POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Polygon Generation & Deep Learning: Towards Industrial Data Fusion for Optimizing Engineering Systems' Performance

MOHAMED TAREK MOHAMED ELHEFNAWY

Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*

Génie industriel

Février 2022

© Mohamed Tarek Mohamed Elhefnawy, 2022.

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée:

Polygon Generation & Deep Learning: Towards Industrial Data Fusion for Optimizing Engineering Systems' Performance

présentée par **Mohamed Tarek Mohamed ELHEFNAWY**

en vue de l'obtention du diplôme de *Philosophiæ Doctor*

a été dûment acceptée par le jury d'examen constitué de :

Luc Désiré ADJENGUE, président
Mohamed-Salah OUALI, membre et directeur
Ahmed RAGAB, membre et codirecteur
Mouloud AMAZOUZ, membre
Elsayed HEMAYED, membre externe

DEDICATION

To my beloved parents and sister. Nothing could be done without their endless support and love.

To Dr. Ahmed Zewail who inspired me to go beyond my imagination through knowledge, perseverance & self-confidence.

ACKNOWLEDGEMENTS

There are many people I wish to thank for their help on this thesis. First, I would like to express my profound gratitude to my research director Dr. Mohamed-Salah Ouali whom I have had the privilege of being his student and whom I am heavily indebted to. I really appreciate his comments, guidance, support, and patience during the whole research process. He shared with me his knowledge that is valuable as gold. I am even more fortunate that he generously gave me more of his valuable time to complete this work.

My profound gratitude goes as well to my co-director of research Dr. Ahmed Ragab whom I received wisdom and knowledge. His words pushed me a lot, provided me with enough confidence to get the best of me. He shared with me his experience and knowledge throughout the work. I really appreciate his dedication and insightful comments. I am deeply grateful to his suggestions for improvements and willingness to go deeply in the technical details throughout the research process. I learned a lot from him not only in the scientific research but in many life aspects. He was like a second father of me who tries to make his son better than him. I am really grateful to him for giving me the chance to work with him and to know my weaknesses before my strengths.

I am honored to have Dr. Luc Adjengue, Dr. Mouloud Amazouz, and Dr. Elsayed Hemayed as jury members to evaluate my thesis.

My gratitude and appreciation go as well to Mr. Hakim Ghezzaz for providing me continuously with technical support, invaluable expertise and prepared industrial datasets for validating the proposed methods in this thesis work. I would like to extend my thanks and appreciation to Mennah and May Hammad for their technical support to work with high performance computing (HPC) infrastructure at Natural Resources Canada. I want to thank all my friends in Montreal whom I consider my second family. Their support and love helped me a lot during this journey.

RÉSUMÉ

L'analyse des données industrielles pose plusieurs défis. Ces données sont collectées à partir de sources hétérogènes et stockées dans des silos isolés. Cela complique leur accessibilité et empêche leur pleine exploitation dans la prise de décision. Par conséquent, il existe un besoin urgent et prioritaire de fusionner ces silos déconnectés pour profiter au maximum de ces données. Toutefois, la fusion de données de sources multiples présente deux principaux défis : la représentation efficace des données afin de permettre une meilleure fusion des données, et la sélection des méthodes de modélisation les plus appropriées en fonction du type et de la quantité des données ainsi que la nature du problème à résoudre. Relever ces défis contribuera à ce que les systèmes industriels complexes maintiennent un fonctionnement économe en énergie, à réduire leur empreinte environnementale et ainsi à atteindre l'excellence opérationnelle.

Cette thèse apporte trois contributions dans résolution de la problématique de la fusion de données issues de l'exploitation de systèmes industriels. L'objectif des trois contributions proposées est d'obtenir une représentation des données efficace et utile qui permet de maximiser la valeur globale des données industrielles disponibles pour faciliter le processus de fusion à différents niveaux sémantiques (brut, information et connaissance). Les contributions utilisent les performances de l'apprentissage profond (DL) dans la modélisation prédictive et générative. Elles permettent de construire des modèles précis et robustes utilisables à plusieurs finalités dans l'industrie. Ces modèles peuvent diagnostiquer avec précision divers systèmes industriels, prédire leurs indicateurs clés de performance et capturer la vraie distribution des systèmes industriels complexes hautement non linéaires pour fournir à l'expert des connaissances précieuses pour prescrire les bonnes actions au bon moment.

Toutes les méthodes proposées dans cette recherche doctorale ont été validées à l'aide d'ensembles de données complexes provenant de différents équipements complexes des procédés industriels : Rebouilleur dans une usine de pâte thermomécanique (TMP), chaudière de récupération de liqueur noire (BLRB) et concentrateur de liqueur noire dans une usine de pâte Kraft. Les méthodes proposées ont été comparées à différentes méthodes d'apprentissage automatique (ML) et d'apprentissage profond largement utilisées dans la littérature. Les résultats indiquent que nos méthodes surpassent les méthodes comparables en termes de précision de prédiction.

Les résultats obtenus montrent que notre approche conduit à des prédictions précises et arrive à capturer la non-linéarité et le comportement dynamique de systèmes industriels complexes avec des distributions mal définies. Cette recherche doctorale ouvre la porte à la fusion de données hétérogènes industrielles incluant des données sensorielles, images, et vidéos.

ABSTRACT

Analysis of industrial data imposes several challenges. This type of data is collected from heterogeneous sources and stored in isolated silos. This hinders the accessibility of this asset and its proper exploitation in the decision making process. Therefore, there is an urgent and prioritized need to merge these disconnected silos to maximize the benefits from this asset. However, merging data from multiple sources has two main challenges: efficient data representation for better data fusion, and selecting the most appropriate modeling technique based on the type and quantity of data as well as the nature of the problem to be solved. Solving these challenges will help the complex industrial systems maintain energy-efficient operations, reduce their environmental footprint, and thus achieve operational excellence.

This thesis makes three contributions as a step for solving the problems of the data fusion resulting from the operation of industrial systems. The objective of the three proposed contributions is to obtain an efficient and useful data representation that helps maximize the global value of the available industrial data to facilitate the fusion process at different semantic levels (raw, information and knowledge levels). Those contributions make use of the performance of deep learning (DL) in predictive and generative modeling. They allow for building accurate and robust models to be used for several purposes in industry. These models can accurately diagnose various industrial systems, predict their key performance indicators (KPIs) and capture the true distribution of highly non-linear complex industrial systems to provide the expert with valuable knowledge to prescribe the right actions at the right time.

All the proposed methods in this doctoral research are validated using challenging datasets collected from different complex equipment in process industries: reboiler system in a theromomechancal pulp mill (TMP), black liquor recovery boiler (BLRB) and concentrator of black liquor in a Kraft pulp mill. The methods were compared to different machine learning and deep learning techniques extensively used in the literature. Our results indicate that our methods outperform the comparable methods in terms of prediction accuracy.

The results obtained show that our proposed approach leads to accurate predictions and successfully captures the non-linearity and dynamic behavior of complex industrial systems with ill-defined distributions. This doctoral research opens the door to the industrial heterogeneous data fusion including sensory data, images, and videos.

