vertical line in the figures are in NCC.

The reason for the design in Figure 4.5 to be non-compliant is the globe valve which allows flow in two directions are used. This leads to a possible scenario of the oil residue entering the oily bilge water holding tank and the subsequent processing systems.

The reason for the design in Figure 4.6 to be non-compliant is that the circled part (a butterfly valve which allows bi-directional flow) of the design allows possible communication among the compartments or bilge wells, hence violating the rule of no communication among compartments.

FIGURE 4.5: An example of an NCC of dataset 0.

FIGURE 4.6: An example of an NCC of dataset 1.

4.7 Experimental Setup and Results

Algorithm 3-5 were implemented using C++, Ubuntu, OpenCV [225] and LIBLINEAR [226].

Each dataset of designs obtained was divided into training and testing set randomly, 50% each. Using the training set, the mathematical description and the linear SVM based learning were performed. A performance evaluation on designs from the testing set took place after the learning stage was completed. This process was repeated 10 times to investigate the performance on generalisation. The average (μ) and the standard deviation (ν) are reported.

As the number of the positive and the negative samples are different, an adjustment using the weight in a linear SVM was done as described later. The effect of the weight on the performance was also studied.

In addition to HoC, the performance of some straightforward concepts was also studied. Instead of recording the frequency of each connection type, Histogram of Components can be obtained by recording the frequency of each component type and normalised. The performance of HoC was compared with this. Next, doing a random guess (50% for compliant and 50% for non-compliant) sequentially from the first to the last test sample was also carried out for a comparison.

For a better understanding of the HoC, a training with less data was also carried out and studied.

4.7.1 Dataset 0: Regulation 12, MARPOL

The type of component, T was set to 12 in the experiment. The components in Table 3.1 of Appendix were taken into the study. Hence the linear separation using SVM was done at $R^{T^2} = R^{144}$.

As the ratio of the negative (non-compliant) to the positive (compliant) samples was about 10, a weight of 10 was given to the positive class during the training of the linear SVM as a compensation, a standard weight w of 1 was used for the negative class. The prediction accuracies are shown in Table 4.1, in the form of the average and the standard deviation, in the form of percentage.

In addition, experiments using $T = 11$ with globe valve and ball valve combined into a common type were also carried out to study the effects of abstraction. The list of the components is available at Table 4.4. The results are provided in Table

TABLE 4.1: The accuracies on dataset 0 in R^{144} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$w = 8$	87.97 ± 5.26	89.74 ± 2.36
$w = 10$	90.78 ± 5.23	81.44 ± 2.04
$w = 12$	91.88 ± 5.35	74.95 ± 2.82

TABLE 4.2: The performance of the different types of descriptors on dataset 0 ($\mu \pm \nu$).

Type	Positive (%)	Negative (%)
Histogram of Components	88.75 ± 4.59	69.51 ± 1.93
Random Guess	48.59 ± 7.01	49.95 ± 1.24

TABLE 4.3: The accuracies on dataset 0 in R^{121} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$w = 8$	89.38 ± 4.53	92.12 ± 3.60
$w = 10$	89.38 ± 4.53	81.68 ± 4.13
$w = 12$	91.09 ± 6.42	75.13 ± 3.08

TABLE 4.4: The list of the components studied in R^{121} .

P	Pump
SNV	Screw-Down Non-Return Valve
V	Bi-Directional Valve
ORT	Oil Residue (Sludge) Tank
OBW	Oily Bilge Water Holding Tank
SDC	Standard Discharge Connection
FOB	Filter/Oily Water Separator/Bilge System
Inc	Incinerator
ORS	Oil Residue (Sludge) Service Tank
NV	Non-Return Valve
J	Joint

4.3, compared with the case without abstraction in Table 4.1, no major variation was observed. This has validated the theoretical assumption that the developed method offers a certain level of flexibility in terms of abstraction of the type of components considered, without significant effects on the performance.

4.7.2 Dataset 1: Bilge System of the Machinery Space

T was set to 7 in the experiments. Hence the linear separation using the SVM was done at $R^{T^2} = R^{49}$. Similar to the first case, as the ratio of the number of

TABLE 4.5: The accuracies on dataset 1 in R^{49} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$w = 5$	94.17 ± 4.27	91.21 ± 1.45
$w = 6$	98.98 ± 1.53	85.65 ± 1.45
$w = 7$	99.92 ± 0.25	81.94 ± 2.31

TABLE 4.6: The performance of the different types of descriptors on dataset 1 ($\mu \pm \nu$).

Type	Positive (%)	Negative (%)
Histogram of Components	95.35 ± 4.93	72.23 ± 2.02
Random Guess	48.43 ± 4.43	50.09 ± 2.00

TABLE 4.7: The accuracies on dataset 2 in R^{225} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$w = 2$	38.35 ± 2.00	80.21 ± 2.07
$w = 3$	80.47 ± 3.06	70.25 ± 1.20
$w = 4$	93.99 ± 1.78	67.65 ± 0.78

TABLE 4.8: The performance of the different types of descriptors on dataset 2 ($\mu \pm \nu$).

Type	Positive (%)	Negative (%)
Histogram of Components	93.14 ± 1.17	66.04 ± 1.04
Random Guess	50.37 ± 2.58	48.89 ± 0.81

the negative (non-compliant) to the positive (compliant) samples was about 6, a weight of 6 was given to the class of compliant for a compensation. The prediction accuracies for dataset 1 using HoC and the other descriptors are shown in Table 4.5 and 4.6 respectively.

4.7.3 Dataset 2: Fuel Oil Transfer System

T was set to 15 in the experiments. Hence the linear separation using the SVM was done at $R^{T^2} = R^{225}$. Similar to the first case, as the ratio of number of the negative (non-compliant) to the positive (compliant) samples was about 3, a weight of 3 was given to the class of compliant for a compensation. The prediction accuracies for dataset 2 using HoC and the other descriptors are shown in Table 4.7 and 4.8 respectively.

TABLE 4.9: The accuracies on dataset 3 in R^{196} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$w = 6$	82.32 ± 2.95	89.79 ± 1.52
$w = 8$	95.00 ± 0.89	70.72 ± 1.36
$w = 10$	96.90 ± 1.17	58.29 ± 1.08

TABLE 4.10: The performance of the different types of descriptors on dataset 3 ($\mu \pm \nu$).

Type	Positive (%)	Negative (%)
Histogram of Components	96.20 ± 2.46	73.59 ± 0.80
Random Guess	51.16 ± 2.87	49.90 ± 0.91

TABLE 4.11: The accuracies on dataset 4 in R^{196} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$w = 4$	39.14 ± 11.03	99.55 ± 0.54
$w = 5$	91.25 ± 5.94	86.18 ± 3.21
$w = 6$	99.30 ± 1.19	64.84 ± 2.47

4.7.4 Dataset 3: Ballast System

T was set to 14 in the experiments. Hence the linear separation using the SVM was done at $R^{T^2} = R^{196}$. Similar to the first case, as the ratio of number of the negative (non-compliant) to the positive (compliant) samples was about 8, a weight of 8 was given to the class of compliant for a compensation. The prediction accuracies for dataset 3 using HoC and the other descriptors are shown in Table 4.9 and 4.10 respectively.

4.7.5 Dataset 4: Isolation of Bilge System

Same as dataset 3, T was set to 14 in the experiments. Hence the linear separation using the SVM was done at $R^{T^2} = R^{196}$. Similar to the first case, as the ratio of number of the negative (non-compliant) to the positive (compliant) samples was about 5, a weight of 5 was given to the class of compliant for a compensation. The prediction accuracies for dataset 4 using HoC and the other descriptors are shown in Table 4.11 and 4.12 respectively.

TABLE 4.12: The performance of the different types of descriptors on dataset 4 ($\mu \pm \nu$).

Type	Positive (%)	Negative (%)
Histogram of Components	88.83 ± 2.02	90.00 ± 2.47
Random Guess	50.86 ± 3.82	50.09 ± 1.41

TABLE 4.13: The accuracies on dataset 0 in R^{144} ($\mu \pm \nu$).

Setting	Performance
$w = 8$	99.16 ± 0.78
$w = 10$	99.31 ± 0.79
$w = 12$	99.12 ± 0.83

TABLE 4.14: The accuracies on dataset 1 in R^{49} ($\mu \pm \nu$).

Setting	Performance
$w = 5$	93.94 ± 0.52
$w = 6$	93.94 ± 0.52
$w = 7$	93.94 ± 0.52

TABLE 4.15: The accuracies on dataset 2 in R^{225} ($\mu \pm \nu$).

Setting	Performance
$w = 2$	75.80 ± 0.46
$w = 3$	75.80 ± 0.46
$w = 4$	75.80 ± 0.46

TABLE 4.16: The accuracies on dataset 3 in R^{196} ($\mu \pm \nu$).

Setting	Performance
$w = 6$	74.72 ± 0.61
$w = 8$	74.72 ± 0.61
$w = 10$	74.72 ± 0.61

4.7.6 NCC

The performance of HoC in identifying the reason of a non-compliance is presented in this section. The ground truth or actual answer in the form of a list of connections in a design that cause the non-compliance is provided manually earlier. It is considered a correct match if a dimension in L_c matches a dimension in the ground truth.

The percentage of the non-compliant connections in the ground truth, also known as the recall, is reported, the accuracies for the five datasets are presented in Table 4.13 to 4.17, respectively.

TABLE 4.17: The accuracies on dataset 4 in R^{196} ($\mu \pm \nu$).

Setting	Performance
$w = 4$	71.94 ± 1.13
$w = 5$	71.94 ± 1.13
$w = 6$	71.81 ± 0.95

TABLE 4.18: The accuracies on dataset 0 in R^{144} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	50.58 ± 14.24	97.94 ± 2.41
$c = 15$	77.77 ± 5.23	98.90 ± 1.84
$c = 20$	87.48 ± 3.25	99.25 ± 1.35

TABLE 4.19: The accuracies on dataset 1 in R^{49} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 1$	95.64 ± 3.07	77.13 ± 3.32
$c = 3$	98.87 ± 2.29	88.65 ± 1.73
$c = 5$	99.95 ± 0.16	92.79 ± 1.78

TABLE 4.20: The accuracies on dataset 2 in R^{225} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	83.32 ± 4.38	70.43 ± 1.94
$c = 15$	85.76 ± 5.03	69.88 ± 2.08
$c = 20$	87.35 ± 4.94	69.42 ± 2.20

TABLE 4.21: The accuracies on dataset 3 in R^{196} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	10.24 ± 9.43	99.13 ± 0.91
$c = 15$	63.91 ± 12.98	93.77 ± 3.71
$c = 20$	84.29 ± 7.13	88.48 ± 3.51

TABLE 4.22: The accuracies on dataset 4 in R^{196} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	53.22 ± 20.40	86.98 ± 9.95
$c = 15$	84.83 ± 7.20	76.49 ± 6.28
$c = 20$	97.22 ± 3.28	74.99 ± 5.09

4.7.7 Less Data

Table 4.18 to 4.22 show the performance of the classifier trained with 20% of the data.

4.8 Visualisation of Model

It is important to have a detailed look at the models (or the weight of each dimension) obtained by learning using a linear SVM. This is done by establishing the correspondence of each dimension with the weight assigned by the linear SVM. For 10 models obtained under 10 iterations of the same setting, the average of each dimension was taken and the 10 dimensions with the highest absolute average weight were selected.

Figure 4.7 shows the plot of an example of the model obtained for dataset 0 obtained in R^{144} . Similarly, Figure 4.8 shows an example of dataset 1.

As shown in Table 4.23 to 4.28, the results or the most influential types of connection are aligned with the established engineering knowledge. The types of connection that lead to an immediate indication of non-compliance is identified successfully such as ORT to OBW (the oil residue tank to the oily bilge water holding tank) and BW to BW (a bilge well to a bilge well).

In addition, the combinations of certain connections could also lead to a non-compliance, HoC has also been able to capture certain parts of this, such as a combination of ORT to GV (the oil residue tank to a globe valve) and GV to OBW (a globe valve to the oily bilge water holding tank) could indicate that ORT is connected to OBW (non-compliant).

FIGURE 4.7: The visualisation of a model in R^{144} for dataset 0. x -axis: Dimension of the learnt model, y -axis: Weight.

FIGURE 4.8: The visualisation of a model in R^{49} for dataset 1. x -axis: Dimension of the learnt model, y -axis: Weight.

TABLE 4.23: Set 0: The dimensions in R^{144} with the highest weights.

Dimension	Weight	Corresponding Type of Connection
42	6.84	ORT to FOB
51	6.61	OBW to ORT
40	6.60	ORT to OBW
38	5.98	ORT to GV
27	5.71	GV to ORT
28	4.73	GV to OBW
34	4.35	GV to J
22	4.06	SNV to J
142	3.65	BV to J
122	3.13	J to GV

TABLE 4.24: Set 1: The dimensions in R^{49} with the highest weights.

Dimension	Weight	Corresponding Type of Connection
32	11.35	BW to BW
31	10.99	BW to BuV
25	10.00	BuV to BW
29	4.84	BW to SNV
35	4.83	J to P
26	4.63	BuV to J
38	4.09	J to BuV
40	3.51	J to J
7	2.74	SNV to P
14	2.59	GV to P

4.9 The performance on other classifiers

For a better understanding on the ability of HoC to generalise in terms of the choice of classifiers, learning with two other classical classifiers, namely random forest [227] and deep neural network (DNN) [186] was also carried out. As the aim here is to have a general understanding on how well these classifiers work with HoC and not to optimise, the values for the key parameters were chosen empirically. A maximum depth of 5 was used for random forest. Learning through DNN with two layers of eight neurons was performed. The performance of each is shown in Table 4.29 and 4.30 respectively.

TABLE 4.25: Set 0: The dimensions in R^{121} with the highest weights.

Dimension	Weight	Corresponding Type of Connection
35	7.12	ORT to V
39	6.91	ORT to FOB
37	6.66	ORT to OBW
47	6.62	OBW to ORT
25	5.72	V to ORT
32	5.35	V to J
26	4.50	V to OBW
110	4.30	J to P
21	4.23	SNV to J
10	4.21	P to J

TABLE 4.26: Set 2: The dimensions in R^{225} with the highest weights.

Dimension	Weight	Corresponding Type of Connection
32	11.27	MV to MV
47	5.11	OV to MV
34	4.89	MV to EV
62	4.34	EV to MV
167	4.33	FT to MV
212	4.24	VT to MV
197	4.13	ET to MV
182	4.11	OT to MV
77	3.93	VV to MV
128	3.81	J to J

TABLE 4.27: Set 3: The dimensions in R^{196} with the highest weights.

Dimension	Weight	Corresponding Type of Connection
86	11.37	Fr to SNV
73	10.65	BlT to BuV
185	9.97	FoT to BuV
59	9.85	BT to BuV
47	9.72	BuV to BlT
130	8.23	J to BT
139	7.83	J to FoT
131	7.57	J to BlT
58	6.88	BT to SNV
55	6.79	BuV to FoT

TABLE 4.28: Set 4: The dimensions in R^{196} with the highest weights.

Dimension	Weight	Corresponding Type of Connection
59	6.81	BT to BuV
32	5.65	SNV to BT
16	4.73	BIP to SNV
29	4.73	SNV to BIP
130	4.48	J to BT
46	3.69	BuV to BT
28	3.38	SNV to GsP
2	3.32	GsP to SNV
73	2.82	BLT to BuV
47	2.77	BuV to BLT

TABLE 4.29: The performance of random forest ($\mu \pm \nu$).

Dataset	Positive (%)	Negative (%)
0	76.56 ± 11.15	96.62 ± 1.76
1	98.50 ± 1.36	94.68 ± 3.41
2	90.12 ± 2.48	85.94 ± 1.95
3	98.98 ± 0.63	80.94 ± 2.59
4	94.92 ± 4.19	92.55 ± 2.60

TABLE 4.30: The performance of deep neural network ($\mu \pm \nu$).

Dataset	Positive (%)	Negative (%)
0	75.31 ± 38.04	99.64 ± 1.05
1	93.23 ± 21.41	99.60 ± 1.11
2	79.45 ± 29.60	84.83 ± 15.04
3	97.64 ± 2.42	99.56 ± 0.61
4	96.64 ± 2.08	98.84 ± 0.89

4.10 Discussion and Further Work

As demonstrated in the previous section, a combination of the HoC and a linear SVM addresses the problem of evaluating a piping design in an automated way well and gives excellent performance. This is achieved by searching for a hyperplane in the space of HoC for the separation of the compliant and non-compliant designs or points.

The effects of weights on learning have also been studied. While giving too much weight to either side brought down the overall performance, the ratio of the number of negative to the positive samples appears to be a good choice for the optimal performance.

The proposed method uses the linear SVM classifier which is fast, leading to a better efficiency in the industry. The automated approach can be seen as an incremental learning process. When the system sees more designs, it gets more experienced and hence is able to capture some experience of the industry in design evaluation. The advantages of fast training make a re-training after sufficient new designs come into the system efficient, which could improve the classification performance of the system over time.

As demonstrated through the search for NCC and the visualisation, the learnt SVM models managed to capture or highlight the types of connections that have significant influence over the compliance of a design, through the weights assigned to individual dimensions. It is encouraging to see that what the SVM has learnt is aligned with the established engineering understanding.

However, the method proposed requires a significant amount of training samples so that the sample set provided for training is representative of the population. This requires some efforts in data preparation and may not be perfect all the time. A solution to this is to develop an algorithm to automatically generate a lot more non-compliant designs by introducing different noises (both randomly and systematically) to a smaller set of designs collected from industry experts.

Compared to straightforward concepts like Histogram of Components and doing a random guess (50% chance of correct guess for each sample), the HoC outperforms significantly and has demonstrated its capability in dealing with compliance in the domain of piping design.

One thing worth noting here is that it is not the aim of this work to optimise the choice of representation and classifiers as this is the first step in the community towards an automated design appraisal and it is probably too early to fine-tune. With the five main aspects of piping design of a ship covered, the aim of this work is to look into the feasibility of having a machine learning based methodology for an automated evaluation of a piping design.

In addition, the robustness of the HoC towards the choice of different classifiers and abstraction has also been demonstrated through experiments. As the aim of this work is not to optimise, it is encouraging to see the developed methodology including the HoC works well with different classifiers.

4.10.1 Directions to Investigate in the Future

While the linear classifier based approach provides good performance, it would be interesting to look at learning the non-linear relationship of the data to achieve a better prediction accuracy. However, in the context of SVM, which kernel is suitable and to be used here remains a question to be investigated.

The search for a single hyperplane in R^d space for a linear separation can be very hard especially when the data is highly non-linearly distributed. This task may be simplified by dividing the single hard classification mechanism into a series of weak classifications, similar to the concept of boosted cascade of weak classifiers [228].

4.11 Summary

In summary, this chapter presents an approach to automate the evaluation process of piping designs in the marine and offshore industry, according to established international rules and guidelines. After transforming a P&ID into a Histogram of Connectivity (HoC) which captures the essential engineering properties of a P&ID, a linear SVM based learning is performed to capture the relationship between the compliant and the non-compliant designs in R^{T^2} space for a separation. The idea of determining the reason of a non-compliance using the learnt SVM models is introduced to allow engineers to follow up if a design is predicted as non-compliant. In average, prediction accuracies of at least 75% have been observed with relatively small standard deviations, this serves as a demonstration of the performance and the robustness of the developed methodology. Similarly, for the case of searching for non-compliant connections (NCC), performance of about 70% with relatively low standard deviations has been observed. However, HoC has not looked into the connectivity information which is beyond the immediate neighbours, this will be looked into in the next chapter.