TABLE OF CONTENTS

DEDICATION III

ACKNOWLEDGEMENTS IV

ABSTRACT VII

LIST OF TABLES XII

LIST OF FIGURES XIV

LIST OF ABBREVIATIONS AND NOTATIONS XVII

CHAPTER 1 INTRODUCTION 1

1.1 Problem statement 2

1.2 Performance prediction of industrial systems: Limitations of existing methods 4

1.3 Deep learning opportunities for accurate prediction of system performance 5

1.4 Research Objectives 6

1.4.1 General objective 6

1.4.2 Specific research objectives 7

1.5 Originality and Success 7

1.6 Scientific Contributions and Deliverables 8

1.7 Potential Impacts: Social and Economic Benefits 9

1.8 Thesis organization 11

CHAPTER 2 RELATED WORK, CHALLENGES & RESEARCH DIRECTIONS 13

2.1 Overview on Data Fusion 13

2.2 Challenges of Existing Data Fusion Methods 14

2.3 Research Directions & Recommendations to Overcome Data Fusion Challenges 16

2.4 Proposed Approach: Polygon Generation and Deep Learning Towards Data Fusion 16

CHAPTER 3 ARTICLE 1: FAULT CLASSIFICATION IN THE PROCESS INDUSTRY USING POLYGON GENERATION AND DEEP LEARNING 18

3.1 Abstract 19

3.2 Introduction 19

3.3 Industrial Fault Classification Using Deep Learning Methods 21

3.4 Proposed method 23

3.4.1 Preprocessing Stage 24

3.4.2 Learning Stage: CNN training and testing 30

3.5 Fault classification case studies 31

3.5.1 Case Study 1: Tennessee Eastman Process (TEP) 31

3.5.2 Case Study 2: The Reboiler System of Heat Recovery Network in TMP Mills 33

3.6 Experimental setup & Results 35

3.6.1 Results of the TEP dataset 37

3.6.2 Results of the Reboiler dataset 39

3.6.3 Remarks 40

3.6.4 Future work 41

3.7 Conclusion 41

3.8 References 42

CHAPTER 4 ARTICLE 2: MULTI-OUTPUT REGRESSION USING POLYGON GENERATION AND CONDITIONAL GENERATIVE ADVERSARIAL NETWORKS 64

4.1 Abstract 65

4.2 Introduction 66

4.3 Background & Related Work 69

4.3.1 Polygon Generation from Numerical Data 69

4.3.2 Generative Modeling: The Generative Adversarial Networks (GANs) 73

4.3.3 Conditional GAN & Image-to-image Translation 75

4.4 Proposed Method: Multi-Output Regression Using Polygon Generation and cGAN ... 77

4.4.1 Training Phase: Building Generator Model 77

4.4.2 Testing Phase: Mapping Images into Numeric Outputs 78

4.5 Case Study: Black Liquor Recovery Boiler (BLRB) in Kraft Pulp & Paper Mills 81

4.5.1 BLRB: Operation and KPIs 81

4.5.2 Experimental setup & Results 82

4.6 Results, Discussion & Future Work 85

4.6.1 Results 87

4.6.2 Discussion and Future Work 91

4.7 Conclusion 93

4.8 References 94

CHAPTER 5 ARTICLE 3: POLYGON GENERATION AND VIDEO-TO-VIDEO TRANSLATION FOR TIME-SERIES PREDICTION IN INDUSTRIAL SYSTEMS 116

5.1 Abstract 117

5.2 Introduction 117

5.3 Background & Related Work 121

5.3.1 Deep Learning for Time-series Prediction in Industrial Systems 121

5.3.2 Polygon Generation for Data Representation 122

5.3.3 Unsupervised Video-to-Video Translation 126

5.4 Proposed Method for Time-Series Prediction 129

5.4.1 Training Phase: Unsupervised Video-to-Video Translation 130

5.4.2 Testing Phase: Mapping Videos into Time-series Outputs 133

5.5 Case Study: Concentrator in Heat Recovery Network (HRN) 134

5.5.1 System Operation and KPIs 135

5.5.2 Experimental Setup 136

5.6 Results, Discussion & Future Work 139

5.6.1 Results 141

5.6.2 Discussion and Future Work 143

5.7 Conclusion 145

5.8 References 145

CHAPTER 6 GENERAL DISCUSSION, CONCLUSION & FUTURE WORK 153

REFERENCES 157

LIST OF TABLES

Table 3.1 - Algorithm: Calculating the Hamiltonian path matrix (Hurley & Oldford, 2010) 29

Table 3.2 - Examples of manipulated, measured, and calculated variables for the HRN.....34

Table 3.3 - Baseline Convolutional Neural Network (CNN) architecture.....36

Table 3.4 - Range of hyperparameters of each fault classifier.....37

Table 3.5 - F1 Scores, Accuracy and False alarm rate (FAR) of each algorithm in TEP dataset..39

Table 3.6 - F1 Scores, Accuracy and False alarm rate (FAR) of each algorithm in the Reboiler data 40

Table 4.1 - Calculations of point coordinates $\overrightarrow{X_j^1}$ on the sides of the polygon for a numeric observation with five variables shown in Figure 1, where $\hat{q}$ and $\hat{l}$ are the unit vectors of x and y directions, respectively 71

Table 4.2 - Calculations of point coordinates $\overrightarrow{Y_h^1}$ on polygon sides for a numeric observation with three outputs shown in Figure 2, where $\hat{q}$ and $\hat{l}$ are the unit vectors of the x and y directions, respectively.....73

Table 4.3 - Examples of manipulated, measured variables and KPIs for the BLRB.....82

Table 4.4 - Range of hyperparameters of each regressor.....85

Table 4.5 - Underestimation (α_j) and overestimation (β_j) parameters for each KPI_j in the penalty function as assigned by the process expert.....86

Table 4.6 - R-squared and Root Mean Square Error (RMSE) values of each algorithm in BLRB dataset.....88

Table 4.7 - Total average penalty scores for each algorithm in BLRB dataset.....88

Table 5.1 - Calculation of point coordinates $\overrightarrow{X_j^1}$ on the sides of the polygon for a numeric observation with six variables shown in Figure 5.1, where $\hat{q}$ and $\hat{l}$ are the unit vectors of x and y directions, respectively. 124

Table 5.2 - Calculation of point coordinates $\overrightarrow{Y_h^1}$ on polygon sides for a numeric observation with four outputs shown in Figure 5.2, where $\hat{q}$ and $\hat{l}$ are the unit vectors of the x and y directions, respectively.....125

Table 5.3 - Examples of manipulated, measured variables and KPIs for the concentrator equipment 136

Table 5.4 - Range of hyperparameters of each time-series predictor using the concentrator equipment.....140

Table 5.5 - Underestimation (α_j) and overestimation (β_j) parameters for each KPI_j in the penalty function as defined by the process expert 141

Table 5.6 - R-squared and root mean square error values of each algorithm in the concentrator dataset.....142

Table 5.7 - Total average penalty scores for each algorithm in the concentrator dataset 142

LIST OF FIGURES

Figure 1.1 - Illustration of published and submitted articles incorporated in this thesis 10

Figure 3.1 Diagram of the proposed method: Polygon Generation and Deep Learning for Classification, *the preprocessing stage for the testing dataset is the same as the training dataset, except for the replacement of the training data standardization with the testing data standardization 25

Figure 3.2 – (a) Each polygon side represents a data variable where Z_{kj} is the standardized scalar value of variable X_j for observation k (b) $\widehat{X_j}$ is the unit vector of each polygon side and $\overrightarrow{X_j}$ represents the zero value of each variable as a point vector and $\overrightarrow{X_j^k}$ is the point vector of the variable value for the observation k 26