Chapter 5

Pyramid of Connectivity Histogram

5.1 Introduction

Following the previous method for automating the evaluation of a piping layout using a Histogram of Connectivity (HoC) and Linear Support Vector Machines (SVM) [229], this work introduces the concept of Pyramid of Connectivity Histogram (PoC) and its integration into the developed framework for the automated evaluation of a piping design. Given a piping design, the corresponding mathematical representation PoC is determined, mapping the design to R^d space theoretically. During the training, each design in the training set is transformed into the corresponding PoC and a linear SVM is used to search for the most suitable hyperplane for the separation of the two groups of points (compliant and non-compliant) in the R^d space. After learning, given an unseen piping design, the machine can provide a prediction on the compliance of the design by mapping it to the same space and checking which side of the hyperplane it lies at.

By integrating PoC with HoC and a linear SVM, significant improvements in the prediction accuracy can be obtained in the five existing datasets of piping design, which has been discussed earlier.

The remaining of the chapter is organised as follows, the concept of Pyramid of Connectivity Histogram (PoC) and integration with a linear SVM are discussed in Section 5.2 and 5.3 respectively. Next, the details on experimental setup and

results are provided in Section 5.4. Lastly, the visualisation of the learnt SVM models, the study of the performance of PoC in the space of HoC, a discussion and a summary are presented in Section 5.5, 5.6, 5.7 and 5.8.

5.2 Pyramid of Connectivity Histogram

The main idea of Pyramid of Connectivity Histogram (PoC) is introduced in this section.

On top of an HoC which looks at the connectivity information among the immediate neighbours (or components) only, as illustrated in Figure 5.1, a PoC attempts to look beyond that. For example, if the component A is connected to B and B is connected to C , A is considered to be connected to C at the resolution level of 2 in a PoC, but not in an HoC. This is motivated by the idea that the connectivity information between the indirect neighbours could also contain useful information as the indication of a non-compliance.

FIGURE 5.1: Beyond immediate neighbours.

The concept of looking beyond the immediate neighbours is extended to multiple levels of resolution. In other words, the process of extracting the connectivity (or HoC) is carried out at each level of resolution (known as l). For example, this is done at the level 2 where each pair of components is connected via exactly one intermediate component. The final descriptor is obtained by concatenating the HoC extracted at different levels, the HoC extracted at each level is normalised before the concatenation takes place. This is summarised in Algorithm 6 and 7.

In this work, the number of types of component is known as T and the number of levels used in forming a mathematical representation of multi-resolution is known as r . The final descriptor x is rT^2 -dimensional, or $x \in R^{rT^2}$.

Figure 5.2 and 5.3 show an example of a compliant and a non-compliant design of dataset 0 respectively. Three types of joints are added to study the ability of an HoC and a PoC to generalise, leading the analysis to be done in R^{225} .

Algorithm 6 Extracting the HoC at each level l .

```

1: for  $0 \leq l < r$  do
2:   Initialise  $x_l \in R^{T^2}$ ,  $x_{lj} = 0$  where  $0 \leq j < T^2$ 
3:    $s = 0$ 
4:   for  $0 \leq p < N$  do
5:     for  $0 \leq q < N$  do
6:       if component  $p$  and  $q$  are connected at level  $l$  then
7:          $m = \text{Type of component } p$ 
8:          $n = \text{Type of component } q$ 
9:          $t = mT + n$ 
10:         $x_{lt} = x_{lt} + 1$ 
11:         $s = s + 1$ 
12:       end if
13:     end for
14:   end for
15:   for  $0 \leq i < T^2$  do
16:      $x_{li} = \frac{x_{li}}{s}$ 
17:   end for
18:   Store  $x_l$  as the HoC for level  $l$ 
19: end for

```

Algorithm 7 Forming a PoC.

```

1: Initialise  $f \in R^{rT^2}$ ,  $f_j = 0$  where  $0 \leq j < rT^2$ 
2: for  $0 \leq l < r$  do
3:   for  $0 \leq j < T^2$  do
4:      $f_{l \times T^2 + j} = x_{lj}$ 
5:   end for
6: end for

```

FIGURE 5.2: A compliant design of dataset 0.

FIGURE 5.3: A non-compliant design of dataset 0.

Figure 5.4 provides a general overview on the concept of PoC. As shown in the figure, HoCs at different levels (denoted as h_l where l corresponds to the level number) are extracted and concatenated together. Figure 5.5 and 5.6 provide a comparison of the PoC extracted from a compliant and non-compliant design for dataset 0 and 2 respectively. The x -axis refers to the dimension and the y -axis refers to the frequency. Clear distinctions could be observed between the signature of the compliant and the non-compliant ones.

FIGURE 5.4: An illustration of Pyramid of Connectivity Histogram.

FIGURE 5.5: A comparison of PoCs for dataset 0, blue: compliant design in figure 3.4, red: non-compliant design in figure 3.3. x -axis: Dimension, y -axis: Value or frequency.

FIGURE 5.6: A comparison of PoCs for dataset 2, blue: compliant design in figure 3.11, red: non-compliant design in figure 3.12. x -axis: Dimension, y -axis: Value or frequency.

For the design in Figure 5.7 under dataset 0 (with three additional types of joints which function similarly), while the HoC in Figure 5.8 is unable to capture directly the information that the oily residue tank (denoted as ORT) is connected to the oily bilge water holding tank (OBW), a PoC would be able to capture that at the level of 2, dimension 275 (starting from 1) which corresponds to the connection of ORT to OBW at level 2 in the PoC signature extracted recorded a normalised frequency of 0.07, as shown in the signature in Figure 5.9. This serves as a simple example of the ability of a PoC beyond an HoC.

FIGURE 5.7: An example of a non-compliant design in dataset 0.

FIGURE 5.8: An HoC of a non-compliant design in dataset 0. x -axis: Dimension, y -axis: Value or frequency.

FIGURE 5.9: A PoC of a non-compliant design in dataset 0. x -axis: Dimension, y -axis: Value or frequency.

However, in Figure 5.10, it is also hard for a PoC to capture the information that in the series of the indirect connections as highlighted, a lack of screw-down non-return valve makes it non-compliant, but there is still a good cue in the PoC signature extracted as shown in the highlighted part of Figure 5.11. In particular, dimension 738 which corresponds to a connection of an OBW to a GV at the level of 4 recorded a frequency of 0.2.

FIGURE 5.10: A difficult example of a non-compliant design in dataset 0.

FIGURE 5.11: A PoC of the difficult example of a non-compliant design in dataset 0.

As an example for dataset 2, Figure 5.12 shows an example of a non-compliant design caused by the lack of a relief valve for a pump as highlighted with a red circle in the figure. In the PoC signature extracted as shown in Figure 5.13, the dimension which corresponds to a connection from a pump (denoted as P) to a joint (denoted as J) at the level of 3 recorded a frequency of 0.04.

FIGURE 5.12: An example of a non-compliant design in dataset 2.

FIGURE 5.13: A PoC of a non-compliant design in dataset 2. x -axis: Dimension, y -axis: Value or frequency.

In the reversed direction, Equation 5.1 can be used to find out which level a dimension k corresponds to, known as c . Next, Equation 5.2, 5.3 and 5.4 can be used to determine which type of connection it corresponds to (denoted as b), the output from the series of equations is the dimension which corresponds to a connection from Type b_s to b_t .

$$c = \lfloor \frac{k}{T^2} \rfloor \quad (5.1)$$

$$b = k - T^2 \times c \quad (5.2)$$

$$b_s = \lfloor \frac{b}{T} \rfloor \quad (5.3)$$

$$b_t = b - T \times b_s \quad (5.4)$$

5.3 Learning by Linear Support Vector Machines

Similar to the previous HoC based approach, example based learning is used to allow a machine to achieve a sufficient understanding on applicable rules via a set of examples.

A linear Support Vector Machine (SVM) is used for the learning of compliant and non-compliant designs in all piping datasets by the machine. After each piping design for training is transformed into a PoC, this set of PoC signatures is fed to a linear SVM in the form of points in R^{T^2} space. Next, the machine is expected to search for the best hyperplane in the space to separate the points into two groups which correspond to the compliant and the non-compliant set respectively. After the training, the machine can give a prediction when it is given an unseen piping design. Algorithm 8 and 9 provide a summary of the training and the prediction stage respectively.

Algorithm 8 Learning by a linear SVM.

- 1: Given N_t training samples where D_i is the i -th sample
 - 2: **for** $0 \leq i < N_t$ **do**
 - 3: Determine the PoC x_i for D_i
 - 4: Store x_i
 - 5: **end for**
 - 6: Use this set of x_i and corresponding label y_i to train an SVM
 - 7: Store the w and the b returned by the SVM
-

Algorithm 9 A prediction by the linear SVM.

- 1: Given a design D_t
 - 2: Determine the PoC x_t of D_t
 - 3: $y_t = \text{sign}(w \cdot x_t + b)$
 - 4: **return** y_t as the predicted label
-

5.4 Experimental Setup and Results

Details on the experimental setup and results are provided in this section.

5.4.1 Setup

The implementation was done using C++, LIBLINEAR [230] and OpenCV [225]. The compliant and non-compliant designs in each dataset were separated into two parts, one for training and another for testing, equally and randomly. The designs in the training set were provided to the linear SVM. After the learning, the designs in the testing set were given to the machine for prediction. Each prediction from

TABLE 5.1: The values of T .

Dataset	T
0	144
1	49
2	225
3	196
4	196

TABLE 5.2: The prediction accuracies under dataset 0 ($\mu \pm \nu$).

Setting	Compliant (%)	Non-Compliant (%)
HoC and Linear SVM	91.41 ± 2.68	82.35 ± 2.88
PoC and Linear SVM	96.09 ± 3.85	92.64 ± 1.49

the machine was compared with the corresponding ground truth to determine the accuracy. For a better understanding of the performance in general, this process was repeated for 10 times, the average (μ) and the standard deviation (ν) were determined subsequently. The results are reported in terms of the percentage of correct predictions (out of total number of positive or negative samples). A value of 5 was used for r , T used for each aspect of the piping design is shown in Table 5.1.

For a further evaluation, experiments were carried out with different values of r . In other words, a PoC was extracted with different numbers of level of resolution. This was to study the effects of the resolution on being too coarse (very small r) and too fine (very large r). A (near-) optimal r was to be determined too. As there were more negative than positive samples, a higher weight was given to the positive class during learning by a linear SVM accordingly.

5.4.2 Results

The performance for all the datasets are shown in Table 5.2 to 5.6. In addition, the effects of the different resolutions on the performance are shown in Table 5.7 to 5.8. Improvements were observed when the learning shifted from the space of HoC to PoC.

TABLE 5.3: Performance on dataset 1 ($\mu \pm \nu$).

w	Compliant (%)	Non-Compliant (%)
4	86.46 ± 3.03	95.42 ± 0.90
5	92.99 ± 5.89	95.20 ± 0.81
6	98.58 ± 2.48	94.78 ± 0.75

TABLE 5.4: The prediction accuracies under dataset 2 ($\mu \pm \nu$).

Setting	Compliant (%)	Non-Compliant (%)
HoC and Linear SVM	80.27 ± 1.94	70.42 ± 0.96
PoC and Linear SVM	94.84 ± 1.05	98.79 ± 0.42

TABLE 5.5: Performance on dataset 3 ($\mu \pm \nu$).

w	Compliant (%)	Non-Compliant (%)
6	99.75 ± 0.78	92.23 ± 0.83
7	99.86 ± 0.45	86.54 ± 1.30
8	99.89 ± 0.33	81.13 ± 1.32

TABLE 5.6: Performance on dataset 4 ($\mu \pm \nu$).

w	Compliant (%)	Non-Compliant (%)
2	1.88 ± 2.22	99.93 ± 0.24
3	86.72 ± 7.28	96.84 ± 1.41
4	99.84 ± 0.49	87.32 ± 2.12

TABLE 5.7: The effects of resolution on dataset 0 ($\mu \pm \nu$).

Resolution	Compliant (%)	Non-Compliant (%)
8	94.53 ± 4.18	90.73 ± 1.69
10	94.38 ± 4.67	90.42 ± 1.62

TABLE 5.8: The effects of resolution on dataset 2 ($\mu \pm \nu$).

Resolution	Compliant (%)	Non-Compliant (%)
8	94.86 ± 1.02	99.97 ± 0.04
10	95.11 ± 0.93	99.96 ± 0.04

5.5 Visualisation of the Learnt Models

A deeper look into the learnt models using a linear SVM is done at this section. This is done by looking into the dimension which corresponds to a relatively high absolute weightage in the learnt model. In other words, dimensions with high absolute value of weights (or $|w|$ in Equation 5.5) are investigated further.

$$y = w \cdot x + b \quad x, w \in \mathbb{R}^{rt^2}, y, b \in \mathbb{R} \quad (5.5)$$

Theoretically, this is to study if the understanding of the PoC and linear SVM is aligned with the existing engineering knowledge. This is achieved by investigating if the weight linear SVM assigns to each dimension captures the relative importance of each type of connection.

This was done by taking the average of 10 models obtained through 10 iterations for each aspect and searching for the 10 dimensions with the highest absolute weights.

5.5.1 Dataset 0: Regulation 12, Annex I, MARPOL

Figure 5.14 and 5.15 show an example of the weights learnt using linear SVM for dataset 0 using HoC and PoC respectively. Table 5.9 and 5.10 show the 10 dimensions with the highest absolute weightage in the learnt SVM model.

FIGURE 5.14: The magnitude of the weights learnt for dataset 0 using HoCs.

TABLE 5.9: Top 10 dimensions with the highest absolute magnitude for dataset 0 using HoCs.

Dimension	Absolute Weight	Connection
50	6.77	ORT to OBW
52	6.74	ORT to FOB
64	6.60	OBW to ORT
48	6	ORT to GV
34	5.51	GV to ORT
35	4.89	GV to OBW
41	4.77	GV to J
26	3.93	SNV to J
176	3.26	BV to J
11	3.19	P to J

FIGURE 5.15: The magnitude of the weights learnt for dataset 0 using PoCs. x -axis: Dimension of the learnt model, y -axis: Absolute weight.

5.5.2 Dataset 1: Bilge system of machinery space

The 10 dimensions with the highest weightage in the learnt model for PoCs is shown in Table 5.11.

TABLE 5.10: Top 10 dimensions with the highest absolute magnitude for dataset 0 using PoCs.

Dimension	Absolute Weight	Connection	Level
52	7.56	ORT to FOB	1
277	5.4	ORT to FOB	2
26	5.03	SNV to J	1
64	4.5	OBW to ORT	1
152	4.09	J to SNV	1
48	3.87	ORT to GV	1
267	3.53	GV to BV	2
50	3.49	ORT to OBW	1
956	3.47	ORT to J	5
258	3.41	GV to GV	2

TABLE 5.11: Top 10 dimensions with the highest absolute magnitude for dataset 1 using PoCs.

Dimension	Absolute Weight	Connection	Level
32	9.46	MV to MV	1
31	6.48	MV to ReV	1
25	6.20	ReV to Sys	1
124	4.59	J to EV	3
81	4.53	VV to Pur	2
131	4.27	J to FT	3
222	4.05	VT to OT	5
78	3.76	VV to OV	2
71	3.53	EV to FT	2
58	3.33	OV to ET	2

5.5.3 Dataset 2: Fuel oil transfer system

Similarly, figure 5.16 and 5.17 show an example of the learnt model using HoC and PoC respectively for dataset 2. Table 5.12 and 5.13 show the 10 dimensions with the highest weightage in the learnt model for HoC and PoC respectively.

FIGURE 5.16: The magnitude of the weights learnt for dataset 2 using HoC.
 x -axis: Dimension of the learnt model, y -axis: Absolute weight.

FIGURE 5.17: The magnitude of the weights learnt for dataset 2 using PoC.
 x -axis: Dimension of the learnt model, y -axis: Absolute weight.

5.5.4 Dataset 3: Ballast system

The 10 dimensions with the highest weightage in the learnt model for PoCs is shown in Table 5.14.

TABLE 5.12: Top 10 dimensions with the highest absolute magnitude for dataset 2 using HoC.

Dimension	Absolute Weight	Connection
33	11.27	MV to MV
48	5.13	OV to MV
35	4.97	MV to EV
183	4.23	OT to MV
198	4.2	ET to MV
168	4.17	FT to MV
63	4.14	EV to MV
213	4.06	VT to MV
129	3.77	J to J
78	3.7	VV to MV

TABLE 5.13: Top 10 dimensions with the highest absolute magnitude for dataset 2 using PoC.

Dimension	Absolute Weight	Connection	Level
804	7.59	J to J	4
969	6.95	EV to J	5
738	6.89	EV to MV	4
483	6.73	MV to MV	3
519	6.4	EV to J	3
33	6.14	MV to MV	1
743	5.48	EV to Ovf	4
48	5.18	OV to MV	1
353	4.92	J to Ovf	2
933	4.86	MV to MV	5

TABLE 5.14: Top 10 dimensions with the highest absolute magnitude for dataset 3 using PoC.

Dimension	Absolute Weight	Connection	Level
86	9.73	Fr to SNV	1
128	7.99	J to SNV	1
289	7.64	Fr to J	2
347	6.63	SC to Flt	2
58	6.38	BT to SNV	1
478	6.01	Fr to SNV	2
518	5.60	J to GsP	2
667	5.58	BIT to J	3
143	5.42	SC to BuV	1
653	5.24	BT to J	4

TABLE 5.15: Top 10 dimensions with the highest absolute magnitude for dataset 4 using PoC.

Dimension	Absolute Weight	Connection	Level
326	5.29	J to BT	2
653	4.77	BT to J	4
522	4.67	J to BT	3
59	4.55	BT to BuV	1
519	4.25	J to BIP	3
718	4.12	J to BT	4
914	4.10	J to BT	5
32	4.04	SNV to BT	1
849	3.85	BT to J	5
16	3.28	BIP to SNV	1

5.5.5 Dataset 4: Isolation of bilge system

The 10 dimensions with the highest weightage in the learnt model for PoCs is shown in Table 5.15.

5.6 Integrating Multi-Resolution into HoC

An additional study was carried out to understand the performance of having multi-resolution in the original space of HoC, as summarised in Algorithm 10. In other words, the connections at different levels are processed together and recorded as a single histogram $x \in R^{T^2}$.

Intuitively, this should also outperform HoC as it contains more information at multi-levels of resolutions. This was also repeated using different resolutions (r), at 5, 10 and 20. The results for datasets 0 and 2 are shown in Table 5.16 and 5.17 respectively. The visualisation of the learnt models for datasets 0 and 2 are provided in Figures 5.18 to 5.19. The 10 dimensions with the highest absolute weightage in the learnt model for dataset 0 and 2 are shown in Tables 5.18 and 5.19. In general, performance of higher than 70% has been observed.

Algorithm 10 Multi-Resolution in the original space of HoC.

```

1: for  $0 \leq l < r$  do
2:   Initialise  $x \in R^{T^2}$ ,  $x_{lj} = 0$  where  $0 \leq j < T^2$ 
3:   for  $0 \leq p < N$  do
4:     for  $0 \leq q < N$  do
5:       if component  $p$  and  $q$  are connected at level  $l$  then
6:          $m = \text{Type of component } p$ 
7:          $n = \text{Type of component } q$ 
8:          $t = mT + n$ 
9:          $x_t = x_t + 1$ 
10:      end if
11:    end for
12:  end for
13: end for

```

TABLE 5.16: Performance on dataset 0 ($\mu \pm \nu$).

Resolution	Compliant (%)	Non-Compliant (%)
5	81.25 ± 3.36	77.41 ± 2.19
10	74.79 ± 4.46	74.79 ± 2.26
20	74.49 ± 3.03	73.07 ± 1.89

TABLE 5.17: Performance on dataset 2 ($\mu \pm \nu$).

Resolution	Compliant (%)	Non-Compliant (%)
5	82.81 ± 8.99	68.8 ± 3.48
10	86.56 ± 6.56	74.17 ± 2.49
20	86.56 ± 6.56	74.14 ± 2.42

FIGURE 5.18: The magnitude of the weights learnt for dataset 0 using PoC in R^{225} .

FIGURE 5.19: The magnitude of the weights learnt for dataset 2 using PoC in R^{225} .

FIGURE 5.20: An example of a PoC in R^{225} for dataset 0.

FIGURE 5.21: An example of a PoC in R^{225} for dataset 2.

TABLE 5.18: Top 10 dimensions with the highest absolute magnitude for dataset 0 using PoC in R^{225} .

Dimension	Absolute Weight	Connection
26	4.8	SNV to J
21	4.72	SNV to SDC
52	3.9	ORT to FOB
48	3.77	ORT to GV
41	3.62	GV to J
64	3.15	OBW to ORT
2	3.1	P to SNV
63	3.02	OBW to GV
42	2.96	GV to BV
50	2.9	ORT to OBW

TABLE 5.19: Top 10 dimensions with the highest absolute magnitude for dataset 2 using PoC in R^{225} .

Dimension	Absolute Weight	Connection
33	10.72	MV to MV
129	8.2	J to J
69	7.97	EV to J
63	5.34	EV to MV
38	4.71	MV to Ovf
68	4.03	EV to Ovf
144	3.63	RV to J
126	3.53	J to VV
130	3.51	J to RV
48	3.09	OV to MV

5.7 Discussion

As demonstrated in Section 5.4, integrating a PoC with an HoC and a linear SVM has led to significant improvements in the performance. This also fits the hypothesis in this chapter that the information on the indirect connectivity or connectivity among the components that are connected via a series of other components contains good amount of cues for the indication on a non-compliance.

Besides, in Section 5.5, it has been shown that the results of learning by a linear SVM match the general understanding of a piping design. The dimensions with the highest weights are among the key aspects that an engineer would look at during a manual design appraisal. This is encouraging. Compared with an HoC, a PoC introduces the ability to capture the level specific connectivity information that is

important to the study of compliance.

For example, in Table 5.10, a PoC managed to identify dimension 956 which corresponds to the connection from an ORT (oil residue tank) to a J (joint) at level of 5. This could be a good cue for non-compliance that an SNV (screw-down non-return valve) is missing between the ORT and the OBW (oily bilge water holding tank), which is required by dataset 0. Besides, in Table 5.13 for dataset 2, the 743-rd dimension which corresponds to the connection from an EV (a settling tank with a valve) to an Ovf (overflow) is highlighted by the machine. This is aligned with the established rule that an overflow needs to be provided for the settling tanks.

The effects of the resolution have also been studied. A resolution r of 5 appears to be a good choice for a near-optimal performance based on the experiments carried out on the five datasets, without being too coarse or fine. However, it is also worth noting that the effects are relatively limited, a PoC performed significantly better than an HoC regardless of the choice of the value for r .

It is counter intuitive that adding the information of multi-resolution in the original space of HoC (R^{T^2}) leads to a slightly worse performance for all datasets. A possible reason for this is that too much information was squeezed into the original feature space and hence caused an overflow of information. As shown in Figure 5.20 (dataset 0) and 5.21 (dataset 2), the PoC extracted in R^{225} is much denser than the one in R^{1125} . However, it is really encouraging that the PoC is able to capture the key attributes that indicate a non-compliance in this relatively cramped space. As shown in Table 5.18 and 5.19, the PoC is able to highlight the types of connection such as an ORT to an FOB (filter / oily water separator / bilge system) for dataset 0 and an EV to an Ovf for dataset 2.

The connectivity information at each level contains different information or cue about the compliance. Based on the experimental results, it seems the best to extract the HoCs, concatenate the level specific signature and handle them separately.

Another interesting observation in the visualisation of the learnt models is that a PoC appears to be able to capture the shape of the learnt HoC model at each level. As shown in Figure 5.14 and 5.15 for dataset 0, Figure 5.16 and 5.17 for dataset 2, the shape of the PoC model appears to be a repeated HoC model for each case. This is also aligned with the common understanding that many types of connections could remain important at different levels for the study of compliance.

5.7.1 Further Works

One thing interesting to explore would be doing a sufficient learning using limited examples. Collecting a huge number of designs for one aspect of piping may not always be possible in the industry due to a range of issues such as privacy and limited sources of data. For these situations, a learning with much less data such as 10 designs per case may be expected. This poses a tough challenge to the linear SVM based framework established which relies heavily on the statistical study of the samples.

One possibility is to reduce the amount of samples needed by introducing domain knowledge to achieve a sufficient understanding. This can be done by transforming existing engineering knowledge in the form of text into a suitable form for use by the SVM. However, this process of capturing the existing knowledge and searching for a suitable mathematical representation remains a question to be addressed.

While a PoC performs well in the five datasets which are representative in the piping design of a ship and offshore platform, it is necessary to perform further study on other aspects of piping to understand the ability of a PoC to generalise.

5.8 Summary

This chapter has presented a mathematical representation known as Pyramid of Connectivity Histogram for the automated evaluation of a piping design. On top of the previously established methodology using Histogram of Connectivity and linear Support Vector Machines, Pyramid of Connectivity Histogram looks at the connectivity information in a piping design beyond the immediate neighbours. The connectivity among the components that are connected via a series of other components is extracted together and the final representation is a point in R^{rT^2} where T is the number of types of component and r is the levels of resolution. The method has resulted in improved performance in all five existing piping datasets, compared to the HoC based methodology. With performance of about 90% observed in this chapter, the goal of using the developed methodology for the automated early warning of a non-compliance of a piping design is made possible.

Chapter 6

Integrated Learning

6.1 Introduction

As the piping design of a ship covers many aspects and functions such as the bilge, the ballast, the cargo and the fresh water system, it might not always be possible to obtain enough examples of both compliant and non-compliant designs for each aspect of the ship or offshore platform as there are less examples for certain parts. In addition, due to the huge amount of existing rules in the industry, it is also time consuming to separate the existing designs into individual aspects before carrying out the process of learning. This leads to a need for the machine to learn from a large pool of designs based on different rules in a less organised way, combine the understanding obtained from each design and perform the reasoning in general. In other words, the machine is expected to learn from designs of different aspects and the only label available beforehand is if the design is compliant or non-compliant. In order to solve this problem, a concept of Integrated Learning (IL) for piping is introduced in this chapter. The main idea is to find a way to learn about designs of different aspects of a ship and rules in one go. This would improve the efficiency of the developed system. Similar to the HoC based approach established, the concept of integrated learning begins with the representation of each piping design as an HoC. Next, HoCs extracted are mapped to a common feature space containing the possible types of component and hence connectivity of all designs. A linear SVM is used to search for a single hyperplane to separate the compliant from the non-compliant designs. This is known as Histogram of Connectivity in Common

Space (HoCC).

In addition, the ability of the HoCC and linear SVM based approach to learn and generalise from a limited amount of designs is also investigated in this chapter. This is of significant importance from an industrial perspective as a huge amount of designs might not always be available for each aspect. While integrated learning enables the machine to learn some common understanding through examples from different aspects, it is also desirable for the machine to learn and generalise from fewer designs.

In summary, this chapter makes three contributions:

1. The introduction of the idea of integrated learning.
2. The use of HoCC to achieve integrated learning.
3. A demonstration of the effectiveness of the HoCC and linear SVM based approach with limited training examples.

The remaining of the chapter is organised as follows: Starting with an introduction to the idea of Histogram of Connectivity in Common Space (HoCC), integrated learning and learning with limited examples using linear SVM are presented in Section 6.2, 6.3 and 6.4. Lastly, experimental results, a discussion and a summary are presented in Section 6.5, 6.6 and 6.7.

6.2 HoC in Common Space

Before an integrated learning can be carried out, it is necessary to move the analysis into a common feature space. As the list of the components for each aspect is different, this is done by combining all lists into a complete list of the component types seen in all designs. Assuming the length of the list of the components is T_1 and T_2 for aspect 1 and 2, the length of the combined list T would be larger than or equal to the length of T_1 and T_2 but smaller than or equal to the length of $T_1 + T_2$. There may be common components in both lists, each type of component needs to be registered once only. A Histogram of Connectivity (HoC) moved to a common space is known as a Histogram of Connectivity in Common Space (HoCC). The idea of common space can be easily extended to N aspects or case studies,

as summarised in Algorithm 11, a common space formed is known as R^{T^2} . The formation of HoCC is summarised in Algorithm 12.

For a better illustration, figure 6.1 and 6.2 show an example of a compliant and a non-compliant design of the dataset 0. The reason for the design in figure 6.2 to be non-compliant is that the globe valve highlighted in the design introduces the possibility of the oil residue (sludge) from entering the filter, the oily water separator and the bilge system.

FIGURE 6.1: A compliant design under aspect 0.

FIGURE 6.2: A non-compliant design under aspect 0.

As shown in Table 3.1 and 3.4, there are common components such as the screw-down non-return valve and the joint. However, it is also worth noting that due to a need for a more effective representation, a certain level of abstraction may exist at different aspects of a design. For example, in aspect 0 (MAPROL: Figure 6.1 and 6.2), the bilge system is taken as a component where it is actually one of the major systems on a ship. It is actually sufficient to be represented as a component since the main aim is to check if the oil residue (sludge) can go to the bilge. At this moment, abstraction is not handled further, which means the bilge system is taken as a component in HoCC, and it is different from a bilge tank. To a certain extent,

abstraction saves the number of component types that need to be considered and allows for a more efficient representation.

Algorithm 11 The establishment of a common space.

```

1: Initialise an empty list of the components  $L$ 
2: for each aspect  $i$  with the list of the components  $L_i$  do
3:   if  $L_i$  does not exist in  $L$  then
4:     Insert  $L_i$  into  $L$ 
5:   end if
6: end for
7: return  $L$ 

```

For dataset 0, figure 6.3 shows the HoCC of a compliant (figure 3.4) and a non-compliant design (figure 3.3). For dataset 3, the HoCC of a compliant (figure 3.13) and a non-compliant design (figure 3.14) are shown in figure 6.4. It can be seen that there are clear distinctions in both comparisons even under very challenging situations. The small variation between the compliant and the non-compliant design has been captured well by the representations or signatures.

FIGURE 6.3: A comparison of the HoCC for dataset 0. Blue: Compliant design, Red: Non-compliant design. x -axis: Dimension, y -axis: Value or frequency.

FIGURE 6.4: A comparison of the HoCC for dataset 3. Blue: Compliant design, Red: Non-compliant design. x -axis: Dimension, y -axis: Value or frequency.

Algorithm 12 The formation of an HoCC.

- 1: **inputs**
 - 2: A piping design and the established list of the common components L
 - 3: Determine the connectivity matrix M
 - 4: Determine the HoCC $\mathbf{x}$ according to L
 - 5: **return** $\mathbf{x}$
-

6.3 Integrated Learning

Similar to the previous approach, after representing each design as an HoCC (or a point $\mathbf{x} \in R^{T^2}$), these points are provided to a linear SVM for learning. After a hyperplane is found in the space, a prediction on the compliance of an unseen design can be performed. This is summarised in Algorithm 13 and 14. Figure 6.5 provides an illustration of the integrated learning.

FIGURE 6.5: An illustration of integrated learning.

Algorithm 13 The summary of integrated learning.

- 1: **for** each aspect of the piping design **do**
 - 2: **for** each piping design in the training set **do**
 - 3: Determine HoCC $\mathbf{x}$ in the common space R^{T^2}
 - 4: Store $\mathbf{x}$ and the corresponding label y
 - 5: **end for**
 - 6: **end for**
 - 7: Perform training on a linear SVM using all $\mathbf{x}$ and y
-

Algorithm 14 Checking the compliance of a piping design.

- 1: Determine the connectivity matrix M of a given piping design
 - 2: Determine the HoCC $\mathbf{x}$
 - 3: $y = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$
 - 4: **return** y
-

6.4 Learning with Limited Examples

The idea of learning with limited examples is also introduced here to make the automated design appraisal closer for an adoption by the industry. As discussed earlier, a huge amount of samples is not always possible for each aspect of the rules. It is beneficial to have a machine which can learn and generalise from a smaller set of samples and be put to use for non-compliance detection in unseen designs. A study on the ability of HoCC and linear SVM to generalise is performed in this chapter.

Using HoCC as the representation, the compliant and non-compliant designs are separated randomly into a training and testing set to study the ability of the trained SVM to generalise. More details will be presented in the next section.

6.5 Experimental Setup and Results

Algorithms 11, 12, 13 and 14 were implemented using C++, Ubuntu, OpenCV [225] and LIBSVM [231].

For integrated learning, besides learning with all, each possible combination of the three selected datasets (0, 3 and 4) was explored, using 20% of the compliant and the non-compliant data for training, the remaining for the performance evaluation. The separation was done randomly.

During the pre-processing, the designs in the training set were transformed into HoCCs and learning using a linear SVM was done. Testing using the HoCC signature of each design in the testing set was done subsequently. This was repeated 10 times for a better understanding of the performance, the mean (μ) and the standard deviation (ν) of the prediction accuracies were reported, in the form of percentage. A training of each SVM took less than a minute typically.

As there were more negative (or non-compliant) examples, the weight of the positive class (or the class with label 1) was adjusted similarly.

The effects of the amount of training data were also investigated by adjusting the amount of training data provided for the learning. In addition to a default of 20%, experiments with 10% and 15% were also carried out. This is augmented by a study of the effects of another important parameter in the linear SVM, the cost (or C). This cost parameter is related to the margin of the hyperplane found or the cost of classifying the training examples wrongly. The whole set of experiments was repeated using three different values of C , 10, 15 and 20.

6.5.1 All Datasets

Table 6.1 to 6.5 show the performance of integrated learning in R^{1521} , when $T = 39$. The positive column shows the prediction accuracies of the trained model on the positive or compliant samples in the testing set, while the negative column shows the ones for the non-compliant samples. Under the very challenging condition of learning with very limited data, accuracies of about 70 – 90% have been observed, which is promising for the aim of doing an early warning of possible non-compliance in a piping design. In fact, the model performed at the range of 80 – 90% most of the time except for the compliant samples of the dataset 0 and the non-compliant designs of the dataset 2.

TABLE 6.1: Total IL: The accuracies on dataset 0 in R^{1521} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 2$	54.44 ± 7.06	98.53 ± 0.64
$c = 3$	60.00 ± 5.64	97.98 ± 0.38
$c = 4$	66.03 ± 6.83	97.54 ± 0.54

TABLE 6.2: Total IL: The accuracies on dataset 1 in R^{1521} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 2$	98.03 ± 1.54	95.57 ± 0.87
$c = 3$	99.69 ± 0.55	96.69 ± 0.60
$c = 4$	99.92 ± 0.25	96.94 ± 0.50

TABLE 6.3: Total IL: The accuracies on dataset 2 in R^{1521} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 2$	97.48 ± 1.26	66.87 ± 0.47
$c = 3$	97.63 ± 1.33	66.98 ± 0.53
$c = 4$	97.83 ± 1.34	67.02 ± 0.57

TABLE 6.4: Total IL: The accuracies on dataset 3 in R^{1521} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 2$	92.39 ± 1.34	93.90 ± 0.69
$c = 3$	94.96 ± 1.58	93.08 ± 0.83
$c = 4$	96.27 ± 1.30	92.74 ± 0.92

TABLE 6.5: Total IL: The accuracies on dataset 4 in R^{1521} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 2$	88.67 ± 2.91	84.38 ± 2.07
$c = 3$	94.22 ± 2.71	83.92 ± 1.48
$c = 4$	96.64 ± 1.48	84.41 ± 1.20

However, it is also necessary to note that a relatively higher value is needed for the cost parameter during the learning of the SVM model. While this lowers the margin of the data points to the hyperplane, it could also increase the vulnerability of the learnt model towards noises as a relatively small increase in the error could cause a data point to cross the hyperplane which may lead to a wrong prediction.

TABLE 6.6: IL 0 and 3: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	35.44 ± 12.78	99.14 ± 1.21
$c = 15$	64.66 ± 6.02	99.33 ± 0.73
$c = 20$	78.64 ± 7.53	99.72 ± 0.42

TABLE 6.7: IL 0 and 3: The accuracies on dataset 3 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	13.41 ± 8.42	98.67 ± 1.33
$c = 15$	69.54 ± 14.30	91.09 ± 3.78
$c = 20$	87.14 ± 8.35	85.87 ± 2.99

TABLE 6.8: IL 0 and 3 with 10% of the data: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	6.12 ± 9.93	98.94 ± 2.50
$c = 15$	18.62 ± 10.27	98.38 ± 2.58
$c = 20$	32.76 ± 11.93	97.78 ± 2.04

TABLE 6.9: IL 0 and 3 with 10% of the data: The accuracies on dataset 3 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	0.35 ± 0.75	99.93 ± 0.20
$c = 15$	8.11 ± 7.70	98.73 ± 1.64
$c = 20$	29.86 ± 17.64	94.83 ± 4.23

6.5.2 Dataset 0 and 3: MARPOL and Ballast System

The accuracies of integrated learning using dataset 0 and 3 are presented in this sub-section. Table 6.6 and 6.7 present the accuracies obtained for dataset 0 and 3, with 20% of the training data. As discussed earlier, three different values of C were used for a better understanding of the performance. Similar to the results above, accuracies of about 70 – 90% have been observed.

Table 6.8 and 6.9 show the performance of the SVM model learnt using 10% of the data. Aligned with the general belief that more data leads to better prediction performance, slightly worse performance has been seen for the case of learning with 10% of the data.

TABLE 6.10: IL 0 and 3 with 15% of the data: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	26.61 ± 19.28	97.83 ± 3.02
$c = 15$	43.85 ± 16.54	97.77 ± 3.38
$c = 20$	60.46 ± 7.63	98.52 ± 2.79

TABLE 6.11: IL 0 and 3 with 15% of the data: The accuracies on dataset 3 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	3.69 ± 5.07	99.54 ± 0.71
$c = 15$	31.10 ± 20.04	95.50 ± 4.87
$c = 20$	68.43 ± 14.03	89.74 ± 6.79

TABLE 6.12: IL 0 and 4: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	48.83 ± 12.40	98.32 ± 1.90
$c = 15$	78.16 ± 4.98	98.99 ± 1.59
$c = 20$	86.31 ± 4.08	99.34 ± 1.15

TABLE 6.13: IL 0 and 4: The accuracies on dataset 4 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	45.90 ± 22.24	89.59 ± 10.59
$c = 15$	88.78 ± 10.47	77.38 ± 8.86
$c = 20$	98.83 ± 3.05	74.15 ± 5.81

Table 6.10 and 6.11 show the performance of the SVM model learnt using 15% of the data. The performance appeared to be much better than the case of 10%. This could be an indication that learning with 15% of the data is the lowest that one should go. However, the performance is probably meeting the minimal expectations from the engineers in terms of the prediction accuracy.