Figure 3.3 - Polygon (square) of a simple numeric dataset of four variables: the iris dataset example 27

Figure 3.4 - A four-sided polygon with points on its sides that represent the observation values after standardization: the iris dataset example 28

Figure 3.5 - Two polygons (images) for one observation with two different Hamiltonian cycles: the iris dataset example 30

Figure 3.6 - Case Study 1: Tennessee Eastman Process (TEP) (Downs & Vogel, 1993).....32

Figure 3.7 - Case Study 2: Reboiler system of Heat Recovery Network (HRN) in TMP mills 34

Figure 3.8 - The main steps of the proposed methodology applied to the two fault classification case studies 35

Figure 4.1 - A polygon generated from a numeric observation of five data variables using the method proposed in (Elhefnawy et al., 2021a), where $\overrightarrow{X_j^k}$ represents the point coordinates of standardized values of observation k for each variable X_j, $\overrightarrow{X_j}$ represents the point coordinates of the zero standardized value of the variable X_j and $\widehat{X_j}$ represents the unit vector of each polygon side. All variables are numbered in a clockwise direction. 71

Figure 4.2 - A polygon generated from a numeric observation of three outputs using the method proposed in (Elhefnawy et al., 2021a), where $\overrightarrow{Y_h^k}$ represents the point coordinates of standardized values of observation k for each output Y_h, $\overrightarrow{Y_h}$ represents the point coordinates of the zero standardized value of the output Y_h and $\widehat{Y_h}$ represents the unit vector of each polygon side. All outputs are numbered in clockwise direction. 72

Figure 4.3 - A simplified schematic of Generative Adversarial Network (GAN) 74

Figure 4.4 - A schematic diagram of the cGAN to translate satellite images into Google Map images 77

Figure 4.5 - A detailed schematic diagram of the proposed method (PGcGAN) 79

Figure 4.6 - Training of the cGAN model to convert polygon images representing input variables into another set of polygon images representing the numerical outputs 80

Figure 4.7 - A schematic diagram summarizes the image processing steps for mapping polygon images into numerical output values 80

Figure 4.8 - A schematic diagram of the Black Liquor Recovery Boiler (BLRB) (Vakkilainen & others, 2005) 81

Figure 4.9 - The U-Net architecture: Generator of Pix2Pix model (Ronneberger et al., 2015).....84

Figure 4.10 - PatchGAN: The discriminator of Pix2Pix model (Demir & Unal, 2018) 84

Figure 4.11 - Prediction of KPI_1 (steam production divided by black liquor flow (SP/BLF)) using the proposed method (PGcGAN) and other ML regression models (a) cGAN (b) MLP (c) DT (d) RF 89

Figure 4.12 - Prediction of KPI_2 (emitted sulphur dioxide (SO_2)) using the proposed method (PGcGAN) and other ML regression models (a) cGAN (b) MLP (c) DT (d) RF.....90

Figure 4.13 - Prediction of KPI_3 (total reduced sulphide (TRS)), the proposed method (PGcGAN) and other ML regression models (a) cGAN (b) MLP (c) DT (d) RF.....91

Figure 4.14 - A schematic diagram summarizing the proposed method (PGcGAN) for predicting KPIs in industrial plants 92

Figure 5.1 - A polygon generated from a numeric observation of six data variables using the method proposed in (Elhefnawy et al., 2021a). All variables are numbered in a clockwise direction. 124

Figure 5.2 - A polygon generated from a numeric observation of the four outputs. All outputs are numbered in clockwise direction.....125

Figure 5.3 - A simplified schematic of (a) unconditional GAN and (b) conditional GAN. Note: Both unconditional and conditional GANs can be applied to several data types such as images, video, text, etc. 127

Figure 5.4 - A schematic diagram of the 3D-CycleGAN to translate camera videos into segmented videos (Adopted from (Bashkirova et al., 2018)).....128

Figure 5.5 - A schematic diagram of the proposed method 129

Figure 5.6 - The change of polygon images over time in the form of a polygon stream through an example of three outputs 131

Figure 5.7 - Training of the 3D-CycleGAN model to obtain the generator $G_{I/O}$ that converts polygon streams representing time-series input variables into another set of polygon streams representing the time-series outputs.....133

Figure 5.8 - A schematic diagram summarizes the image processing steps for mapping every frame in polygon streams into time-series outputs.....134

Figure 5.9 - A schematic diagram of the concentrator equipment in the HRN.....136

Figure 5.10 - The generator architecture in the 3D-CycleGAN.....138

Figure 5.11 - PatchGAN: The discriminator of the 3D-CycleGAN 138

Figure 5.12 - Prediction of KPI_2 (Concentrator efficiency) using the proposed method (PG + 3DCycleGAN) and other prediction models (a) LSTM (b) RNN (c) 1D-CNN.....143

LIST OF ABBREVIATIONS AND NOTATIONS

Abbreviations

AEM	Abnormal event management
AI	Artificial intelligence
ANFIS	Adaptive neuro-fuzzy inference system
ANN	Artificial neural network
ARIMA	Autoregressive integrated moving average
ATSW	Adaptive time-series window
BL	Black liquor
BLRB	Black liquor recovery boiler
cGAN	Conditional generative adversarial networks
CNN	Convolutional neural network
DBN	Deep belief network
DEA	Data envelopment analysis
DL	Deep learning
DRIP	Data-rich-but-information-poor
DT	Decision tree
FAR	False alarm rate
GAN	Generative adversarial networks
GHG	Greenhouse gas
GPU	Graphical processing unit
GBR	Gradient boosting regression
GRU	Gated recurrent unit
HRN	Heat recovery network

HPC	High performance computing
HWES	Holt Winter's exponential smoothing
ICA	Independent component analysis
ICT	Information and communication technology
IoT	Internet of things
KPI	Key performance indicator
k-NN	k-nearest neighbor
LFE	Large final emitter
LP	Low pressure
LSTM	Long short-term memory
ML	Machine learning
MLP	Multi-layer perceptron
NAS	Neural architecture search
PAA	Past accidents analysis
PCA	Principal component analysis
PCP	Parallel coordinate plots
PG	Polygon generation
PHA	Process hazard analysis
QDA	Quadratic discriminant analysis
RBFNN	Radial basis function neural network
ReLU	Rectified linear unit
RF	Random forest
RMSE	Root mean square error
RNN	Recurrent neural network

RBFNN	Radial basis function neural network
SES	Simple exponential smoothing
SFA	Stochastic frontier analysis
SLMLP	Supervised local multi-layer perceptron
SVM	Support vector machine
SVR	Support vector regression
TEP	Tennessee Eastman process
TMP	Thermomechanical pulp
TRS	Total reduced sulphide

Notations

$\boldsymbol{X_j}$	j^{th} input variable
$\boldsymbol{Y_h}$	h^{th} output
$\hat{\boldsymbol{q}}, \hat{\boldsymbol{l}}$	The unit vector of x and y directions respectively
$\overline{\boldsymbol{X_j}}$	Mean of jth input variable
$\overline{\boldsymbol{Y_h}}$	Mean of hth output
$\boldsymbol{\delta_j}$	Standard deviation of jth input variable
$\boldsymbol{\delta_h}$	Standard deviation of hth output
$\boldsymbol{x_{kj}}$	Value of jth input variable for k^{th} observation
$\boldsymbol{y_{kh}}$	Value of hth output for k^{th} observation
$\boldsymbol{Z_{kj}}$	Standardized value of jth input variable for k^{th} observation
$\boldsymbol{Z_{kh}}$	Standardized value of hth output for k^{th} observation
$\widehat{\boldsymbol{X_j}}$	Unit vector of the polygon side representing j^{th} input variable
$\widehat{\boldsymbol{Y_h}}$	Unit vector of the polygon side representing h^{th} output