6.5.3 Dataset 0 and 4: MARPOL and Isolation of Bilge

The accuracies of integrated learning using dataset 0 and 4 are presented in this sub-section. Table 6.12 and 6.13 present the accuracies obtained for dataset 0 and 4 respectively, with 20% of the training data. With a reasonable choice of values for the key parameters involved, performance of 70 – 90% has been obtained.

TABLE 6.14: IL 0 and 4 with 10% of the data: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	15.09 ± 11.22	97.57 ± 4.33
$c = 15$	30.86 ± 14.71	96.58 ± 3.78
$c = 20$	43.45 ± 16.34	96.12 ± 3.63

TABLE 6.15: IL 0 and 4 with 10% of the data: The accuracies on dataset 4 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	15.15 ± 19.22	93.12 ± 11.67
$c = 15$	30.56 ± 21.27	86.64 ± 12.80
$c = 20$	46.93 ± 22.24	81.09 ± 13.03

TABLE 6.16: IL 0 and 4 with 15% of the data: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	38.99 ± 16.04	95.84 ± 4.97
$c = 15$	55.96 ± 12.36	97.45 ± 3.12
$c = 20$	71.65 ± 7.96	98.52 ± 2.09

TABLE 6.17: IL 0 and 4 with 15% of the data: The accuracies on dataset 4 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	32.71 ± 13.66	89.18 ± 8.91
$c = 15$	56.15 ± 8.25	80.73 ± 4.14
$c = 20$	76.93 ± 12.34	79.19 ± 3.63

Table 6.14 and 6.15 show the performance of the SVM model learnt using 10% of the data. Significantly worse performance was observed, possibly down to half of the case of 20% in terms of the prediction accuracy for the positive samples sometimes. The performance also varied much more over the 10 iterations.

Table 6.16 and 6.17 show the performance of the SVM model learnt using 15% of the data. With $c = 20$, reasonable prediction accuracies were obtained.

6.5.4 Dataset 3 and 4: Ballast System

The accuracies of integrated learning using dataset 3 and 4 are presented in this sub-section. Table 6.18 and 6.19 present the accuracies obtained for dataset 3

TABLE 6.18: IL 3 and 4: The accuracies on dataset 3 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	59.21 ± 21.42	94.27 ± 4.48
$c = 15$	89.63 ± 5.01	85.12 ± 6.94
$c = 20$	94.86 ± 2.12	85.60 ± 5.45

TABLE 6.19: IL 3 and 4: The accuracies on dataset 4 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	72.88 ± 10.03	83.43 ± 7.65
$c = 15$	89.27 ± 4.26	69.09 ± 5.04
$c = 20$	92.54 ± 4.66	71.55 ± 3.60

TABLE 6.20: IL 3 and 4 with 10% of the data: The accuracies on dataset 3 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	2.19 ± 3.35	99.85 ± 0.29
$c = 15$	30.68 ± 23.76	96.07 ± 4.40
$c = 20$	57.25 ± 20.76	90.06 ± 7.03

TABLE 6.21: IL 3 and 4 with 10% of the data: The accuracies on dataset 4 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	5.15 ± 7.00	99.40 ± 0.92
$c = 15$	38.57 ± 22.61	88.73 ± 10.31
$c = 20$	71.08 ± 7.50	75.67 ± 11.21

and 4 respectively, with 20% of the training data. Similar to the previous cases, performance of about 70 – 90% has been obtained.

Table 6.20 and 6.21 show the performance of the SVM model learnt using 10% of the data. Similarly, the effort of trying to optimise the performance via the key parameter c was not very successful.

Table 6.22 and 6.23 show the performance of the SVM model learnt using 15% of the data. Decent performance was seen, except the prediction accuracy for the negative samples of the dataset 4, which is at about 65%.

TABLE 6.22: IL 3 and 4 with 15% of the data: The accuracies on dataset 3 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	18.61 ± 15.23	98.16 ± 2.11
$c = 15$	71.01 ± 18.02	89.85 ± 5.40
$c = 20$	86.98 ± 8.11	85.08 ± 5.42

TABLE 6.23: IL 3 and 4 with 15% of the data: The accuracies on dataset 4 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	35.32 ± 21.76	93.09 ± 7.60
$c = 15$	84.63 ± 6.47	74.38 ± 10.86
$c = 20$	92.11 ± 5.84	67.88 ± 7.59

TABLE 6.24: IL 0, 3 and 4: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	38.93 ± 10.65	99.21 ± 1.04
$c = 15$	65.83 ± 6.74	99.47 ± 0.73
$c = 20$	79.90 ± 6.47	99.73 ± 0.38

TABLE 6.25: IL 0, 3 and 4: The accuracies on dataset 3 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	64.64 ± 21.48	91.43 ± 5.92
$c = 15$	91.14 ± 5.91	81.72 ± 7.53
$c = 20$	95.36 ± 2.18	82.89 ± 5.80

TABLE 6.26: IL 0, 3 and 4: The accuracies on dataset 4 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	77.61 ± 9.64	79.95 ± 8.35
$c = 15$	91.37 ± 4.91	65.47 ± 5.41
$c = 20$	94.39 ± 4.52	69.01 ± 4.13

6.5.5 Dataset 0, 3 and 4: Isolation of Bilge and Ballast System

The accuracies of integrated learning using dataset 0, 1 and 2 are presented in this sub-section. Table 6.24, 6.25 and 6.26 present the accuracies obtained for dataset 0, 1 and 2 respectively, with 20% of the training data. Similarly, three different values of C were used for a better understanding of performance. Consistent performance within the range of 70 – 90% was seen.

TABLE 6.27: IL 0, 3 and 4 with 10% of the data: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	6.64 ± 10.16	98.86 ± 2.52
$c = 15$	20.86 ± 9.78	98.19 ± 2.81
$c = 20$	33.88 ± 10.57	97.83 ± 2.07

TABLE 6.28: IL 0, 3 and 4 with 10% of the data: The accuracies on dataset 3 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	3.20 ± 4.27	99.47 ± 0.92
$c = 15$	33.48 ± 22.12	94.58 ± 5.39
$c = 20$	58.36 ± 19.86	88.79 ± 6.67

TABLE 6.29: IL 0, 3 and 4 with 10% of the data: The accuracies on dataset 4 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	5.58 ± 8.13	99.11 ± 1.35
$c = 15$	43.03 ± 23.16	86.37 ± 10.62
$c = 20$	73.51 ± 8.64	72.55 ± 10.55

TABLE 6.30: IL 0, 3 and 4 with 15% of the data: The accuracies on dataset 0 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	28.53 ± 19.92	97.90 ± 2.62
$c = 15$	48.07 ± 12.86	97.91 ± 3.36
$c = 20$	62.39 ± 7.66	98.61 ± 2.80

Table 6.27, 6.28 and 6.29 show the performance of the SVM model learnt using 10% of the data. With this case of learning three aspects of piping at one go, a similar degradation in terms of performance was observed.

Table 6.30, 6.31 and 6.32 show the performance of the SVM model learnt using 15% of the data. As expected, the performance is approximately between the case of 20 and 10%.

TABLE 6.31: IL 0, 3 and 4 with 15% of the data: The accuracies on dataset 3 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	23.95 ± 16.87	96.62 ± 3.39
$c = 15$	75.53 ± 15.69	87.61 ± 5.29
$c = 20$	89.42 ± 7.26	82.94 ± 5.39

TABLE 6.32: IL 0, 3 and 4 with 15% of the data: The accuracies on dataset 4 in R^{576} ($\mu \pm \nu$).

Setting	Positive (%)	Negative (%)
$c = 10$	43.12 ± 23.78	89.74 ± 9.25
$c = 15$	87.94 ± 5.68	71.60 ± 10.18
$c = 20$	93.58 ± 4.40	66.27 ± 7.52

6.6 Discussion and Further Work

As shown in Section 6.5, the HoCC and linear SVM based approach works well in achieving the aim of integrated learning, with all the five piping aspects presented. With as low as 20% of the data used for training, the method is still able to differentiate the compliant and the non-compliant designs reasonably well. With more data coming in, the performance of the method is expected to increase significantly. It was observed that the cost parameter C in the linear SVM had significant influence on the prediction accuracy. Theoretically, C relates to the margin of the hyperplane from the training points or the cost of classifying the training examples wrongly. While a small C can result in a high training error, as shown by the low performance of the trained SVM in the previous section, it is not desirable to have a system being too sensitive to where the hyperplane lies. In other words, it is more advisable to maintain a relatively low C to keep the ability of a trained SVM to generalise.

Besides, it is encouraging to see the developed approach works well with significantly less designs. Using 20% for the training data and the remaining 80% for testing, an accuracy around 70% to 80% could be achieved. However, further improvement is needed before an adoption by industry. Related to this, the performance actually improved when more training data was given. This is aligned with the common understanding that more training data leads to better predictions. With integrated learning, it is possible now to learn in a less organised way by putting the compliant and the non-compliant designs into a basket using HoCC

and linear SVM. This makes the aim of an automated design appraisal on a piping layout one step closer to the adoption by the industry as it reduces the manual efforts needed for the preparation before a learning significantly.

6.6.1 Directions to Investigate in the Future

While the HoCC demonstrates excellent performance on five existing datasets, a further testing on more aspects of piping is needed to study the ability of the method to generalise.

Beyond an overall layout, mechanical and related properties are also key factors that need to be evaluated if a design is compliant. For example, the type of fluid or gas going through each segment of the piping needs to be taken into consideration during the design so that the pipeline can sustain with a sufficient strength including stress and thermal. The HoCC may need to be extended further to include this aspect into the process of compliance evaluation.

Next, deep learning has been very successful in computer vision [205] and natural language [186] recently. It would also be interesting to investigate if the use of deep neural networks can improve the performance of HoCC further, through the high amount of non-linearities across the layers of neurons.

6.7 Summary

In summary, this chapter presents a method to do integrated learning on piping designs of different aspects using HoCC. It starts by mapping the mathematical representation of each piping design (or P&ID) into a common feature space. Next, learning by a linear SVM takes place. Under a very challenging condition of learning with limited data, the developed method with HoCC and linear SVM has shown an encouraging performance of at least 70% approximately. However, the standard deviations have appeared to be relatively higher and have gone beyond 10%, although not frequently seen. This would need a further investigation and improvement. While the results are closer to a deployment of the learnt models for an early warning on non-compliance during the process of a design, the gap for further improvement has also been highlighted.

Chapter 7

Transfer Learning

7.1 Introduction

This chapter uses transfer learning on the latest CNN architectures to design networks, named Piping Net (PipNet) that automatically identify compliant versus non-compliant designs. To train PipNet, the methodology starts with networks already trained on common visual tasks, such as scene or object recognition, and fine-tune them on all five piping datasets introduced in previous works, including the one on oil residue handling and another on bilge system of ships [229] [232].

The idea of adopting transfer learning for this problem is based on the hypothesis that visual features learnt for more general vision tasks can be helpful in the context of engineering, and more specifically for a piping design in this work. As requirements on the amount of training samples for transfer learning are considerably less restrictive than training a CNN from scratch, this approach is consistent with current industrial practices where the amount of samples is considerably smaller than what is needed to allow effective training of deep neural networks from scratch.

The main contributions of this chapter are twofold: first, the concept of developing domain specific engineering knowledge from raw piping drawings in a visual manner; and second, the proposal of a framework named PipNet to achieve the automation of design appraisal via transfer learning on existing trained CNN models.

7.2 Visual Understanding of Design

The proposed solution to automated vision-based design appraisal is to use deep learning algorithms to learn from raw data and search for the best representations of the existing engineering knowledge. In the context of computer vision, this translates into feeding designs (in the form of images) and corresponding labels ($(1, 0)$ for compliant and $(0, 1)$ for non-compliant) to Convolutional Neural Networks (CNN) for learning. The first few convolutional layers of the CNN are expected to extract visual features and represent the existing components and the connectivity among them in a design. Next, based on the established understanding of the basic information of a design, subsequent layers including the fully connected ones are expected to disambiguate whether a design is compliant or not.

7.3 Methodology

The methodology mainly uses transfer learning to obtain a fine-tuned network using available piping designs.

7.3.1 Piping Net (PipNet)

The key component in this work is a CNN framework known as Piping Net (PipNet), which will be used for the automated visual analysis of engineering designs. PipNet does not necessitate a pre-defined architecture. The idea is to re-use learnt visual features in a trained CNN to enable transfer learning for engineering designs. This is achieved by fine-tuning of a pre-trained CNN using the available piping designs.

Abstracting from network architectures to a certain extent, the scientific premise here is that general visual features learnt through common tasks in computer vision such as scene and object recognition can be helpful to an engineering setting. This work tests this premise using experimental data in piping designs. In other words, this work tests whether a high-performance PipNet can be obtained by performing fine-tuning pre-trained CNNs on piping designs.

TABLE 7.1: CNN architectures investigated for piping design appraisal.

Destination	Source	Pre-trained on
PipNet	VGG16	Places365
	GoogleNet [234]	Places365
	ResNet152	Places365

7.3.2 Transfer Learning and Fine-Tuning

In general, there are two main approaches to perform transfer learning. The first approach keeps the weights of the entire network constant excluding the last layer. The aim is then to retrain the last layer for classification. The second approach allows the weights of the entire network to change, including the last layer for classification. Section 7.4 shows experimental results of both cases. Table 7.1 lists the representative CNN architectures and the models pre-trained on places using Places365 dataset [233] studied in this work.

7.3.3 Data: Insufficiency and Imbalance

The design of a piping systems on-board a ship is often separated into different sections, such as the ballast system which takes in and removes sea water to maintain the balance of a ship while cruising, and the bilge system which drains the dirty water. To learn the proper visual representations, conventional machine learning algorithms would expect a lot of compliant and non-compliant samples for each aspect. This may not always be possible, since each major ship construction project results in a limited amount of compliant designs and there are a limited number of ships or platforms built to date. In addition, organisations may not want to share designs due to confidentiality concerns. Compared with training a CNN from scratch, transfer learning is much less restrictive in terms of the amount of samples required, which makes it a good candidate approach to automate the visual analysis of piping designs.

In this work, each original dataset was randomly separated into two sets, one for training and one for testing, each using 50% of the data. This consisted of 64 positive (or compliant), 659 negative (or non-compliant) for dataset 0, and 127 positive, 767 negative samples for dataset 1 respectively. The same applies for the other three datasets of piping designs.

Given that there were more non-compliant than compliant designs, the frequency

of compliant samples being presented during the fine-tuning stage was increased to balance the presentation of each condition. For example, each compliant sample was shown approximately 9 times more for dataset 0 and 5 times more for dataset 1.

7.3.4 Flexibility in Representation

In order to evaluate the flexibility and the robustness of the PipNet towards representations, piping designs with components shown in the form of abbreviations instead of conventional symbols (like Figure 7.1 and 7.2) were used for transfer learning. For extension to more aspects of piping designs, it is important to study the ability of PipNet to generalise in terms of the form of representation for engineering knowledge.

Figure 7.1 and 7.2 show a negative (non-compliant) example of dataset 0 and 1 respectively, represented in the form of abbreviations. The reason for the design in Figure 7.1 to be non-compliant is that ball valves which allow bi-directional flow are used in the highlighted part of the design, this leads to a possibility of oil residue (from ORT) to flow into the oily bilge water holding tank (OBW). Similarly, for dataset 1, the reason of non-compliance of the design in Figure 7.2 is the connection between the two bilge wells, as highlighted with a red circle.

FIGURE 7.1: An example of a non-compliant design in dataset 0.

FIGURE 7.2: An example of a non-compliant design in dataset 1.

7.4 Experiments

For all the five experimental datasets (0 to 4) for performance evaluation, images were first resized to 224×224 . Then each design (in the form of an image) were provided to CNNs for training, together with the corresponding compliant / non-compliant label. Following fine-tuning, the designs in the testing set were used to evaluate the performance of PipNet obtained.

The fine-tuning was implemented in C++ using Caffe [235] on three systems with 32-48GB RAM in Ubuntu Linux. The Graphics Processing Unit (GPU)(s) on-board was two GeForce GTX 1080Ti with 3,584 cores, Tesla K20Xm with 2,688 cores and Titan X with 3,072 cores respectively. A learning of 5,000 iterations was completed in approximately two hours, with a learning rate of 0.001 and a batch size of 10.

The performance of the PipNet on the test set for the first 5,000 iterations of the fine-tuning (with a step of 1000 iterations) are shown in Figure 7.3-7.22 for both cases of fine-tuning the entire network and only the last layer of the network. Similarly, the performance on the case of symbols is shown in Figure 7.23-7.42. C indicates compliant, NC indicates non-compliant, number refers to the dataset involved. The x -axis refers to the number of iterations and the y -axis refers to the prediction accuracy of PipNet obtained for each class (compliant or non-compliant).

7.4.1 Abbreviation

The performance of the fine-tuned networks on the abbreviation version of drawing is presented in this section. For each dataset, there are four figures or pieces of key information involved. The first two figures show the performance of the fine-tuned model on compliant and non-compliant samples in testing set after each 1000 steps of fine-tuning, for the case of fine-tuning the entire network. The next two figures show the performance for the case of learning the last layer only.

As shown in the figures, it is very encouraging to see that a convergence is achieved after a while of fine-tuning for all the five piping datasets. This demonstrates the ability of the developed methodology to reason and generalise. Figure 7.3 to 7.6 show the performance of the fine-tuned networks for dataset 0.

The graphs can be seen in a pair of two, for the first two, it can be seen that the lines converge to y value of 1 after a while. This means that the performance of the networks on both the compliant and non-compliant samples converged to 95 – 100% after a while. This is very encouraging.

For the next two which correspond to the case of learning the last layer only, convergence happened but slower. This will be discussed in next section. In fact, similar performance was observed for the other four datasets, in following figures.

FIGURE 7.3: Set 0: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.4: Set 0: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.5: Set 0: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.6: Set 0: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

Figure 7.7 to 7.10 show the performance of the fine-tuned networks for dataset 1. Similar to the first case, the performance of the case of learning the last layer only converged slower.

FIGURE 7.7: Set 1: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.8: Set 1: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.9: Set 1: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.10: Set 1: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

Figure 7.11 to 7.14 show the performance of the fine-tuned networks for dataset 2. Similar performance and convergence were observed.

FIGURE 7.11: Set 2: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.12: Set 2: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.13: Set 2: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.14: Set 2: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

Figure 7.15 to 7.18 show the performance of the fine-tuned networks for dataset 3. In the first two figures, while the performance of the case of fine-tuning the entire network converged well, VGGNet in the case of learning the last layer only (next two figures) did not appear to converge. After 5000 iterations, the network got approximately 100% performance on the compliant test samples and 0% on the non-compliant ones, this is very far from expectations.

FIGURE 7.15: Set 3: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.16: Set 3: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.17: Set 3: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.18: Set 3: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

Figure 7.19 to 7.22 show the performance of the fine-tuned networks for dataset 4. Similar to dataset 3, while the networks converged for the case of fine-tuning the entire networks in general, VGGNet was unable to converge.

FIGURE 7.19: Set 4: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.20: Set 4: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.21: Set 4: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.22: Set 4: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

7.4.2 Symbol

The performance of the fine-tuned networks on the symbol version of drawing is presented in this section. Similar to the version of abbreviation earlier, a convergence was also observed. This serves as a further demonstration of the ability to reason and generalise of the developed method, across different architectures without too much of parameter tuning. Figure 7.23 to 7.26 show the performance of the fine-tuned networks for dataset 0. Similar to the case of abbreviation, the learning converged well in general.

FIGURE 7.23: Set 0: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.24: Set 0: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.25: Set 0: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.26: Set 0: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

Figure 7.27 to 7.30 show the performance of the fine-tuned networks for dataset 1. Similar performance as dataset 0 was observed.

FIGURE 7.27: Set 1: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.28: Set 1: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.29: Set 1: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.30: Set 1: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

Figure 7.31 to 7.34 show the performance of the fine-tuned networks for dataset 2. While the case of fine-tuning the entire network converged well, VGGNet did not perform well in the case of learning the last layer only.

FIGURE 7.31: Set 2: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.32: Set 2: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.33: Set 2: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.34: Set 2: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

Figure 7.35 to 7.38 show the performance of the fine-tuned networks for dataset 3. Similar issue of not converging well was observed for VGGNet and GoogleNet, in the case of learning the last layer only.

FIGURE 7.35: Set 3: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.36: Set 3: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.37: Set 3: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.38: Set 3: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

Figure 7.39 to 7.42 show the performance of the fine-tuned networks for dataset 4. It is interesting to note that the case of learning the last layer only outperformed the case of fine-tuning the entire network, using ResNet.

FIGURE 7.39: Set 4: The performance per 1000 iterations: Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.40: Set 4: The performance per 1000 iterations: Non-Compliant. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.41: Set 4: The performance per 1000 iterations: Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

FIGURE 7.42: Set 4: The performance per 1000 iterations: Non-Compliant, last layer. Blue: VGG, Yellow: GoogleNet, Green: ResNet. x -axis: Number of iteration in the process of fine-tuning, y -axis: The prediction accuracy on the testing set.

7.5 Discussion and Future Work

While being a challenging task, CNN based transfer learning has demonstrated a strong potential and the effectiveness in understanding of a piping design visually and determining the compliance. As shown in Section 7.4, PipNet has generalised well, yielding very high prediction accuracies (approximately 95 – 100%) on all testing sets. As demonstrated in the experimental results, visual features learnt in much more generic tasks of image recognition such as scene recognition were useful in an engineering setting.

For the case of dataset 4, where ResNet converged and performed better under the setting of the last layer only, it is believed that the dataset is more challenging and has prevented the other networks from learning properly. Before any further conclusion on this special case, further testing is needed.

As the prediction accuracy improved and stabilised faster, the training appeared to be better and smoother in the case of fine-tuning over the entire network, compared with re-training the last layer only. This is also aligned with the existing understanding that allowing fine-tuning over the entire network provides more and possibly closer to sufficient level of freedom for transfer learning.

In general, PipNet performed well regardless of the representations used for the components (symbols or abbreviations). The hypothesis of re-using learnt features for piping has been validated through experiments.

Besides, the performance of PipNet is quite close to HoC and linear SVM based methods developed previously [232]. It is worth noting that this comparison is

based on the assumption that the input (piping design) to the HoC and linear SVM approach is accurate and in vector form, not containing any errors that may occur during the vectorisation or symbol recognition, which is not a fair comparison for PipNet.

While computational resources are becoming better and cheaper with GPUs, the computational complexity has to be reasonable in cases where the access to high performance computing is not available. Transfer learning makes this possible by removing the need to train a network from scratch. This work shows that transfer learning is a promising approach for piping design appraisal, and probably other engineering problems with similar challenges.

While it is possible to generate random noise to non-compliant designs to get more samples, these samples are not useful legitimate piping designs. Given that obtaining large databases of designs is typically an unattainable goal in many industry cases, the proposed method in this work is a promising solution.

Following this work, it would be interesting to explore if similar approaches can generalise to other domains, such as the design of electrical circuits, which are also represented by graph data structures.

For further work, visualisation into deeper layers of PipNet can be used as a way to pinpoint the exact locations where designs become non-compliant, thus enabling engineers to quickly resolve compliance issues. In an industrial setting, this is of significant value as it allows the engineer to follow up and perform correction on parts that cause non-compliance.

Due to the limitation on access and computational resources, CNNs used in this work were pre-trained for scene recognition. It is necessary and also interesting to try fine-tuning CNNs trained on other visual tasks such as object recognition to further validate the hypothesis of re-using visual features for the analysis of engineering designs.

7.6 Summary

In summary, this chapter has presented a CNN and transfer learning based approach for the evaluation of piping designs of ships in order to determine whether they comply with international regulations. Raw piping designs in the form of images were used to fine-tune a pre-trained (on another visual task) CNN for transfer

learning, resulting in a Piping Net (PipNet). Generalising under different established CNN architectures, PipNet has demonstrated encouraging performance on all five piping datasets. Accuracies in the range of 95 – 100% have been observed for all five piping datasets, making industrial adoption very close.

Chapter 8

Discussion and Conclusion

8.1 Summary of contributions

In summary, referring back to the problem statement of this thesis:

This project aims to address the problem by developing an automated framework for the evaluation of a piping design. Given a P&ID as input, the developed framework is to tell if the layout of the design is compliant or non-compliant, according to a set of international rules or guidelines.

Based on the problem statement established, the methodology developed for the automated appraisal of a piping design works well for all five existing piping aspects or datasets. Contributions of this thesis consist of four major parts, as listed below. This also serves as one of the milestones in data-driven engineering design which unlocks the possibility of having a machine that can understand, interpret and analyse the compliance of an engineering design, powered by the state-of-the-art in machine and deep learning.

1. Starting from a mathematical representation based on Histogram of Connectivity (HoC), the method has demonstrated the effectiveness and the efficiency for this task, when coupled with Linear Support Vector Machines (SVM). By comparing each dimension of the weight vector solved in primal

form, the visualisation of the models learnt serves as a validation that the SVM learns what an engineer expects it to.

2. Beyond that, extending the mathematical representation established, a Pyramid of Connectivity Histogram (PoC) improves the performance of the learnt models in a higher dimensional space. PoC essentially extends HoC to the connectivity information beyond the immediate ones. The HoC at each level of resolution is concatenated and the resulting vector is used as the representation for the learning with a linear SVM in a similar way subsequently. Significant improvement of performance has been observed through the experiments carried out.
3. As one major issue for deployment in the industry, integrated learning has made concurrent learning of multiple aspects of piping design possible, in a common space known as Histogram of Connectivity in Common Space (HoCC). Given few aspects of piping design to be studied at one go and built upon HoC, the method starts with mapping HoC of each aspect of the piping design to a common space of higher dimensions, which considers all types of component in the piping designs. This new space of representation is known as HoCC. The learning of a linear SVM classifier takes place subsequently.
4. As discussed in chapter 7, beyond hand-crafted representations, the deep learning powered method enables the machine (or neural network) to identify the most effective representation and hence provides much more flexibility to the machine in the process of learning. Hence, the machine may see something that an engineer is not aware of such as possible sources of non-compliance. This can augment the ability of HoC, PoC and HoCC which learn mainly based on guidelines from engineers.

Combining the existing knowledge in the engineering world and giving the machine the flexibility to explore by itself, this project has unlocked the possibility of the automated evaluation of a piping design of a ship or offshore platform, through machine learning. There are two observations which are worth highlighting in this work, namely the influence of the key parameters such as the cost factor in SVM and the choices of classifiers on the performance. As the primary aim of this thesis is to develop a set of methodology for the machine to do an automated appraisal of a piping design, it is very encouraging to see that the mathematical representations

and fine-tuning algorithms worked well. The effects of the key parameters and the choices of classifiers are relatively minimal, this is aligned with the broader aim of this project, which is to develop a set of methodology which works well with a wide range of values for the key parameters and the choices of classifiers such as random forest. From another angle, this has demonstrated the robustness of the developed methodology, in terms of developers' choices of parameter values and classifiers according to actual situation in each case.

8.2 Discussion

Following a more detailed discussion on possible directions for further work in each chapter, this section provides a discussion on further work at the level of a project.

8.2.1 Capturing existing knowledge via HoC, PoC and HoCC

The developed methodology of this thesis has successfully combined the existing engineering knowledge and the learning by samples, which is a close replication of an engineer's experience of learning. However, further research could also lead to a better way to integrate the existing knowledge and the learning by samples. In particular, while this thesis has demonstrated that the incorporation of existing domain knowledge through manually designed representations is possible and promising, it is believed that there is a huge room for improvement. The search for a better, more flexible and suitable form to represent the existing domain knowledge would be an interesting and essential direction to look into for further improvement.

It may also be interesting to look into an integration of PoC with HoCC. While this would lead to a much higher-dimensional space which may increase the computational cost significantly, performance improvement is also expected. However, an issue of whether this improvement over performance is actually needed at the cost of slower computation for an industrial deployment needs to be taken into consideration too.

8.2.2 Learning by self-exploration: Transferring common visual features

Next, it has been highlighted earlier that in addition to the existing domain knowledge which has been established by engineers over the years, the ability to do self-exploration with examples would greatly enhance the learning experience of a machine, towards the final aim of the automated evaluation of a piping design on-board a ship. The search for an effective way or customisation to allow the developed methodology to perform reasoning by itself would be an essential direction to look into.

Powered by the idea of transfer learning [208], it has been demonstrated in chapter 7 that visual features acquired from the learning about place recognition can be reused for this domain of the piping design of ships. However, for uses in heavy engineering systems such as ships, there is also an emphasis on why and how beyond a decision. The main concern here would be the issue of interpretability of the models learnt, there needs to be a way to illustrate or explain why the fine-tuned deep convolutional neural network arrives at such a decision. Recent breakthroughs in making deep learning more interpretable [197] may be of interest regarding this question.

8.2.3 Industrial adoption

While it has been demonstrated in this thesis that machine learning based automated appraisal of a piping design on-board ship is feasible, further study and development is needed before an industrial adoption. From the angle of research and development, main questions in the part of research have been answered through the four primary components of this work. Based on the current version of the methodology developed, directions which may lead to important breakthroughs in future have also been highlighted.

From the perspective of development or engineering, work to incorporate models and algorithms developed into a software for commercial deployment is needed. While it is foreseen that the engineering aspect of this work leading to a commercial software does not contain too much significant scientific challenges, it is of equal

importance to a successful deployment of achievements of this research to actual use.

Next, there is also a need to highlight that having learnt models only on five piping aspects would not be sufficient for an industrial adoption. A straightforward but also essential task would be to collect minimally sufficient samples which are expected to be at the scale of hundreds to thousands so that the machine is completely trained on all aspects of piping design on-board ships.

Lastly, it is also important to bring up that this thesis focusses on the design rules that require interpretation, such as those highlighted through the five datasets. As discussed earlier, there are also explicit rules that can be incorporated into an expert system in the form of a rule base. While this is not the primary aim of the thesis, relatively straightforward efforts are also needed to incorporate these rules into an expert system, which will play a part in the ultimate aim of the automated evaluation of a piping design.

Following the current practice on the design appraisal in the industry, which is a completely manual process of evaluating a piping design, it is foreseen that the methodology developed in this work can bring huge increase in both the efficiency and the effectiveness to the current practice. While it is hard to develop a theoretical guarantee on decisions made by the classifiers, including those that are predicted as compliant, these classifiers could be deployed as an automated tool to assist engineers and naval architects during the early stage of the design of a ship or offshore platform. The tool could try to eliminate as much non-compliance in the piping design as possible, and hence ensure that the final design submitted for review by experts are as close to compliance as possible.

In general, the performance of learnt classifiers in this work has been observed to be very close to the range of 80 – 100%. This is in-line with the goal of the project of doing an early warning on possible non-compliance during the early stage of designing the piping system of a ship or offshore platform. While the system may not be able to identify all non-compliance in a piping design, it is expected to make a design as compliant as possible when it is sent to experts for review and final approval.

8.2.4 Automated generation of compliant designs for industrial use

Following a series of achievements in this work, a natural question to ask would be if it is possible to generate a compliant piping design for an immediate use in the process of designing a new ship, in an automated manner. From the perspective of the industry, this would lead to a great increase in both the effectiveness and the efficiency in the engineering of a new ship. Theoretically, by reusing the learnt SVM and CNN models obtained so far, it is definitely possible to reverse engineer and recreate some compliant designs.

An immediate possibility of doing an automated generation of a compliant piping design would be a nearest neighbour and linear SVM classifiers based approach. Given a set of specifications engineers or construction service providers desire, the nearest neighbour which corresponds to the closest compliant design the machine has seen so far in a parameter space pre-defined could be found. Starting from this compliant design, a systematic evolution of the design could be performed according to provided specifications from the engineers or construction service providers. At each step, the new design could be fed into the linear SVM classifiers for a check on compliance. If a non-compliant design is detected, the latest change is reversed and the exploration moves back to the previous step. This iterates until a suitable design which matches the provided specifications has been determined.

However, a few questions and challenges exist in this direction of the research. The first question would be on the significance of the gain over this automated generation. As the aim of doing an automated generation of a compliant design requires much more imagination, compared with the task of evaluating if a piping design is compliant, the difference in the aim of learning may have limited the ability of the machine to generalise and imagine. From another angle of consideration, as the factor of performance optimisation of a piping design is not considered during the learning about compliance, it would be challenging for the machine to imagine by itself on what a more optimal design would be, hence designs generated by the learnt models in this work might not be that optimal as expected or very similar to those compliant designs it has seen before. Hence, the question on the significance of the actual gain over the automated design synthesis arises.

The adoption of a risk-based methodology may be a possible solution to this. By trying to strike a balance in ensuring the compliance and providing sufficient freedom to the machine for self-exploration, the aim translates to achieving an equilibrium for the risk, at the cost of having a good chance of identifying a better design, which could be beyond an experienced engineer's imagination. Along the direction of deep learning, the recent breakthrough in the research of Generative Adversarial Networks [236] may be able to bring the learnt CNN models to another level.

Appendix A

Rules Applicable from the Original Text

Aspect 0: Regulation 12, Annex I, MARPOL [237]

The oil residue (sludge) tank(s) shall have no discharge connections to the bilge system, oily bilge water holding tank(s), tank top or oily water separators except that:

1. the sludge tank discharge piping and bilge-water piping may be connected to a common piping leading to the standard discharge connection referred to in regulation 13; the connection of both systems to the possible common piping leading to the standard discharge connection referred to in regulation 13 shall not allow for the transfer of sludge to the bilge system;

Aspect 1: Bilge system [222]

4.1 General

4.1.1 The bilge drainage arrangements in the machinery space are to comply with Pt 5, Ch 13, 3.1 General, except that the arrangements are to be such that any water which may enter this compartment can be pumped out through at least two bilge suctions when the ship is on an even keel, and is either upright or has a list of not more than 5°. One of these suctions is to be a branch bilge suction, i.e.

a suction connected to the main bilge line, and the other is to be a direct bilge suction, i.e. a suction led direct to an independent power pump. Examples of the necessary arrangements are detailed in Pt 5, Ch 13, 4.2 Machinery space with double bottom and Pt 5, Ch 13, 4.3 Machinery space without double bottom.

4.6 Machinery space - Emergency bilge drainage

4.6.1 In addition to the bilge suctions detailed in Pt 5, Ch 13, 4.1 General, an emergency bilge suction is to be provided in each main machinery space.

7.2 Prevention of communication between compartments

7.2.1 The arrangement of valves, cocks and their connections is to be such as to prevent the possibility of one watertight compartment being placed in communication with another, or of dry cargo spaces, machinery spaces or other dry compartments being placed in communication with the sea or with tanks. For this purpose, screw-down non-return valves are to be provided in the following fittings:

- Bilge valve distribution chests.
- Bilge suction hose connections, whether fitted direct to the pump or on the main bilge line.
- Direct bilge suctions and bilge pump connections to main bilge line.

Aspect 2: Fuel oil transfer system [222]

Chapter 13: 3.5 Fore and after peaks

3.5.1 Fuel oil, lubrication oil and other flammable liquids are not to be carried out in forepeak tanks.

Chapter 14

1. 4.3 Relief valves on pumps: 4.3.1 All pumps which are capable of developing a pressure exceeding the design pressure of the system are to be provided with relief valves. Each relief valve is to be in closed circuit, i.e. arranged to discharge back to the suction side of the pump and to effectively limit the pump discharge pressure to the design pressure of the system.
2. 4.4 Pump connection: 4.4.1 Valves or cocks are to be interposed between the pumps and the suction and discharge pipes, in order that any pump may be shut off for opening up and overhauling.

3. 4.7 Valves and cocks: 4.7.3 Every fuel oil suction pipe from a double bottom tank is to be fitted with a valve or cock.
4. 4.8 Valves on deep tanks and their control arrangements: 4.8.1 Every fuel oil suction pipe from a storage, settling and daily service tank situated above the double bottom, and every fuel oil levelling pipe within the boiler room or engine room, is to be fitted with a valve or cock secured to the tank.
5. 4.11 Filling arrangements: 4.11.2 Provision is to be made against over-pressure in the filling pipelines, and any relief valve fitted for this purpose is to discharge to an overflow tank or other safe position.
6. 4.18 Fuel oil service tanks: 4.18.2: Two fuel oil service tanks, for each type of fuel used on board, necessary for propulsion and generator systems, are to be provided. Each tank is to have a capacity for at least eight hours' operation, at sea, at maximum continuous rating of the propulsion plant and/or generating plant associated with that tank.
7. 12.4.6 Fuel oil tanks: 12.4.6 Means are to be provided to eliminate the possibility of overflow from fuel oil service tanks into the machinery space and to safeguard against overflow of oil from fuel oil service tanks through the air pipe. See Pt 5, Ch 13 Ship Piping Systems regarding the termination of air pipes.

Aspect 3: Ballast system [222]

6.2 General service pumps

6.2.1 The bilge pumping units, or pumps, required by Pt 5, Ch 13, 6.1 Number of pumps may also be used for ballast, fire or general service duties of an intermittent nature, but they are to be immediately available for bilge duty when required, see also SOLAS 1974 as amended Reg. II-2/C, Regulation 10 - Fire fighting, as applicable.

11.1 Stand-by arrangements for ballast pumping

11.1.1 Where ballasting/de-ballasting is required for ship operation or trading purposes stand-by ballast pumping arrangements are to be provided, see also Pt 5, Ch 13, 6.2 General service pumps 6.2.1 and Pt 5, Ch 15, 2.4 Drainage of ballast tanks and void spaces within the range of the cargo tanks 2.4.4.

Aspect 4: Isolation of bilge system [222]7.3 Isolation of bilge system

7.3.1 Bilge pipes which are required for draining cargo or machinery spaces are to be entirely distinct from sea inlet pipes or from pipes which may be used for filling or emptying spaces where water or oil is carried. This does not, however, exclude a bilge ejection connection, a connecting pipe from a pump to its suction valve chest, or a deep tank suction pipe suitably connected through a changeover device to a bilge, ballast or oil line.

Appendix B

Legend for Symbols in Piping Designs

A legend for the symbols used in piping designs in this work is provided in Table [B.1](#).

TABLE B.1: List of symbols used in the piping designs.