$\overrightarrow{X_j}$	Point coordinates of the zero standardized value of the variable X_j
$\overrightarrow{Y_h}$	Point coordinates of the zero standardized value of the output Y_h
$\overrightarrow{X_j^k}$	Point coordinates of the standardized values of kth observation for variable X_j
$\overrightarrow{Y_h^k}$	Point coordinates of the standardized values of kth observation for output Y_h
$G_{B/A}$	Generator of the GAN architecture translating observations from domain B to domain A
$G_{A/B}$	Generator of the GAN architecture translating observations from domain A to domain B
D_A	Discriminator of the GAN architecture for observations generated from domain A
D_B	Discriminator of the GAN architecture for observations generated from domain B
L_{GAN}	Adversarial loss of the GAN architecture
L_{cyc}	Cycle consistency loss of the GAN architecture
$\mathcal{L}$	Total loss of the GAN architecture
Q	Universal image quality index
R^2	R-squared values
SP/BLF	Steam production divided by black liquor flow
$\bar{x}$	Mean of the pixel values of the frames of the polygon stream representing the outputs
$\bar{y}$	Mean of the pixel values of the frames of the synthesized polygon stream using the GAN model
σ_x	Standard deviation of the pixel values of the frames of the polygon stream representing the outputs
σ_y	Standard deviation of the pixel values of the frames of the synthesized polygon stream using the GAN model
α_j	The underestimation parameter of each KPI_j in the penalty function

$\boldsymbol{\beta_j}$	The overestimation parameter of each KPI_j in the penalty function
$\boldsymbol{L^j(t_k)}$	The penalty function of each KPI_j at instance t_k
$\boldsymbol{L^j_{AP}}$	The average penalty function of each KPI_j
$\boldsymbol{L_{model}}$	The total average penalty function of each time-series regression model

CHAPTER 1 INTRODUCTION

Although process industries provide services and products that are of paramount importance to modern human life, their operation requires massive amounts of energy, which are mainly generated by the consumption of natural resources, leading to the emission of harmful chemicals and gases. Examples of these industries are oil & gas, pulp & paper, cement, steel & iron, and chemical processes. Energy efficiency in these industries is a major and prime concern that impacts the environment and economy. Among the reasons for energy inefficiency in such systems is the improper monitoring and control of the operation of these industrial sites. This leads to deteriorated system performance due to high-frequency abnormal situations/faults, poor adaptability to system disturbances, downtime, and excessive maintenance activities in addition to human errors. The process industries are using an approach called past accidents analysis in which causes of abnormalities along with their contributing factors are analyzed (Athar et al., 2019). It was reported in (Arendt & Lorenzo, 2010; FP, 1996; Kourniotis et al., 2000) that equipment failure, operational failure, and human factors are among the highest contributing factors of these abnormalities. The difficulties and challenges encountered in the diagnosis of abnormalities and prediction of future failures in the complex industrial processes are mainly attributed to the complexity of such processes with thousands of variables (Ge, 2017). Several malfunctions or faults of industrial plants occur due to unanticipated dynamic interactions between their units which results in different sources of uncertainty during the operation of these units. Moreover, the process industry still heavily relies on human expertise to optimize the design and operation of equipment and units in these processes. This expertise is expensive, rare, and limited due to the increasing system complexity that hinders knowledge acquisition and process optimization.

Therefore, there is a great potential to improve the monitoring of these plants and eliminate any possible risk that adversely affects their operation. A potential solution for most of these difficulties and challenges is to exploit the advancement of digital technologies found in industry to build an efficient process hazard analysis (PHA) system to save the health and safety of humans and reduce any related economic losses. The current PHA systems are mainly based on physical models that are built for specific equipment in the plant by using the first principles and high-level involvement of human expertise (Kletz, 1988). However, developing a PHA for the whole industrial plant is a challenging task. Fortunately, the industrial processes are equipped with

numerous amount of sensing devices resulting in a massive amount of collected data. This collected data in each industrial process represents an important asset and valuable source of knowledge. Exploiting different data sources collected in the plant can be the keystone to building a systematic PHA system that accurately diagnoses abnormalities and achieves an energy-efficient process with a reduced environmental footprint as a step towards operational excellence.

A big challenge in the processing of industrial data is that they are acquired from heterogeneous sources such as temperature sensors, pressure sensors, cameras, LiDAR sensors, IoT, etc. Moreover, these data are stored as different types (text, images, video, sound, point cloud, etc.) in a disconnected way, which imposes several challenges to analyze them. One of the challenges is the storage of these data in isolated silos located in different departments and units in the plants. These disconnected silos hold raw data that is hard to be accessed and handled by data analytics tools. The question is how the industry can leverage these disconnected data silos and maximize their global value? Therefore, there are prioritized needs for many industries to clean and process these data in a way that makes them easily accessible, connected, and fully exploitable.

In response to these needs, efficient data fusion methods are required to blend different sources & types of data at different levels; data level, information level, and decision level. Data fusion aims at extracting useful knowledge by merging the available data for the purpose of greater quality, where the term “greater quality” differs according to the addressed industrial application (Wald, 1999).

In order to extract the desired knowledge from these heterogeneous data sources with minimal human effort, here come the role of artificial intelligence (AI) algorithms for the automation of data fusion. With the aid of AI, one can develop intelligent fusion methods that provide accurate models representing complex interactions and phenomena in the monitored process. Moreover, these models can ensure the connectivity between different departments in the enterprise, thus constructing a complete picture for the performance of the overall plant.

1.1 Problem statement

Given the numerous amount of heterogeneous data in industrial processes and the available high computational infrastructure, deep learning (DL) can be used as an efficient AI tool for

building more accurate and representative models compared to other classical analytical techniques (LeCun et al., 2015; Lv et al., 2016; Goodfellow, Bengio, 2017). It offers an important opportunity to develop robust data-driven models for several decision-making applications (Rolnick et al., 2019). Given the complexity, nonlinearity, and non-stationarity of the industrial processes and the interaction between the process variables, DL can be used as an end-to-end learning predictive technique where manual feature engineering is no longer needed. Moreover, DL can be used as a generative modeling technique (e.g. generative adversarial networks; GAN) to capture the true data distribution in complex industrial processes.

In this thesis, two main challenges are addressed upon dealing with the fusion of data of different types.

Challenge 1: Efficient representation of the available data for the best model performance. The first challenge is related to data representation. The adoption of AI in process industries is linked with the success of the entire AI life cycle and addressing its needs (Gärtler et al., 2021). One of the main components of the AI life cycle is data preparation and representation. Since data acts as a fuel for the DL architectures in the modeling process, the quality of the prepared data significantly impacts the overall quality of the final DL models. Therefore, this data needs to be represented efficiently to maximally exploit the DL modeling capability (Bengio et al., 2013), thus maximizing the value of data and contributing to the success of the entire AI life cycle. Most industrial processes suffer from ill-defined data distributions that are hard to capture without an efficient representation. In these industrial applications, the collected data especially in continuous industrial processes are often characterized as data-rich-but-information-poor (DRIP) which makes the data-driven modeling approach more challenging (Gärtler et al., 2021). Fortunately, data-centric AI (Andrew Ng Launches A Campaign For Data-Centric AI, 2021.; Wu, 2021) is emerging nowadays to focus more on representing the available data given the sophisticated DL techniques, but low-quality data.