Description	Symbol
Pump
Pump for General Service
Pump for Ballast
Screw-Down Non-Return Valve
Globe Valve
Ball Valve
Butterfly Valve

List of Author's Awards, Patents, and Publications¹

Journal Articles

- **W. C. Tan**, I. Chen, S. J. Pan and H. K. Tan, "Analysis With Histogram of Connectivity: For Automated Evaluation of Piping Layout," in IEEE Transactions on Automation Science and Engineering (**T-ASE**), vol. 15, no. 1, pp. 381-392, Jan. 2018.

Conference Proceedings

- **W.C. Tan**, I.M. Chen, D. Pantazis and S.J. Pan, "PipNet: Vision-based Analysis of Piping Design in Maritime Engineering," 2018 14th IEEE Conference on Automation Science and Engineering (**CASE**), Munich, 2018.
- **W.C. Tan**, P.C. Goh, I.M. Chen and K.H. Chua, "Learning with Corrosion Feature: For Automated Quantitative Risk Analysis of Corrosion Mechanism," 2018 14th IEEE Conference on Automation Science and Engineering (**CASE**), Munich, 2018.
- **W.C. Tan**, C.Y. Weng, Y. Zhou, K.H. Chua and I.M. Chen, "Historical Data is Useful for Navigation Planning: Data Driven Route Generation for Autonomous Ship," In International Conference on Robotics and Automation (**ICRA**), Brisbane, Australia, 2018.

¹The superscript * indicates joint first authors

- **W.C. Tan**, I.M. Chen, P.C. Goh and H.K. Tan, "Software for Automated Visual Assessment of Coating Damage," In Offshore Technology Conference (**OTC Asia**), Kuala Lumpur, Malaysia, 2018.
- **W. C. Tan**, I.M. Chen and H. K. Tan, "Multiple binary classifiers to analyse decision of non-compliance: For automated evaluation of piping layout," 2017 13th IEEE Conference on Automation Science and Engineering (**CASE**), Xi'an, 2017, pp. 80-85.
- **W. C. Tan**, P. C. Goh, A. Causo, I.M. Chen and H. K. Tan, "Automated vision based detection of blistering on metal surface: For robot," 2017 13th IEEE Conference on Automation Science and Engineering (**CASE**), Xi'an, 2017, pp. 74-79.
- **Wei Chian Tan**, I-Ming Chen, Sinno Jialin Pan and Hoon Kiang Tan, "Automated design evaluation on layout of Piping and Instrumentation Diagram using Histogram of Connectivity," 2016 IEEE International Conference on Automation Science and Engineering (**CASE**), Fort Worth, TX, 2016, pp. 1295-1300.
- **Wei Chian Tan**, I-Ming Chen and Hoon Kiang Tan, "Automated identification of components in raster piping and instrumentation diagram with minimal pre-processing," 2016 IEEE International Conference on Automation Science and Engineering (**CASE**), Fort Worth, TX, 2016, pp. 1301-1306.

Bibliography

- [1] S. A. Khan and S. Rehman, “Iterative non-deterministic algorithms in on-shore wind farm design: A brief survey,” *Renewable and Sustainable Energy Reviews*, vol. 19, pp. 370 – 384, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1364032112006521>
- [2] F. Corno and F. Razzak, “Intelligent energy optimization for user intelligible goals in smart home environments,” *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 2128–2135, Dec 2012.
- [3] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, “Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,” *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s00366-011-0241-y>
- [4] —, “Mixed variable structural optimization using firefly algorithm,” *Computers & Structures*, vol. 89, no. 2324, pp. 2325 – 2336, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045794911002185>
- [5] A. Baykasoglu, “Design optimization with chaos embedded great deluge algorithm,” *Applied Soft Computing*, vol. 12, no. 3, pp. 1055 – 1067, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494611004601>
- [6] K. B. Mawussi and L. Tapie, “A knowledge base model for complex forging die machining,” *Computers & Industrial Engineering*, vol. 61, no. 1, pp. 84 – 97, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0360835211000714>
- [7] Y.-Y. Hsu, P.-H. Tai, M.-W. Wang, and W.-C. Chen, “A knowledge-based engineering system for assembly sequence planning,” *The International*

- Journal of Advanced Manufacturing Technology*, vol. 55, no. 5, pp. 763–782, 2011. [Online]. Available: <http://dx.doi.org/10.1007/s00170-010-3093-5>
- [8] G. Moslemipour, T. S. Lee, and D. Rilling, “A review of intelligent approaches for designing dynamic and robust layouts in flexible manufacturing systems,” *The International Journal of Advanced Manufacturing Technology*, vol. 60, no. 1, pp. 11–27, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s00170-011-3614-x>
- [9] A. R. Yildiz, “A new hybrid artificial bee colony algorithm for robust optimal design and manufacturing,” *Applied Soft Computing*, vol. 13, no. 5, pp. 2906 – 2912, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494612001962>
- [10] H.-S. Mok, C.-H. Kim, and C.-B. Kim, “Automation of mold designs with the reuse of standard parts,” *Expert Systems with Applications*, vol. 38, no. 10, pp. 12 537 – 12 547, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417411005562>
- [11] J. Hiller and H. Lipson, “Automatic design and manufacture of soft robots,” *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 457–466, April 2012.
- [12] G. Francesca, M. Brambilla, A. Brutschy, L. Garattoni, R. Miletitch, G. Podevijn, A. Reina, T. Soleymani, M. Salvaro, C. Pincioli, V. Trianni, and M. Birattari, *An Experiment in Automatic Design of Robot Swarms*. Cham: Springer International Publishing, 2014, pp. 25–37. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-09952-1_3
- [13] A. A. Fahmy, M. Kalyoncu, and M. Castellani, “Automatic design of control systems for robot manipulators using the bees algorithm,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2011. [Online]. Available: <http://pii.sagepub.com/content/early/2011/11/06/0959651811425312.abstract>
- [14] S. Doncieux and J.-B. Mouret, “Beyond black-box optimization: a review of selective pressures for evolutionary robotics,” *Evolutionary Intelligence*, vol. 7, no. 2, pp. 71–93, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s12065-014-0110-x>

- [15] H. Komoto and T. Tomiyama, “A framework for computer-aided conceptual design and its application to system architecting of mechatronics products,” *Computer-Aided Design*, vol. 44, no. 10, pp. 931 – 946, 2012, fundamentals of Next Generation CAD/E Systems. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0010448512000401>
- [16] J. ju Cui and D. yu Wang, “Application of knowledge-based engineering in ship structural design and optimization,” *Ocean Engineering*, vol. 72, pp. 124 – 139, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S002980181300259X>
- [17] Y.-H. Wu and H.-J. Shaw, “Document based knowledge base engineering method for ship basic design,” *Ocean Engineering*, vol. 38, no. 13, pp. 1508 – 1521, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0029801811001594>
- [18] H. Yang, J. Chen, N. Ma, and D. Wang, “Implementation of knowledge-based engineering methodology in ship structural design,” *Computer-Aided Design*, vol. 44, no. 3, pp. 196 – 202, 2012, applications in Ship and Floating Structure Design and Analysis. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0010448511001540>
- [19] Y. Xue, D. Clelland, B. Lee, and D. Han, “Automatic simulation of ship navigation,” *Ocean Engineering*, vol. 38, no. 1718, pp. 2290 – 2305, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0029801811002514>
- [20] R. Akhavian and A. H. Behzadan, “Knowledge-based simulation modeling of construction fleet operations using multimodal-process data mining,” *Journal of Construction Engineering and Management*, vol. 139, no. 11, p. 04013021, 2013. [Online]. Available: [http://dx.doi.org/10.1061/\(ASCE\)CO.1943-7862.0000775](http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000775)
- [21] G. Pellegrino, F. Cupertino, and C. Gerada, “Automatic design of synchronous reluctance motors focusing on barrier shape optimization,” *IEEE Transactions on Industry Applications*, vol. 51, no. 2, pp. 1465–1474, March 2015.

- [22] C. Svärd and M. Nyberg, “Automated design of an fdi system for the wind turbine benchmark,” *J. Control Sci. Eng.*, vol. 2012, pp. 1:1–1:1, Jan. 2012. [Online]. Available: <http://dx.doi.org/10.1155/2012/989873>
- [23] C. A. Ventura, P. A. Jacobs, A. S. Rowlands, P. Petrie-Repar, and E. Sauret, “Preliminary design and performance estimation of radial inflow turbines: An automated approach,” *Journal of Fluids Engineering*, vol. 134, no. 3, pp. 031102–031102, Mar. 2012. [Online]. Available: <http://dx.doi.org/10.1115/1.4006174>
- [24] Q. Zhu, X. Xu, J. Wang, and F. Xiao, “Development of dynamic simplified thermal models of active pipe-embedded building envelopes using genetic algorithm,” *International Journal of Thermal Sciences*, vol. 76, pp. 258 – 272, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1290072913002354>
- [25] C. Wang and Q. Liu, “Projection and geodesic-based pipe routing algorithm,” *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 3, pp. 641–645, July 2011.
- [26] M. Cabaleiro, B. Riveiro, P. Arias, J. Caamao, and J. Viln, “Automatic 3d modelling of metal frame connections from lidar data for structural engineering purposes,” *{ISPRS} Journal of Photogrammetry and Remote Sensing*, vol. 96, pp. 47 – 56, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0924271614001804>
- [27] D. C. Nolan, C. M. Tierney, C. G. Armstrong, T. T. Robinson, and J. E. Makem, “Automatic dimensional reduction and meshing of stiffened thin-wall structures,” *Engineering with Computers*, vol. 30, no. 4, pp. 689–701, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s00366-013-0317-y>
- [28] M. Mehrabi, M. Sharifpur, and J. Meyer, “Application of the fcm-based neuro-fuzzy inference system and genetic algorithm-polynomial neural network approaches to modelling the thermal conductivity of aluminawater nanofluids,” *International Communications in Heat and Mass Transfer*, vol. 39, no. 7, pp. 971 – 977, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0735193312001169>

- [29] S. Qvist and E. Greenspan, “The {ADOPT} code for automated fast reactor core design,” *Annals of Nuclear Energy*, vol. 71, pp. 23 – 36, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S030645491400125X>
- [30] T. Mareda, L. Gaudard, and F. Romerio, “A parametric genetic algorithm approach to assess complementary options of large scale windsolar coupling,” *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 2, pp. 260–272, April 2017.
- [31] X. Guo, S. Liu, M. Zhou, and G. Tian, “Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and petri nets,” *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2435–2446, Nov 2016.
- [32] N. A. Okaeme and P. Zanchetta, “Hybrid bacterial foraging optimization strategy for automated experimental control design in electrical drives,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 668–678, May 2013.
- [33] G. Y. Sizov, P. Zhang, D. M. Ionel, N. A. O. Demerdash, and M. Rosu, “Automated bi-objective design optimization of multi-mw direct-drive pm machines using ce-fea and differential evolution,” in *2011 IEEE Energy Conversion Congress and Exposition*, Sept 2011, pp. 3672–3678.
- [34] —, “Automated multi-objective design optimization of pm ac machines using computationally efficient fea and differential evolution,” *IEEE Transactions on Industry Applications*, vol. 49, no. 5, pp. 2086–2096, Sept 2013.
- [35] B. Houska, H. J. Ferreau, and M. Diehl, “Acado toolkit an open-source framework for automatic control and dynamic optimization,” *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011. [Online]. Available: <http://dx.doi.org/10.1002/oca.939>
- [36] V. Machairas, A. Tsangrassoulis, and K. Axarli, “Algorithms for optimization of building design: A review,” *Renewable and Sustainable Energy Reviews*, vol. 31, pp. 101 – 112, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1364032113007855>

- [37] S. Zhang, J. Teizer, J.-K. Lee, C. M. Eastman, and M. Venugopal, "Building information modeling (bim) and safety: Automatic safety checking of construction models and schedules," *Automation in Construction*, vol. 29, pp. 183 – 195, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0926580512000799>
- [38] J. Melzner, S. Zhang, J. Teizer, and H.-J. Bargstdt, "A case study on automated safety compliance checking to assist fall protection design and planning in building information models," *Construction Management and Economics*, vol. 31, no. 6, pp. 661–674, 2013. [Online]. Available: <http://dx.doi.org/10.1080/01446193.2013.780662>
- [39] S. Zhang, F. Boukamp, and J. Teizer, "Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (jha)," *Automation in Construction*, vol. 52, pp. 29 – 41, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0926580515000242>
- [40] L. Jiang and R. M. Leicht, "Automated rule-based constructability checking: Case study of formwork," *Journal of Management in Engineering*, vol. 31, no. 1, p. A4014004, 2015. [Online]. Available: [http://dx.doi.org/10.1061/\(ASCE\)ME.1943-5479.0000304](http://dx.doi.org/10.1061/(ASCE)ME.1943-5479.0000304)
- [41] J.-K. Lee, J. Lee, Y. suk Jeong, H. Sheward, P. Sanguinetti, S. Abdelmohsen, and C. M. Eastman, "Development of space database for automated building design review systems," *Automation in Construction*, vol. 24, pp. 203 – 212, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0926580512000441>
- [42] H. Kim, A. Stumpf, and W. Kim, "Analysis of an energy efficient building design through data mining approach," *Automation in Construction*, vol. 20, no. 1, pp. 37 – 43, 2011, global convergence in construction. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0926580510001044>
- [43] A.-T. Nguyen, S. Reiter, and P. Rigo, "A review on simulation-based optimization methods applied to building performance analysis," *Applied Energy*, vol. 113, pp. 1043 – 1058, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0306261913007058>

- [44] B. B. Ekici and U. T. Aksoy, "Prediction of building energy needs in early stage of design by using {ANFIS}," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5352 – 5358, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417410011656>
- [45] A. Stadel, J. Eboli, A. Ryberg, J. Mitchell, and S. Spatari, "Intelligent sustainable design: Integration of carbon accounting and building information modeling," *Journal of Professional Issues in Engineering Education and Practice*, vol. 137, no. 2, pp. 51–54, 2011. [Online]. Available: [http://dx.doi.org/10.1061/\(ASCE\)EI.1943-5541.0000053](http://dx.doi.org/10.1061/(ASCE)EI.1943-5541.0000053)
- [46] A. Di Nardo, M. Di Natale, G. F. Santonastaso, V. G. Tzatchkov, and V. H. Alcocer-Yamanaka, "Water network sectorization based on a genetic algorithm and minimum dissipated power paths," *Water Science and Technology: Water Supply*, vol. 13, no. 4, pp. 951–957, 2013. [Online]. Available: <http://ws.iwaponline.com/content/13/4/951>
- [47] N. Ghajarnia, O. B. Haddad, and M. A. Mario, "Performance of a novel hybrid algorithm in the design of water networks," *Proceedings of the Institution of Civil Engineers - Water Management*, vol. 164, no. 4, pp. 173–191, 2011. [Online]. Available: <http://www.icevirtuallibrary.com/doi/abs/10.1680/wama.1000028>
- [48] A. Di Nardo, M. Di Natale, G. F. Santonastaso, and S. Venticinque, "An automated tool for smart water network partitioning," *Water Resources Management*, vol. 27, no. 13, pp. 4493–4508, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s11269-013-0421-1>
- [49] E. Creaco and M. Franchini, "Fast network multi-objective design algorithm combined with an a posteriori procedure for reliability evaluation under various operational scenarios," *Urban Water Journal*, vol. 9, no. 6, pp. 385–399, 2012. [Online]. Available: <http://dx.doi.org/10.1080/1573062X.2012.690432>
- [50] S. Bureerat and K. Sriworam, "Simultaneous topology and sizing optimization of a water distribution network using a hybrid multiobjective evolutionary algorithm," *Applied Soft Computing*, vol. 13, no. 8, pp. 3693 – 3702, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494613001154>

- [51] W. Bi, G. Dandy, and H. Maier, “Improved genetic algorithm optimization of water distribution system design by incorporating domain knowledge,” *Environmental Modelling & Software*, vol. 69, pp. 370 – 381, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1364815214002631>
- [52] C. Siew and T. T. Tanyimboh, “Penalty-free feasibility boundary convergent multi-objective evolutionary algorithm for the optimization of water distribution systems,” *Water Resources Management*, vol. 26, no. 15, pp. 4485–4507, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s11269-012-0158-2>
- [53] F. Zheng, A. R. Simpson, and A. C. Zecchin, “A combined nlp-differential evolution algorithm approach for the optimization of looped water distribution systems,” *Water Resources Research*, vol. 47, no. 8, pp. n/a–n/a, 2011, w08531. [Online]. Available: <http://dx.doi.org/10.1029/2011WR010394>
- [54] F. Zheng, A. C. Zecchin, and A. R. Simpson, “Self-adaptive differential evolution algorithm applied to water distribution system optimization,” *Journal of Computing in Civil Engineering*, vol. 27, no. 2, pp. 148–158, 2013. [Online]. Available: [http://dx.doi.org/10.1061/\(ASCE\)CP.1943-5487.0000208](http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000208)
- [55] F. Zheng, A. R. Simpson, and A. C. Zecchin, “Coupled binary linear programming differential evolution algorithm approach for water distribution system optimization,” *Journal of Water Resources Planning and Management*, vol. 140, no. 5, pp. 585–597, 2014. [Online]. Available: [http://dx.doi.org/10.1061/\(ASCE\)WR.1943-5452.0000367](http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000367)
- [56] E. Creaco and G. Pezzinga, “Multiobjective optimization of pipe replacements and control valve installations for leakage attenuation in water distribution networks,” *Journal of Water Resources Planning and Management*, vol. 141, no. 3, p. 04014059, 2015. [Online]. Available: [http://dx.doi.org/10.1061/\(ASCE\)WR.1943-5452.0000458](http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000458)
- [57] C. R. Suribabu, “Heuristic-based pipe dimensioning model for water distribution networks,” *Journal of Pipeline Systems Engineering and*

- Practice*, vol. 3, no. 4, pp. 115–124, 2012. [Online]. Available: [http://dx.doi.org/10.1061/\(ASCE\)PS.1949-1204.0000104](http://dx.doi.org/10.1061/(ASCE)PS.1949-1204.0000104)
- [58] A. Haghighi and A. E. Bakhshipour, “Optimization of sewer networks using an adaptive genetic algorithm,” *Water Resources Management*, vol. 26, no. 12, pp. 3441–3456, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s11269-012-0084-3>
- [59] Q. Liu and C. Wang, “Multi-terminal pipe routing by steiner minimal tree and particle swarm optimisation,” *Enterprise Information Systems*, vol. 6, no. 3, pp. 315–327, 2012. [Online]. Available: <http://dx.doi.org/10.1080/17517575.2011.594910>
- [60] T. Krueger, T. Page, K. Hubacek, L. Smith, and K. Hiscock, “The role of expert opinion in environmental modelling,” *Environmental Modelling & Software*, vol. 36, pp. 4 – 18, 2012, thematic issue on Expert Opinion in Environmental Modelling and Management. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S136481521200028X>
- [61] N. V. Hernandez, G. O. Kremer, L. Schmidt, and P. A. Herrera, “Development of an expert system to aid engineers in the selection of design for environment methods and tools,” *Expert Systems with Applications*, vol. 39, no. 10, pp. 9543 – 9553, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417412003612>
- [62] A. H. Alavi and A. H. Gandomi, “A robust data mining approach for formulation of geotechnical engineering systems,” *Engineering Computations*, vol. 28, no. 3, pp. 242–274, 2011. [Online]. Available: <http://dx.doi.org/10.1108/026444401111118132>
- [63] R. Alur, T. A. Henzinger, and M. Y. Vardi, “Theory in practice for system design and verification,” *ACM SIGLOG News*, vol. 2, no. 1, pp. 46–51, Jan. 2015. [Online]. Available: <http://doi.acm.org/10.1145/2728816.2728827>
- [64] T. Yoshimura and E. S. Kuh, “Efficient algorithms for channel routing,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 1, no. 1, pp. 25–35, January 1982.

- [65] I. Kuon and J. Rose, "Exploring area and delay tradeoffs in fpgas with architecture and automated transistor design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 1, pp. 71–84, Jan 2011.
- [66] R. Vural and T. Yildirim, "Analog circuit sizing via swarm intelligence," *{AEU} - International Journal of Electronics and Communications*, vol. 66, no. 9, pp. 732 – 740, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1434841112000052>
- [67] S. I. Sulaiman, T. K. A. Rahman, I. Musirin, S. Shaari, and K. Sopian, "An intelligent method for sizing optimization in grid-connected photovoltaic system," *Solar Energy*, vol. 86, no. 7, pp. 2067 – 2082, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0038092X12001478>
- [68] I. Meedeniya, B. Buhnova, A. Aleti, and L. Grunske, "Reliability-driven deployment optimization for embedded systems," *Journal of Systems and Software*, vol. 84, no. 5, pp. 835 – 846, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121211000069>
- [69] H. Tian, H. Zhang, Q. Ma, Z. Xiao, and M. D. F. Wong, "A polynomial time triple patterning algorithm for cell based row-structure layout," in *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2012, pp. 57–64.
- [70] I. G. Harris, "Extracting design information from natural language specifications," in *Proceedings of the 49th Annual Design Automation Conference*, ser. DAC '12. New York, NY, USA: ACM, 2012, pp. 1256–1257. [Online]. Available: <http://doi.acm.org/10.1145/2228360.2228591>
- [71] A. Jafari, E. Bijami, H. R. Bana, and S. Sadri, "A design automation system for {CMOS} analog integrated circuits using new hybrid shuffled frog leaping algorithm," *Microelectronics Journal*, vol. 43, no. 11, pp. 908 – 915, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0026269212001206>
- [72] F. G. Zadegan, U. Ingelsson, G. Carlsson, and E. Larsson, "Design automation for ieee p1687," in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–6.

- [73] Q. Zhu, K. Vaidyanathan, O. Shacham, M. Horowitz, L. Pileggi, and F. Franchetti, “Design automation framework for application-specific logic-in-memory blocks,” in *2012 IEEE 23rd International Conference on Application-Specific Systems, Architectures and Processors*, July 2012, pp. 125–132.
- [74] T. B. Soeiro, J. Mhlethaler, J. Linnr, P. Ranstad, and J. W. Kolar, “Automated design of a high-power high-frequency lcc resonant converter for electrostatic precipitators,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 4805–4819, Nov 2013.
- [75] M. A. Bamakhrama, J. T. Zhai, H. Nikolov, and T. Stefanov, “A methodology for automated design of hard-real-time embedded streaming systems,” in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE ’12. San Jose, CA, USA: EDA Consortium, 2012, pp. 941–946. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2492708.2492944>
- [76] A. Trentin, P. Zanchetta, J. Clare, and P. Wheeler, “Automated optimal design of input filters for direct ac/ac matrix converters,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 7, pp. 2811–2823, July 2012.
- [77] Y. Wang, K.-M. Yu, C. C. Wang, and Y. Zhang, “Automatic design of conformal cooling circuits for rapid tooling,” *Computer-Aided Design*, vol. 43, no. 8, pp. 1001 – 1010, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0010448511001060>
- [78] R. Martins, N. Loureno, S. Rodrigues, J. Guilherme, and N. Horta, “Aida: Automated analog ic design flow from circuit level to layout,” in *Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), 2012 International Conference on*, Sept 2012, pp. 29–32.
- [79] S. Miryala, M. Montazeri, A. Calimera, E. Macii, and M. Poncino, “A verilog-a model for reconfigurable logic gates based on graphene pn-junctions,” in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE ’13. San Jose, CA, USA: EDA Consortium, 2013, pp. 877–880. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2485288.2485499>
- [80] F. Koushanfar, S. Fazzari, C. McCants, W. Bryson, M. Sale, P. Song, and M. Potkonjak, “Can eda combat the rise of electronic counterfeiting?” in

- Proceedings of the 49th Annual Design Automation Conference*, ser. DAC '12. New York, NY, USA: ACM, 2012, pp. 133–138. [Online]. Available: <http://doi.acm.org/10.1145/2228360.2228386>
- [81] R. Wille, R. Drechsler, C. Osewold, and A. Garcia-Ortiz, “Automatic design of low-power encoders using reversible circuit synthesis,” in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '12. San Jose, CA, USA: EDA Consortium, 2012, pp. 1036–1041. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2492708.2492966>
- [82] M. Owaida, N. Bellas, K. Daloukas, and C. D. Antonopoulos, “Synthesis of platform architectures from opencl programs,” in *Field-Programmable Custom Computing Machines (FCCM), 2011 IEEE 19th Annual International Symposium on*, May 2011, pp. 186–193.
- [83] M. Saeedi and I. L. Markov, “Synthesis and optimization of reversible circuits—a survey,” *ACM Comput. Surv.*, vol. 45, no. 2, pp. 21:1–21:34, Mar. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2431211.2431220>
- [84] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhang, “High-level synthesis for fpgas: From prototyping to deployment,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 4, pp. 473–491, April 2011.
- [85] A. Chakrabarti, K. Shea, R. Stone, J. Cagan, M. Campbell, N. V. Hernandez, and K. L. Wood, “Computer-based design synthesis research: An overview,” *Journal of Computing and Information Science in Engineering*, vol. 11, no. 2, pp. 021 003–021 003, June 2011. [Online]. Available: <http://dx.doi.org/10.1115/1.3593409>
- [86] A. Cilardo, E. Fusella, L. Gallo, and A. Mazzeo, “Automated synthesis of fpga-based heterogeneous interconnect topologies,” in *2013 23rd International Conference on Field programmable Logic and Applications*, Sept 2013, pp. 1–8.
- [87] J. Teich, “Hardware/software codesign: The past, the present, and predicting the future,” *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1411–1430, May 2012.

- [88] S.-A. Li, C.-C. Hsu, C.-C. Wong, and C.-J. Yu, “Hardware/software co-design for particle swarm optimization algorithm,” *Information Sciences*, vol. 181, no. 20, pp. 4582 – 4596, 2011, special Issue on Interpretable Fuzzy Systems. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S002002551000335X>
- [89] C. Weis, N. Wehn, L. Igor, and L. Benini, “Design space exploration for 3d-stacked drams,” in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–6.
- [90] C. Silvano, W. Fornaciari, G. Palermo, V. Zaccaria, F. Castro, M. Martinez, S. Bocchio, R. Zafalon, P. Avasare, G. Vanmeerbeeck, C. Ykman-Couvreux, M. Wouters, C. Kavka, L. Onesti, A. Turco, U. Bondi, G. Mariani, H. Posadas, E. Villar, C. Wu, F. Dongrui, Z. Hao, and T. Shibin, *MULTICUBE: Multi-Objective Design Space Exploration of Multi-Core Architectures*. Dordrecht: Springer Netherlands, 2011, pp. 47–63. [Online]. Available: http://dx.doi.org/10.1007/978-94-007-1488-5_4
- [91] B. Yu, J.-R. Gao, D. Ding, Y. Ban, J.-s. Yang, K. Yuan, M. Cho, and D. Z. Pan, “Dealing with ic manufacturability in extreme scaling,” in *Proceedings of the International Conference on Computer-Aided Design*, ser. ICCAD ’12. New York, NY, USA: ACM, 2012, pp. 240–242. [Online]. Available: <http://doi.acm.org/10.1145/2429384.2429430>
- [92] E. A. Lee and A. L. Sangiovanni-Vincentelli, “Component-based design for the future,” in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–5.
- [93] P. K. Sahu and S. Chattopadhyay, “A survey on application mapping strategies for network-on-chip design,” *Journal of Systems Architecture*, vol. 59, no. 1, pp. 60 – 76, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1383762112000902>
- [94] R. Alur, “Formal verification of hybrid systems,” in *Embedded Software (EMSOFT), 2011 Proceedings of the International Conference on*, Oct 2011, pp. 273–278.
- [95] Z. Chen, D. Zhang, R. Zhu, Y. Ma, P. Yin, and F. Xie, “A review of automated formal verification of ad hoc routing protocols for wireless sensor

- networks,” *Sensor Letters*, vol. 11, no. 5, pp. 752–764, 2013-05-01T00:00:00. [Online]. Available: <http://www.ingentaconnect.com/content/asp/senlet/2013/00000011/00000005/art00002>
- [96] D. Majstorovic, I. Celanovic, N. D. Teslic, N. Celanovic, and V. A. Katic, “Ultralow-latency hardware-in-the-loop platform for rapid validation of power electronics designs,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4708–4716, Oct 2011.
- [97] H. Yu, Y. Ma, Y. Glouche, J.-P. Talpin, L. Besnard, T. Gautier, P. L. Guernic, A. Toom, and O. Laurent, “System-level co-simulation of integrated avionics using polychrony,” in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC ’11. New York, NY, USA: ACM, 2011, pp. 354–359. [Online]. Available: <http://doi.acm.org/10.1145/1982185.1982263>
- [98] K. Gent and M. S. Hsiao, “Functional test generation at the rtl using swarm intelligence and bounded model checking,” in *2013 22nd Asian Test Symposium*, Nov 2013, pp. 233–238.
- [99] C. Ioannides and K. I. Eder, “Coverage-directed test generation automated by machine learning – a review,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 17, no. 1, pp. 7:1–7:21, Jan. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2071356.2071363>
- [100] H. Zeng, P. Kazemian, G. Varghese, and N. McKeown, “Automatic test packet generation,” in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT ’12. New York, NY, USA: ACM, 2012, pp. 241–252. [Online]. Available: <http://doi.acm.org/10.1145/2413176.2413205>
- [101] Z. Xu and J. Q. Lu, “High-speed design and broadband modeling of through-strata-vias (tsvs) in 3d integration,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 2, pp. 154–162, Feb 2011.
- [102] G. Shi, “A survey on binary decision diagram approaches to symbolic analysis of analog integrated circuits,” *Analog Integrated Circuits and Signal Processing*, vol. 74, no. 2, pp. 331–343, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s10470-011-9773-8>

- [103] C. Nie and H. Leung, “A survey of combinatorial testing,” *ACM Comput. Surv.*, vol. 43, no. 2, pp. 11:1–11:29, Feb. 2011. [Online]. Available: <http://doi.acm.org/10.1145/1883612.1883618>
- [104] P. A. Milder, R. Bouziane, R. Koutsoyannis, C. R. Berger, Y. Benlachtar, R. I. Killey, M. Glick, and J. C. Hoe, “Design and simulation of 25 gb/s optical ofdm transceiver asics,” *Opt. Express*, vol. 19, no. 26, pp. B337–B342, Dec 2011. [Online]. Available: <http://www.opticsexpress.org/abstract.cfm?URI=oe-19-26-B337>
- [105] Y. Thonnart, E. Beigne, and P. Vivet, “A pseudo-synchronous implementation flow for wchb qdi asynchronous circuits,” in *Asynchronous Circuits and Systems (ASYNC), 2012 18th IEEE International Symposium on*, May 2012, pp. 73–80.
- [106] F. Caraffini, F. Neri, and L. Picinali, “An analysis on separability for memetic computing automatic design,” *Information Sciences*, vol. 265, pp. 1 – 22, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025513009006>
- [107] A. K. Singh, M. Shafique, A. Kumar, and J. Henkel, “Mapping on multi/many-core systems: Survey of current and emerging trends,” in *Proceedings of the 50th Annual Design Automation Conference*, ser. DAC ’13. New York, NY, USA: ACM, 2013, pp. 1:1–1:10. [Online]. Available: <http://doi.acm.org/10.1145/2463209.2488734>
- [108] A. Morgado, F. Heras, M. Liffiton, J. Planes, and J. Marques-Silva, “Iterative and core-guided maxsat solving: A survey and assessment,” *Constraints*, vol. 18, no. 4, pp. 478–534, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s10601-013-9146-2>
- [109] V. Alimguzhin, F. Mari, I. Melatti, I. Salvo, and E. Tronci, “Automatic control software synthesis for quantized discrete time hybrid systems,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, Dec 2012, pp. 6120–6125.
- [110] J. Karsek, R. Burget, and O. Morsk, “Towards an automatic design of non-cryptographic hash function,” in *Telecommunications and Signal Processing (TSP), 2011 34th International Conference on*, Aug 2011, pp. 19–23.

- [111] A. Aleti, B. Buhnova, L. Grunske, A. Koziolok, and I. Meedeniya, “Software architecture optimization methods: A systematic literature review,” *IEEE Transactions on Software Engineering*, vol. 39, no. 5, pp. 658–683, May 2013.
- [112] Y. HACHAICHI and J. FEKI, “An automatic method for the design of multidimensional schemas from object oriented databases,” *International Journal of Information Technology & Decision Making*, vol. 12, no. 06, pp. 1223–1259, 2013. [Online]. Available: <http://www.worldscientific.com/doi/abs/10.1142/S0219622013500351>
- [113] R. C. Barros, M. P. Basgalupp, A. C. P. L. F. de Carvalho, and A. A. Freitas, “Automatic design of decision-tree algorithms with evolutionary algorithms,” *Evolutionary Computation*, vol. 21, no. 4, pp. 659–684, Jan. 2013. [Online]. Available: http://dx.doi.org/10.1162/EVCO_a_00101
- [114] L. M. L. d. Campos, M. Roisenberg, and R. C. L. d. Oliveira, “Automatic design of neural networks with l-systems and genetic algorithms - a biologically inspired methodology,” in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, July 2011, pp. 1199–1206.
- [115] R. C. Barros, A. T. Winck, K. S. Machado, M. P. Basgalupp, A. C. de Carvalho, D. D. Ruiz, and O. N. de Souza, “Automatic design of decision-tree induction algorithms tailored to flexible-receptor docking data,” *BMC Bioinformatics*, vol. 13, no. 1, pp. 1–14, 2012. [Online]. Available: <http://dx.doi.org/10.1186/1471-2105-13-310>
- [116] R. C. Barros, M. P. Basgalupp, A. C. de Carvalho, and A. A. Freitas, “Towards the automatic design of decision tree induction algorithms,” in *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation*, ser. GECCO '11. New York, NY, USA: ACM, 2011, pp. 567–574. [Online]. Available: <http://doi.acm.org/10.1145/2001858.2002050>
- [117] A. Farahat and A. Ebneenasir, “A lightweight method for automated design of convergence in network protocols,” *ACM Trans. Auton. Adapt. Syst.*, vol. 7, no. 4, pp. 38:1–38:36, Dec. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2382570.2382574>

- [118] M. Lopez-Ibanez and T. Stutzle, “The automatic design of multiobjective ant colony optimization algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 6, pp. 861–875, Dec 2012.
- [119] W. Yang, M. R. Prasad, and T. Xie, *A Grey-Box Approach for Automated GUI-Model Generation of Mobile Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 250–265. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37057-1_19
- [120] M. Semenkina and E. Semenin, *Hybrid Self-configuring Evolutionary Algorithm for Automated Design of Fuzzy Classifier*. Cham: Springer International Publishing, 2014, pp. 310–317. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-11857-4_35
- [121] R. Nehme and N. Bruno, “Automated partitioning design in parallel database systems,” in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD ’11. New York, NY, USA: ACM, 2011, pp. 1137–1148. [Online]. Available: <http://doi.acm.org/10.1145/1989323.1989444>
- [122] L. Liu, D. Sheridan, V. Athavale, and S. Vasudevan, “Automatic generation of assertions from system level design using data mining,” in *Formal Methods and Models for Codesign (MEMOCODE), 2011 9th IEEE/ACM International Conference on*, July 2011, pp. 191–200.
- [123] A. Filieri, H. Hoffmann, and M. Maggio, “Automated design of self-adaptive software with control-theoretical formal guarantees,” in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE 2014. New York, NY, USA: ACM, 2014, pp. 299–310. [Online]. Available: <http://doi.acm.org/10.1145/2568225.2568272>
- [124] C. Ykman-Couvreur, P. Avasare, G. Mariani, G. Palermo, C. Silvano, and V. Zaccaria, “Linking run-time resource management of embedded multi-core platforms with automated design-time exploration,” *IET Computers Digital Techniques*, vol. 5, no. 2, pp. 123–135, March 2011.
- [125] W. Jia, K. A. Shaw, and M. Martonosi, “Stargazer: Automated regression-based gpu design space exploration,” in *Performance Analysis of Systems and*

- Software (ISPASS)*, 2012 IEEE International Symposium on, April 2012, pp. 2–13.
- [126] E. Guerra, J. de Lara, M. Wimmer, G. Kappel, A. Kusel, W. Retschitzegger, J. Schönböck, and W. Schwinger, “Automated verification of model transformations based on visual contracts,” *Automated Software Engineering*, vol. 20, no. 1, pp. 5–46, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s10515-012-0102-y>
- [127] D. Batory, R. Gonçalves, B. Marker, and J. Siegmund, *Dark Knowledge and Graph Grammars in Automated Software Design*. Cham: Springer International Publishing, 2013, pp. 1–18. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-02654-1_1
- [128] A. Egyed, “Automatically detecting and tracking inconsistencies in software design models,” *IEEE Transactions on Software Engineering*, vol. 37, no. 2, pp. 188–204, March 2011.
- [129] J. Xu, “Rule-based automatic software performance diagnosis and improvement,” *Performance Evaluation*, vol. 69, no. 11, pp. 525 – 550, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0166531609001631>
- [130] A. G. Bayrak, F. Regazzoni, D. Novo, and P. Ienne, *Sleuth: Automated Verification of Software Power Analysis Countermeasures*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 293–310. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40349-1_17
- [131] B. N. Nguyen, B. Robbins, I. Banerjee, and A. Memon, “Guitar: an innovative tool for automated testing of gui-driven software,” *Automated Software Engineering*, vol. 21, no. 1, pp. 65–105, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s10515-013-0128-9>
- [132] D. Amalfitano, A. R. Fasolino, P. Tramontana, S. De Carmine, and A. M. Memon, “Using gui ripping for automated testing of android applications,” in *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE 2012. New York, NY, USA: ACM, 2012, pp. 258–261. [Online]. Available: <http://doi.acm.org/10.1145/2351676.2351717>