Challenge 2: Selecting the most suitable DL technique according to the data type, quantity, and the addressed problem. It is hard to capture the whole picture of the industrial process using only one data source. Every data type gives a complementary insight into the operation of the process. Accordingly, it is a prioritized need to maximize the value of each data source through efficient and accurate modeling. There is a wide spectrum of problems that exploit

the industrial data including classification, regression, time-series prediction, etc. The DL algorithms can be used to develop discriminative and generative modeling approaches depending on the types and amount of data and the addressed application. Therefore, the selection of the right DL algorithms is a significant task to efficiently tackle each data-driven problem. It is well-known among the researchers and practitioners of the AI community that each data type has different nature and structure where it can be used to train different DL techniques for the best model performance and least computational demand. One of the most important needs in the industry is the accurate prediction of the system performance for better monitoring and control, which is the main task of this thesis work. The key is to select the right DL method and make use of its modeling power given the available heterogeneous data to satisfy industrial needs. Diversifying of DL methods allows their adoption in processing different types of data depending on the final objective. Using an ensemble of DL algorithms is one of the main characteristics of this doctoral research as a step towards efficient data fusion for enhancing the performance of industrial systems.

1.2 Performance prediction of industrial systems: Limitations of existing methods

Despite the existence of several classical techniques for predicting industrial system performance, they have some limitations that hinder their adoption in several industrial applications. The main limitations are listed according to the perspective of researchers and practitioners as follows:

- Manual feature engineering including feature selection and extraction consumes time and requires effort by the human expert before using classical modeling techniques. There exist several data representation and visualization tools that can help in the preparation of high-dimensional industrial data. One of these tools commonly used in industry is the parallel coordinate plot (PCP) (Inselberg, 2009). Although the experts can extract useful patterns from PCP, there are some limitations, mainly related to the ordering of variables that can significantly change its representation in addition to the manual extraction of the patterns which is challenging and time-consuming. These limitations hinder the extraction of all the representative patterns.

- Existing modeling approaches mainly focus on the optimization of the ML and DL architectures regardless of the quality of the exploited data and its representation. Given that the data is the fuel of various AI techniques, there is an urgent need to improve the data quality and empower its representation to significantly enhance the model's prediction performance using the same architecture. This can impact the whole MLOps cycle in which the data preparation phase is quite ignored.
- Most classical analytical and ML methods assume impractical assumptions about the distribution of the industrial data. They still rely on the human expert who has significant knowledge about the industrial operation, however, it is hard for the expert to efficiently address a complex operation given its non-linearity, non-stationarity, and interactions between system components.
- Existing data fusion methods focus on merging data at the information (feature) and knowledge (decision) levels. This can result in a loss of information content in the original raw data. Accordingly, incorporating the data fusion at the raw level would maximize the value of the heterogeneous data and better leverage its knowledge content for accurate system performance prediction.

1.3 Deep learning opportunities for accurate prediction of system performance

The DL approach has some important advantages over the other comparable ML and classical analytical techniques in terms of the prediction of the system performance (LeCun et al., 2015; Lv et al., 2016; Goodfellow, Bengio, 2017). The main advantages are listed as follows:

- **End-to-end learning technique**: Manual feature engineering is a tedious process done by the process expert. Fortunately, in the case of training deep neural network architectures, the feature extraction is implicitly embedded within the model architecture at different levels after each layer. Therefore, the expert saves most of the effort and time spent in the data preparation stage including feature selection and extraction (LeCun et al., 2015).
- **Capturing high nonlinearity and dynamic behavior of complex industrial equipment**: Given the flexibility of network architectures and its adaptation capability with different data

structures, DL can efficiently model non-stationary data distribution in a robust and accurate way which is an urgent need for industrial systems (Ian Goodfellow Yoshua Bengio, 2017).

- **Unprecedented performance especially in the computer vision field**: DL techniques, in particular CNNs and GANs, achieved a breakthrough in computer vision such as video and image classification, video and image segmentation, transferring the images and videos from one domain to another, and improving the image and video resolution (M. Y. Liu et al., 2021).

This thesis work makes use of these opportunities and advantages through developing a powerful visual data representation with information-rich input that helps improve the prediction of the system performance through the exploitation of DL predictive and generative modeling capabilities. The aim is to improve industrial data quality through efficient representation with the aid of high computational infrastructure. As a result, temporal and spatial patterns can be extracted from the data when an accurate data fusion model can be built for extracting and aggregating the knowledge found in different sources of data.

1.4 Research Objectives

1.4.1 General objective

The ultimate objective of this doctoral research is to facilitate the fusion of heterogeneous industrial data through diversified DL techniques according to the data types, structures, and formats and the addressed applications. This can help develop deployable platforms that can clean, merge and process data collected from heterogeneous sources and provide the user with accurate knowledge for decision making.

In this doctoral research, different methods are developed by combining both DL techniques and a novel powerful data representation for solving different modeling problems. These developed methods are applied to different data types: numerical sensory data, images, and videos. The data collected from the historian of different industrial equipment is the fuel of these methods. These novel methods will be discussed in the subsequent chapters of this thesis.

1.4.2 Specific research objectives

The above general objective is achieved by fulfilling the following specific objectives:

1. Implementation of a novel and efficient data representation method that can better exploit the numerical data available in industrial systems. The output of this method is used to train DL architecture for predicting the class of the unseen numerical data acquired from the targeted processes. In the industrial context, this method can be used for different classification problems such as fault diagnosis in complex equipment.
2. Implementation of a multi-output regression method using the developed data representation technique and generative DL. The DL is used to translate the input data into the numeric outputs without making impractical assumptions on data distribution. The developed regression method can be used for predicting multiple KPIs in industrial equipment using the collected sensory data.
3. Implementation of a time-series prediction method using the developed data representation technique and recurrent DL techniques. Unsupervised DL techniques are taken into consideration to save the effort of the process expert in labeling the industrial data. The developed method can capture the data distribution of highly non-linear dynamic industrial processes and properly predict the time-series KPIs based on input streaming data.

1.5 Originality and Success

The novel data representation technique called “Polygon Generation” proposed in this doctoral research is inspired by the concept of PCP (Inselberg & Dimsdale, 1990), which was introduced as a powerful data visualization and pattern recognition tool in the process industry (Choi et al., 2009; Dunia et al., 2013). As mentioned previously, there are some limitations to applying PCP in industry, mainly represented in the time-consuming manual extraction of patterns and the change in the extracted pattern upon changing the order of data variables. Polygon generation is based on the Hamiltonian cycles that can transform the data into graph representations (polygon images) that systematically represent all interrelationships between the data variables. This helps facilitate the process of pattern extraction and recognition through these representative polygon images.

The main finding in this doctoral research is that polygon generation as a data representation technique along with DL as a powerful predictive and generative modeling approach with the aid of high computational infrastructure can improve the monitoring and control of various industrial systems, thus improving their performance. The developed novel methods can be used as cornerstones for efficient industrial data fusion to provide the user with the maximum value of knowledge for better decision-making. The novelty of this doctoral research is stated in two points (1) the implementation of a data representation method to improve the quality of the collected industrial data (2) the application of the DL predictive and generative approaches in the context of the performance evaluation of the industrial equipment. To the best of our knowledge, the material presented in this thesis is not published or written elsewhere except where due references are cited.

The developed methods proved their successful performance through different use cases in the process industry (namely; recovery boiler, reboiler, and concentrator equipment) in Kraft and thermomechanical pulp and paper mills located in Canada. The obtained results are promising and indicate that these developed methods can be successfully applied in other industrial applications.

1.6 Scientific Contributions and Deliverables

The research conducted in this thesis was followed in an evolutionary manner to achieve the specific objectives stated previously. The scientific contributions of this doctoral research can be summarized as follows:

- The first contribution introduces an innovative data representation method that converts numeric data into representative graphs (polygons). The polygon generation and predictive deep learning were applied successfully as an accurate classification method.
- The second contribution introduces an accurate and representative multi-output regression method based on polygon generation and image-to-image translation. The idea is to better express all interrelationships between data inputs and outputs (KPIs).
- The third contribution introduces a novel time-series prediction method based on polygon generation and unsupervised video-to-video translation to deal with data streams that are commonly available in industrial systems.

The proposed methods in these contributions have an embedded feature extraction capability that allows “end-to-end learning”. They were tested successfully on several case studies based on real and simulated data. The methods constitute a generic approach based on polygon generation and deep predictive & generative modeling and can be applied in many industrial problems.

The outcomes of these contributions are:

1. Three peer-reviewed journal articles.
2. A thesis compiling the findings, including the three articles.
3. A related software was developed based on different codes mainly implemented using open-source packages.

This doctoral research is documented in the three articles incorporated into this thesis. The first article addresses the development of the data representation technique and using it as a classification method. The second one addresses the multi-output regression problem, while the third addresses the time-series prediction. The first article in this thesis has been approved and published in the “Journal of Intelligent Manufacturing, Springer”. The second article has been submitted to the journal of “Expert Systems with Applications, Elsevier”. The third one has been submitted to the journal of “Computers in Industry, Elsevier”. Figure 1.1 illustrates the evolutionary approach followed in this doctoral research along with the three articles involved.

1.7 Potential Impacts: Social and Economic Benefits

An initial outcome of this doctoral research is the obtained results that were published and submitted in recognized international journals. The aim is to share our insights and experience with a wide spectrum of researchers and practitioners who are experiencing hardships to deal with challenging industrial data. This will also enforce the link between both academic and industrial communities.

As mentioned previously, the Canadian industries including pulp and paper mills, the oil and gas industry, etc. use excessive energy resources and contribute significantly to the GHG emissions. Accurate prediction performance in such industries can result in efficient monitoring and control in these industries that would have positive impacts on Canada, both socially and

economically. This doctoral research targets the development of an advanced decision support tool that exploits heterogeneous data sources and improves the prediction accuracy in the Canadian supply chains in various industries, thus efficiently monitoring these industries. In addition to these economic benefits, this can help tackle the problem of climate change which is a big challenge worldwide. Helping the Canadian industries to operate in an energy-efficient way will increase their competitiveness, thus maximizing the benefits of Canada's natural resources and significantly contributing to enhanced prosperity for Canadians. We are planning to extend this research in the future to the interpretation of the DL and extracting explainable rules which is one of the interests of our research team at Polytechnique Montréal.

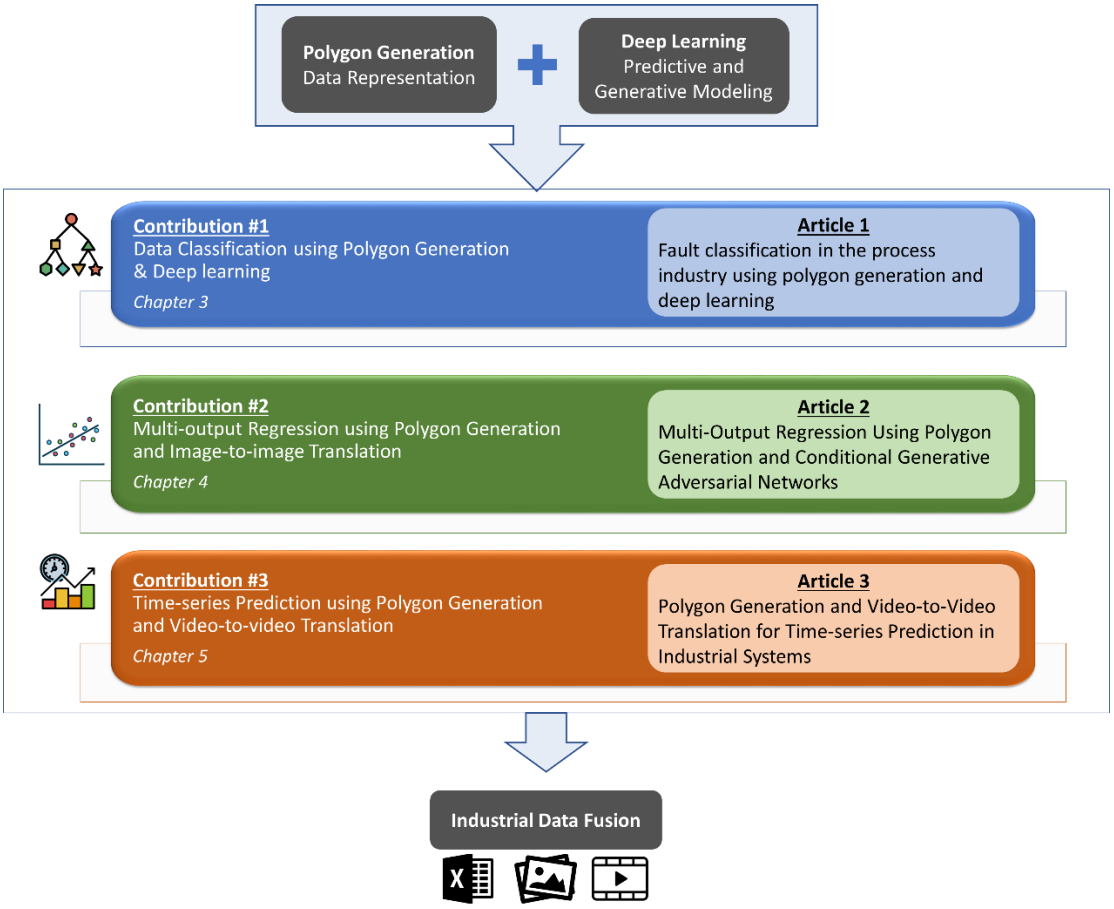


Figure 1.1 - Illustration of published and submitted articles incorporated in this thesis

1.8 Thesis organization

This thesis is divided into six chapters. Chapter 1 explains the problem being studied and the challenges tackled in this thesis, and it also states the main and specific objectives of this research and explains its originality, its potential social and economic impacts, and the targeted deliverables.

Chapter 2 includes background about the existing data fusion techniques at different levels. The purpose is to give an overview and a preliminary number of references for the fusion methods, their limitations along with recommendations and potential research directions.

Chapters 3 through 5 present the three articles incorporated into this thesis. They include the developed methods used throughout this doctoral research.