- [133] P. Gilbert, B.-G. Chun, L. P. Cox, and J. Jung, “Vision: Automated security validation of mobile apps at app markets,” in *Proceedings of the Second International Workshop on Mobile Cloud Computing and Services*, ser. MCS ’11. New York, NY, USA: ACM, 2011, pp. 21–26. [Online]. Available: <http://doi.acm.org/10.1145/1999732.1999740>
- [134] S. Bucur, V. Ureche, C. Zamfir, and G. Candea, “Parallel symbolic execution for automated real-world software testing,” in *Proceedings of the Sixth Conference on Computer Systems*, ser. EuroSys ’11. New York, NY, USA: ACM, 2011, pp. 183–198. [Online]. Available: <http://doi.acm.org/10.1145/1966445.1966463>
- [135] T. Ball, V. Levin, and S. K. Rajamani, “A decade of software model checking with slam,” *Commun. ACM*, vol. 54, no. 7, pp. 68–76, July 2011. [Online]. Available: <http://doi.acm.org/10.1145/1965724.1965743>
- [136] K. Lind and R. Heldal, *A Model-Based and Automated Approach to Size Estimation of Embedded Software Components*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 334–348. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24485-8_24
- [137] D. Westermann, R. Krebs, and J. Happe, *Efficient Experiment Selection in Automated Software Performance Evaluations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 325–339. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24749-1_24
- [138] A. Reder and A. Egyed, “Computing repair trees for resolving inconsistencies in design models,” in *Automated Software Engineering (ASE), 2012 Proceedings of the 27th IEEE/ACM International Conference on*, Sept 2012, pp. 220–229.
- [139] M. Kessentini, W. Kessentini, H. Sahraoui, M. Boukadoum, and A. Ouni, “Design defects detection and correction by example,” in *Program Comprehension (ICPC), 2011 IEEE 19th International Conference on*, June 2011, pp. 81–90.
- [140] S. Anand, E. K. Burke, T. Y. Chen, J. Clark, M. B. Cohen, W. Grieskamp, M. Harman, M. J. Harrold, P. McMinn, A. Bertolino, J. J. Li, and H. Zhu, “An orchestrated survey of methodologies

- for automated software test case generation,” *Journal of Systems and Software*, vol. 86, no. 8, pp. 1978 – 2001, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121213000563>
- [141] D. M. Rafi, K. R. K. Moses, K. Petersen, and M. V. Mäntylä, “Benefits and limitations of automated software testing: Systematic literature review and practitioner survey,” in *Proceedings of the 7th International Workshop on Automation of Software Test*, ser. AST ’12. Piscataway, NJ, USA: IEEE Press, 2012, pp. 36–42. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2663608.2663616>
- [142] A. Christopoulou, E. Giakoumakis, V. E. Zafeiris, and V. Soukara, “Automated refactoring to the strategy design pattern,” *Information and Software Technology*, vol. 54, no. 11, pp. 1202 – 1214, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950584912001024>
- [143] L. F. Perrone, C. S. Main, and B. C. Ward, “Safe: Simulation automation framework for experiments,” in *Proceedings of the Winter Simulation Conference*, ser. WSC ’12. Winter Simulation Conference, 2012, pp. 249:1–249:12. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2429759.2430095>
- [144] B. Akay and D. Karaboga, “Artificial bee colony algorithm for large-scale problems and engineering design optimization,” *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1001–1014, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s10845-010-0393-4>
- [145] J. qing Zhao, L. Wang, P. Zeng, and W. hui Fan, “An effective hybrid genetic algorithm with flexible allowance technique for constrained engineering design optimization,” *Expert Systems with Applications*, vol. 39, no. 5, pp. 6041 – 6051, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417411016836>
- [146] T. V. Sibalija and V. D. Majstorovic, “An integrated approach to optimise parameter design of multi-response processes based on taguchi method and artificial intelligence,” *Journal of Intelligent Manufacturing*, vol. 23, no. 5, pp. 1511–1528, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s10845-010-0451-y>

- [147] K. Amadori, M. Tarkian, J. Ivander, and P. Krus, “Flexible and robust {CAD} models for design automation,” *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 180 – 195, 2012, knowledge based engineering to support complex product design. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474034612000055>
- [148] D. J. Gerber, S.-H. E. Lin, B. P. Pan, and A. S. Solmaz, “Design optioneering: Multi-disciplinary design optimization through parameterization, domain integration and automation of a genetic algorithm,” in *Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design*, ser. SimAUD ’12. San Diego, CA, USA: Society for Computer Simulation International, 2012, pp. 11:1–11:8. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2339453.2339464>
- [149] J. R. R. A. Martins and A. B. Lambe, “Multidisciplinary design optimization: A survey of architectures,” *AIAA Journal*, vol. 51, no. 9, pp. 2049–2075, July 2013. [Online]. Available: <http://dx.doi.org/10.2514/1.J051895>
- [150] T. Robinson, C. Armstrong, and R. Fairey, “Automated mixed dimensional modelling from 2d and 3d {CAD} models,” *Finite Elements in Analysis and Design*, vol. 47, no. 2, pp. 151 – 165, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168874X10001447>
- [151] G.-D. Kapos, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, “An integrated framework for automated simulation of sysml models using devs,” *SIMULATION*, vol. 90, no. 6, pp. 717–744, 2014. [Online]. Available: <http://sim.sagepub.com/content/90/6/717.abstract>
- [152] K. Hoad, S. Robinson, and R. Davies, “Autosimoo: a framework for automated analysis of simulation output,” *Journal of Simulation*, vol. 5, no. 1, pp. 9–24, 2011. [Online]. Available: <http://dx.doi.org/10.1057/jos.2010.22>
- [153] G. S. Hornby, J. D. Lohn, and D. S. Linden, “Computer-automated evolution of an x-band antenna for nasa’s space technology 5 mission,” *Evolutionary Computation*, vol. 19, no. 1, pp. 1–23, March 2011.
- [154] Z. Ding, M. Jiang, H. Chen, Z. Jin, and M. Zhou, “Petri net based test case generation for evolved specification,” *Science China Information*

- Sciences*, vol. 59, no. 8, p. 080105, Jul 2016. [Online]. Available: <https://doi.org/10.1007/s11432-016-5598-5>
- [155] P. Zhang, S. Shu, and M. Zhou, “An online fault detection model and strategies based on svm-grid in clouds,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, pp. 445–456, Mar 2018.
- [156] G. L. Rocca, “Knowledge based engineering: Between {AI} and cad. review of a language based technology to support engineering design,” *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 159 – 179, 2012, knowledge based engineering to support complex product design. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474034612000092>
- [157] R. Sung, J. M. Ritchie, H. J. Rea, and J. Corney, “Automated design knowledge capture and representation in single-user cad environments,” *Journal of Engineering Design*, vol. 22, no. 7, pp. 487–503, 2011. [Online]. Available: <http://dx.doi.org/10.1080/09544820903527187>
- [158] C.-C. Gu, J. Hu, Y.-H. Peng, and S. Li, “Fcbs model for functional knowledge representation in conceptual design,” *Journal of Engineering Design*, vol. 23, no. 8, pp. 577–596, 2012. [Online]. Available: <http://dx.doi.org/10.1080/09544828.2011.629318>
- [159] A. K. Goel, S. Vattam, B. Wiltgen, and M. Helms, “Cognitive, collaborative, conceptual and creative four characteristics of the next generation of knowledge-based {CAD} systems: A study in biologically inspired design,” *Computer-Aided Design*, vol. 44, no. 10, pp. 879 – 900, 2012, fundamentals of Next Generation CAD/E Systems. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0010448511000820>
- [160] Y. Chen, Z.-L. Liu, and Y.-B. Xie, “A knowledge-based framework for creative conceptual design of multi-disciplinary systems,” *Computer-Aided Design*, vol. 44, no. 2, pp. 146 – 153, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0010448511000613>
- [161] F. Demoly, X.-T. Yan, B. Eynard, L. Rivest, and S. Gomes, “An assembly oriented design framework for product structure engineering and assembly sequence planning,” *Robotics and Computer-Integrated*

- Manufacturing*, vol. 27, no. 1, pp. 33 – 46, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0736584510000517>
- [162] K. Maniya and M. Bhatt, “An alternative multiple attribute decision making methodology for solving optimal facility layout design selection problems,” *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 542 – 549, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0360835211001100>
- [163] T. V. Sibalija, V. D. Majstorovic, and Z. D. Miljkovic, “An intelligent approach to robust multi-response process design,” *International Journal of Production Research*, vol. 49, no. 17, pp. 5079–5097, 2011. [Online]. Available: <http://dx.doi.org/10.1080/00207543.2010.511476>
- [164] S. Chen, Y. Zheng, C. Cattani, and W. Wang, “Modeling of biological intelligence for scm system optimization,” p. 10, 2012. [Online]. Available: <http://dx.doi.org/10.1155/2012/769702>
- [165] C. C. Hayes, A. K. Goel, I. Y. Tumer, A. M. Agogino, and W. C. Regli, “Intelligent support for product design: Looking backward, looking forward,” *Journal of Computing and Information Science in Engineering*, vol. 11, no. 2, pp. 021007–021007, June 2011. [Online]. Available: <http://dx.doi.org/10.1115/1.3593410>
- [166] M. Bhatt and C. Freksa, *Spatial Computing for Design—an Artificial Intelligence Perspective*. Dordrecht: Springer Netherlands, 2015, pp. 109–127. [Online]. Available: http://dx.doi.org/10.1007/978-94-017-9297-4_7
- [167] M. Utting, A. Pretschner, and B. Legeard, “A taxonomy of model-based testing approaches,” *Software Testing, Verification and Reliability*, vol. 22, no. 5, pp. 297–312, 2012. [Online]. Available: <http://dx.doi.org/10.1002/stvr.456>
- [168] H. Wang, A. L. Johnson, and R. H. Bracewell, “The retrieval of structured design rationale for the re-use of design knowledge with an integrated representation,” *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 251 – 266, 2012, knowledge based engineering to support complex product design. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474034612000109>

- [169] S. Krish, “A practical generative design method,” *Computer-Aided Design*, vol. 43, no. 1, pp. 88 – 100, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0010448510001764>
- [170] P. Baroni, M. Romano, F. Toni, M. Aurisicchio, and G. Bertanza, *An Argumentation-Based Approach for Automatic Evaluation of Design Debates*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 340–356. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40624-9_21
- [171] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297. [Online]. Available: <http://dx.doi.org/10.1007/BF00994018>
- [172] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167. [Online]. Available: <http://dx.doi.org/10.1023/A:1009715923555>
- [173] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, “Core vector machines: Fast svm training on very large data sets,” *J. Mach. Learn. Res.*, vol. 6, pp. 363–392, Dec. 2005. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1046920.1058114>
- [174] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.
- [175] W. Zhang, H. Zhang, J. Liu, K. Li, D. Yang, and H. Tian, “Weather prediction with multiclass support vector machines in the fault detection of photovoltaic system,” *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 520–525, 2017.
- [176] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Fisher Vector Faces in the Wild,” in *British Machine Vision Conference*, 2013.
- [177] J. Wu, W.-C. Tan, and J. M. Rehg, “Efficient and effective visual codebook generation using additive kernels,” *J. Mach. Learn. Res.*, vol. 12, pp. 3097–3118, Nov. 2011. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1953048.2078205>

- [178] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sept 2010.
- [179] E. Yildizer, A. M. Balci, M. Hassan, and R. Alhajj, “Efficient content-based image retrieval using multiple support vector machines ensemble,” *Expert Systems with Applications*, vol. 39, no. 3, pp. 2385 – 2396, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417411012176>
- [180] D. Tao, L. Jin, W. Liu, and X. Li, “Hessian regularized support vector machines for mobile image annotation on the cloud,” *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 833–844, June 2013.
- [181] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.
- [182] A. Chaudhuri and K. De, “Fuzzy support vector machine for bankruptcy prediction,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2472 – 2486, 2011, the Impact of Soft Computing for the Progress of Artificial Intelligence. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S156849461000253X>
- [183] G. Orr, W. Pettersson-Yeo, A. F. Marquand, G. Sartori, and A. Mechelli, “Using support vector machine to identify imaging biomarkers of neurological and psychiatric disease: A critical review,” *Neuroscience & Biobehavioral Reviews*, vol. 36, no. 4, pp. 1140 – 1152, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0149763412000139>
- [184] P. Konar and P. Chattopadhyay, “Bearing fault detection of induction motor using wavelet and support vector machines (svms),” *Applied Soft Computing*, vol. 11, no. 6, pp. 4203 – 4211, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494611001104>
- [185] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.

- [186] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: <http://dx.doi.org/10.1038/nature14539>
- [187] G. Huang, G.-B. Huang, S. Song, and K. You, “Trends in extreme learning machines: A review,” *Neural Networks*, vol. 61, pp. 32 – 48, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608014002214>
- [188] R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning, “Parsing Natural Scenes and Natural Language with Recursive Neural Networks,” in *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2011.
- [189] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *CoRR*, vol. abs/1406.1078, 2014. [Online]. Available: <http://arxiv.org/abs/1406.1078>
- [190] J. T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, “Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 7304–7308.
- [191] R. Sarikaya, G. E. Hinton, and A. Deoras, “Application of deep belief networks for natural language understanding,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 778–784, April 2014.
- [192] L. Deng, G. Tur, X. He, and D. Hakkani-Tur, “Use of kernel deep convex networks and end-to-end learning for spoken language understanding,” in *Spoken Language Technology Workshop (SLT), 2012 IEEE*, Dec 2012, pp. 210–215.
- [193] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
- [194] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *British Machine Vision Conference*, 2015.

- [195] W. Ouyang and X. Wang, “Joint deep learning for pedestrian detection,” in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 2056–2063.
- [196] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, “Convolutional-recursive deep learning for 3d object classification,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 656–664. [Online]. Available: <http://papers.nips.cc/paper/4773-convolutional-recursive-deep-learning-for-3d-object-classification.pdf>
- [197] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, “Learning deep features for scene recognition using places database,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 487–495. [Online]. Available: <http://papers.nips.cc/paper/5349-learning-deep-features-for-scene-recognition-using-places-database.pdf>
- [198] S.-h. Zhong, Y. Liu, and Y. Liu, “Bilinear deep learning for image classification,” in *Proceedings of the 19th ACM International Conference on Multimedia*, ser. MM ’11. New York, NY, USA: ACM, 2011, pp. 343–352. [Online]. Available: <http://doi.acm.org/10.1145/2072298.2072344>
- [199] D. Ciregan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 3642–3649.
- [200] C. Dong, C. C. Loy, K. He, and X. Tang, *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV*. Cham: Springer International Publishing, 2014, ch. Learning a Deep Convolutional Network for Image Super-Resolution, pp. 184–199. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10593-2_13
- [201] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22Nd ACM International Conference on Multimedia*, ser. MM ’14. New York, NY, USA: ACM,

- 2014, pp. 675–678. [Online]. Available: <http://doi.acm.org/10.1145/2647868.2654889>
- [202] M. Ranzato, F. J. Huang, Y. L. Boureau, and Y. LeCun, “Unsupervised learning of invariant feature hierarchies with applications to object recognition,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.
- [203] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.
- [204] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [205] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [206] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: <http://arxiv.org/abs/1608.06993>
- [207] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, “Network dissection: Quantifying interpretability of deep visual representations,” in *Computer Vision and Pattern Recognition*, 2017.
- [208] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, October 2010.
- [209] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, “Transfer learning using computational intelligence: A survey,” *Knowledge-Based Systems*, vol. 80, no. Supplement C, pp. 14 – 23, 2015, 25th anniversary of Knowledge-Based Systems. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950705115000179>

- [210] L. Shao, F. Zhu, and X. Li, “Transfer learning for visual categorization: A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019–1034, May 2015.
- [211] H. C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, May 2016.
- [212] T. Heimann, P. Mountney, M. John, and R. Ionasec, “Real-time ultrasound transducer localization in fluoroscopy images by transfer learning from synthetic training data,” *Medical Image Analysis*, vol. 18, no. 8, pp. 1320 – 1328, 2014, special Issue on the 2013 Conference on Medical Image Computing and Computer Assisted Intervention. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1361841514000619>
- [213] M. L. Littman, “Reinforcement learning improves behaviour from evaluative feedback,” *Nature*, vol. 521, no. 7553, pp. 445–451, May 2015. [Online]. Available: <http://dx.doi.org/10.1038/nature14540>
- [214] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016. [Online]. Available: <http://dx.doi.org/10.1038/nature16961>
- [215] Y. Tang, “Deep learning using support vector machines,” *CoRR*, vol. abs/1306.0239, 2013. [Online]. Available: <http://arxiv.org/abs/1306.0239>
- [216] T. Liu, B. Tian, Y. Ai, L. Li, D. Cao, and F. Wang, “Parallel reinforcement learning: a framework and case study,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 4, pp. 827–835, July 2018.
- [217] R. Raina, A. Madhavan, and A. Y. Ng, “Large-scale deep unsupervised learning using graphics processors,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML ’09. New

- York, NY, USA: ACM, 2009, pp. 873–880. [Online]. Available: <http://doi.acm.org/10.1145/1553374.1553486>
- [218] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511–I–518 vol.1.
- [219] International Maritime Organization, “Annex I Regulations for the prevention of pollution by oil, International convention for the prevention of pollution from ships (marpol).” [Online]. Available: [http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/International-Convention-for-the-Prevention-of-Pollution-from-Ships-\(MARPOL\).aspx](http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/International-Convention-for-the-Prevention-of-Pollution-from-Ships-(MARPOL).aspx)
- [220] —, “Resolution MEPC.117(52) Amendments to the annex of the protocol of 1978 relating to the International convention for the prevention of pollution from ships, 1973,” 2004. [Online]. Available: [http://www.imo.org/blast/blastDataHelper.asp?data_id=15720&filename=117\(52\).pdf](http://www.imo.org/blast/blastDataHelper.asp?data_id=15720&filename=117(52).pdf)
- [221] —, “Resolution MEPC.187(59) Amendments to the annex of the protocol of 1978 relating to the International convention for the prevention of pollution from ships, 1973,” 2009. [Online]. Available: [http://www.imo.org/blast/blastDataHelper.asp?data_id=26472&filename=187\(59\).pdf](http://www.imo.org/blast/blastDataHelper.asp?data_id=26472&filename=187(59).pdf)
- [222] Lloyd’s Register, “Rules and regulations for the classification of ships.” [Online]. Available: <http://www.lr.org/en/RulesandRegulations/ships.aspx>
- [223] D. G. Austin, *Chemical Engineering Drawing Symbols*. G. Godwin, 1979.
- [224] D. Liu, H. Qian, G. Dai, and Z. Zhang, “An iterative {SVM} approach to feature selection and classification in high-dimensional datasets,” *Pattern Recognition*, vol. 46, no. 9, pp. 2531 – 2537, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031320313000836>
- [225] G. Bradski, *Dr. Dobb’s Journal of Software Tools*, 2000.
- [226] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “Liblinear: A library for large linear classification,” *J. Mach.*

- Learn. Res.*, vol. 9, pp. 1871–1874, June 2008. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1390681.1442794>
- [227] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: <https://doi.org/10.1023/A:1010933404324>
- [228] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511–I–518 vol.1.
- [229] W. C. T. S. J. Pan, I. M. Chen, and H. K. Tan, “Automated design evaluation on layout of piping and instrumentation diagram using histogram of connectivity,” in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*, 2016.
- [230] C.-J. H. X.-R. W. R.-E. Fan, K.-W. Chang and C.-J. Lin, “Liblinear: A library for large linear classification,” *Journal of Machine Learning Research*, 2008.
- [231] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [232] W. C. Tan, I. M. Chen, S. J. Pan, and H. K. Tan, “Analysis with histogram of connectivity: For automated evaluation of piping layout,” *IEEE Transactions on Automation Science and Engineering*, vol. PP, no. 99, pp. 1–12, 2017.
- [233] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [234] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [235] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv preprint arXiv:1408.5093*, 2014.

- [236] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>
- [237] I. A. of Classification Societies, “Uniform application of marpol annex i, revised regulation 12,” 2018.