Chapter 3 introduces a novel data representation technique (polygon generation) that maps the numerical observations into images of polygons. These polygons represent all interrelationships between the data variables in a systematic way based on Hamiltonian cycles. They are used for training DL architecture for building robust and accurate classifiers. The proposed method was applied on a simulated process (Tennessee Eastman process) in addition to real industrial data collected from reboiler system in a heat recovery network (HRN) of a thermomechanical pulp (TMP) mill located in Canada.

Chapter 4 presents a novel multi-output regression method based on polygon generation and DL. The polygon generation technique is separately applied to both input variables and outputs to synthesize two different sets of polygon images. Then, an image-to-image translation technique based on generative adversarial networks (GANs) is used for mapping the input polygons into the output ones. The developed regression method was applied to a black liquor recovery boiler (BLRB) in a pulp mill in Canada.

Chapter 5 presents a novel time-series prediction method based on video-to-video translation and polygon generation. The highly dynamic behavior of industrial systems is represented in the form of polygon streams (videos). The polygon videos representing the time-series inputs are translated into another set of polygon videos that represent the time-series outputs (KPIs) in an unsupervised way using 3D cycle-consistent GANs. This proposed time-series prediction method was applied on a black liquor concentrator in a Kraft pulp mill located in Canada.

Chapter 6 presents a general discussion on the three methods incorporated in this thesis, followed by conclusions and future work directions to extend this doctoral research

CHAPTER 2 RELATED WORK, CHALLENGES & RESEARCH DIRECTIONS

AI and related digital technologies have a significant impact on the automation of many industrial processes and making machines smarter. This could not happen without the advancement in information and communication technology (ICT). Advanced communication systems in addition to the progress in Big data & Internet of Things (IoT) are major tools that motivate the industrial processes to satisfy the needs of enterprises, customers, and society (Lasi et al., 2014). Moreover, the availability of various data sources such as specialized Web databases (Barrett et al., 2012; Hendrickx et al., 2014; Karagkouni et al., 2017), ontologies (Y. He et al., 2009; Olsina & Martin, 2004; Simmonds et al., 2004) and digital libraries (McEntyre & Lipman, 2001) in addition to an extra amount of data either generated by humans or machines greatly enhances the chance to improve the extraction of useful knowledge in many industrial applications (Mandreoli & Montangero, 2019). Unfortunately, the analysis of complex phenomena in the industry cannot be achieved using only a single type of datasets (Gomez-Cabrero et al., 2014). As a result, there is an urgent need to fuse or blend different types of data in an integrated and unified manner. The aim is to make the available data cleaned, connected, and fully exploitable. This can help construct a complete picture of the performance of the targeted system's operation.

2.1 Overview on Data Fusion

Data fusion can be defined in different ways according to the degree of generality and the research area. A general definition of data fusion is stated in (White, 1991) as "*A process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats as well as their significance*". Another definition with more generality and flexibility is stated in (Wald, 1999) as "*Data fusion is a formal framework in which means and tools are expressed for the alliance of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of 'greater quality' will depend upon the application*". This definition emphasizes the effect of data fusion on quality. The authors in (Lahat et al., 2015) focus on the importance of data fusion by asking the question on

"how to exploit diversity?" and how to make use of this opportunity to enhance the extracted knowledge and benefits in a way that cannot be achieved using single dataset. Accordingly, they adopt a definition for data fusion as: "it is the analysis of several datasets such that the datasets can interact and inform each other" (Lahat et al., 2015).

Data fusion is a challenging and essential endeavor that recently attracts interest in different disciplines such as remote sensing and multisensory (Castanedo, 2013; Khaleghi et al., 2013), IoT & Big data (Alam et al., 2017; Camacho et al., 2016) owing to the enormous growth of available data and the motivation to model complex systems via data-driven approaches (Cocchi, 2019; Kitchin, 2014; Martens, 2015).

2.2 Challenges of Existing Data Fusion Methods

This section summarizes the challenges of the existing fusion methods. We refer to comprehensive review papers and books on the topic of how these fusion methods are developed along with their mathematical infrastructures. The aim is to give an overview of the limitations of these methods and identify the knowledge gap in the literature.

The existing data fusion methods suffer from several issues and challenges at each level (raw, information, and knowledge levels). These review papers (Alam et al., 2017; Gärtler et al., 2021; Gheisari et al., 2017; Himeur et al., 2020; Lau et al., 2019; Martin-Lopo et al., 2020; Meng et al., 2020; Uysal & Sogut, 2017; Qingchen Zhang et al., 2018; Zitnik et al., 2019) compile and analyze most of these challenges. The main challenges are summarized as follows:

- **Low data quality**: Most of the data acquired from different data sources in industrial systems are imprecise, incomplete, vague, and uncertain. Since, "garbage in, garbage out" theorem applies in data processing, this imperfection significantly impacts the data quality, thus the performance of the data fusion methods. Data veracity needs to be ensured to overcome data incompleteness and incorrectness and to obtain representative and informative data. Moreover, some sensors can be regarded as less significant than others or even unworthy, accordingly, data acquisition is a critical process that needs human expertise and carefulness (Lau et al., 2019; P. Li et al., 2020; Mart\'\in-Lopo et al., 2020; Sakpal, 2021; Sridharan, 2017).

- **Data heterogeneity**: Data are collected from heterogeneous data sources in industrial processes, which imposes challenges on their fusion in terms of different structures, types, distributions, and sampling frequencies. Moreover, another challenge is the incompatibility of different data formats due to the absence of its standardization. All these data sources need to be efficiently exploited for capturing the whole picture of the monitored process from different perspectives (Himeur et al., 2020; Meng et al., 2020).
- **Isolated data sources**: Another challenge is that the industrial data exist in isolated silos located in different locations, thus, they are disconnected. Moreover, data governance and privacy have imposed more constraints, as the misuse of private and sensitive information in industrial processes may lead to catastrophic events such as information theft and identity fraud. It is hard to find industrial datasets in open repositories which limits the application of advanced data processing techniques through the AI community in industrial processes (Gärtler et al., 2021; Gheisari et al., 2017; Petzold et al., 2020).
- **Unannotated and misaligned data**: Errors and uncertainties in the industrial data collection make it challenging to obtain the ground truth of the collected data. Hence, annotating the data and obtaining correct labels are main problems that hinder the exploitation of these data sources. Manual annotation is applicable for small datasets but nearly impossible and exhausting in the case of large datasets in most industrial systems. This can hinder the proper application of supervised learning algorithms. Moreover, any misalignment can drastically impact the fusion model performance. Data captured from different sensors have to be correctly aligned for a proper data fusion process (Alam et al., 2017; Meng et al., 2020).
- **Dynamic and inconsistent data**: Data in industrial processes are characterized by their non-linearity and high dynamic behavior. The data distribution is subject to continuous changes resulting in data with non-stationary nature. Data non-stationarity imposes challenges for accurate real-time monitoring. Moreover, most industrial data suffer from different noise sources due to sensor imprecision and other external factors such as the environment and human errors (Alam et al., 2017; Meng et al., 2020).

2.3 Research Directions & Recommendations to Overcome Data Fusion Challenges

According to several recommendations in the literature and readings, some potential research directions have been suggested to overcome the previously mentioned challenges. The main recommendations are summarized as follows:

- Improving the data representation regardless of the modeling techniques will significantly enhance the data quality. Using deep representation learning can help leverage the information hidden in the available data, thus representing the complex and interacting phenomena within the process. An efficient data representation can be a building block for explaining the deployed data-driven models (Zitnik et al., 2019).
- Using state-of-art learning techniques to model complex industrial processes with ill-defined distributions and high-dimensional data. As a promising learning technique, DL has achieved a breakthrough compared to the past analytical and conventional ML algorithms in terms of improved performance and prediction precision. Using diversified DL techniques can provide higher fusion fidelity and facilitate the fusion process (Alam et al., 2017; Gheisari et al., 2017; Himeur et al., 2020; Meng et al., 2020; Petzold et al., 2020; Qingchen Zhang et al., 2018).
- Using cloud computing and high computational infrastructure for facilitating data processing and analysis. The cloud infrastructure would be the default option for data management, processing, and modeling. This will facilitate the exploitation of a large amount of imagery data and videos that are complementary sources of knowledge to better represent the industrial system dynamics (Himeur et al., 2020; Martin-Lopo, Boal, & Sánchez-Miralles, 2020; Uysal & Sogut, 2017).

2.4 Proposed Approach: Polygon Generation and Deep Learning Towards Data Fusion

The proposed approach comprises three different methods to address the challenges of the current data fusion techniques and the limitations of the model-based approaches in different

industrial applications. These methods follow the recommendations and make use of the breakthrough of DL and its capability to encounter ill-defined distributions in the process industry. The proposed methods focus on the data representation and how all interrelationships between data variables can be expressed in a systematic way, thus maximizing the global value of the industrial data. Moreover, these methods open the door for the fusion of different data sources in terms of format and structure for the sake of more robust and accurate models.

The proposed approach in this thesis is divided into three main parts. The first part introduces the polygon generation approach and how it can use the state-of-art DL architectures for accurate classification models in industrial processes. This classification method is validated through a benchmark dataset in the process industry and also tested successfully using a real industrial dataset collected from a reboiler system in a thermomechanical pulp mill. This work is published in (Elhefnawy et al., 2021b)

The second part concerns the adaptation of the polygon generation approach to be used for regression problems not only classification ones. For this purpose, generative modeling plays an important role in the translation of polygon images representing inputs into other polygon images representing the output values. This approach was tested successfully using a complex industrial dataset acquired from a black liquor recovery boiler (BLRB) in a pulp & paper mill.

The last part concerns dealing with time-series prediction using the polygon generation approach and generative DL. After successful testing in different scenarios of supervised learning (classification and regression) using different datasets with the assumption of independent and identically distributed (i.i.d) variables, this part deals with the time-series datasets as polygon streams. The generative DL technique can translate a set of polygon streams representing time-series inputs into other corresponding polygon streams representing time-series outputs. The proposed method does not need the tedious labeling procedures done by the process expert as it works in an unsupervised way. Moreover, the model's trustworthiness is encountered using a metric for measuring the quality of synthesized streams. This approach was tested successfully using a complex dataset from concentrator equipment in Kraft pulp and paper mills.

CHAPTER 3 ARTICLE 1: FAULT CLASSIFICATION IN THE PROCESS INDUSTRY USING POLYGON GENERATION AND DEEP LEARNING

Mohamed Tarek Mohamed Elhefnawy, Ahmed Ragab, Mohamed-Salah Ouali

Published in: JOURNAL OF INTELLIGENT MANUFACTURING

DOI 10.1007/s10845-021-01742-x

3.1 Abstract

This paper proposes a novel data preprocessing method that converts numeric data into representative graphs (polygons) expressing all of the relationships between data variables in a systematic way based on Hamiltonian cycles. The advantage of the proposed method is that it has an embedded feature extraction capability in which each generated polygon depicts a class-specific representation in the data, thereby supporting accurate "end-to-end learning" in industrial fault classification applications. Moreover, the generated polygons can play a significant role in the interpretation of trained deep learning fault classifiers. The performance of the proposed method was demonstrated using a benchmark dataset in the process industry. It was also tested successfully to classify challenging faults in major equipment in a thermomechanical pulp mill located in Canada. The results of the proposed method show better performance than other comparable fault classifiers.

3.2 Introduction

Process industries are among the industrial systems that are characterized as the largest greenhouse gas (GHG) emitters according to (Bush et al., 2019). Oil & gas refineries, pulp & paper mills, iron & steel mills, cement mills and chemical processes are common examples of these large final emitters (LFEs) (Talbot & Boiral, 2013). Inefficient monitoring of such processes can lead to adverse environmental impacts such as pollution, global warming and climate change, which is the biggest challenge currently facing the planet (Rolnick et al., 2019). Therefore, there is a clear need to develop an efficient abnormal event management (AEM) system for such processes in order to ensure optimal and energy-efficient operations that will have positive impacts on the environment, people's safety, and the economy. Fault classification is one of the major procedures in AEM systems that are responsible for providing the decision-maker with comprehensible and comprehensive knowledge about a given fault in its initial phase in order to reduce any losses caused by that fault (Heo & Lee, 2018). A key factor for process operators and engineers is to understand the different fault classifiers and their relative merits in order to be able to select the appropriate classification method to apply in a particular situation.

A powerful fault classifier should be as generic and flexible as possible, and easy to adapt to a broad category of different highly nonlinear and complex dynamic systems found in process industries, such as recovery boilers, heat exchangers and evaporators (Ragab, Koujok, et al., 2019). The complexity of such systems is one of the major factors that hinder the applicability of model-based fault classification methods (Tidriri et al., 2016). The limitation of this family of methods is that a detailed model of the system is required in order to efficiently monitor its operating state (Sanderson & Gruen, 2006). Obtaining such models can be very difficult, time-consuming, and expensive, particularly for large-scale systems with many interacting variables.

Fortunately, process industries are equipped with an enormous amount of sensors and, accordingly, huge volumes of data observations have become available. The potential benefits offered by such data, as well as the advances in deep learning (DL) methods, are enough to support an investment decision towards accurate fault classification in digitalized industrial processes. The DL classifiers are very popular candidates that have been proven to be successful in a large number of industrial fault classification applications (LeCun et al., 2015; Lv et al., 2016; Goodfellow, Bengio, 2017). Specifically, the convolutional neural network (CNN) has been widely and successfully applied as DL classifiers in many applications in which the imagery data is a treasure for its learning process to produce accurate classification models (Zhang et al., 2019). These networks have been reported and it has been proven that they perform very well in fault classification applications (Afrasiabi et al., 2019; D. Lee et al., 2016; K. B. Lee et al., 2017; Wu & Zhao, 2018).

As numerical observations represent the major data sources in most process industries, it has become a prioritized need to better investigate the relationships between a huge number of process variables with the aim of developing accurate fault classifiers. Process operators still rely on data visualization tools such as correlograms (Wilke, 2019), principal component analysis (PCA) (Telea, 2007), and others, to show the complex relationships between variables. However, they cannot fully express the nonlinear and dynamic behaviors in complex systems of process industries, in addition to the unrealistic assumptions made on industrial data distributions such as normally distributed data. Therefore, there is a clear need to better transform and visualize these multi-dimensional data and to use the transformed data to train the CNNs and construct more accurate and representative classification models.

This motivates us to find a proper way to convert these observations into useful graph representations and then exploit the power of CNNs as pattern classifiers. A novel method for preprocessing and visualizing numeric data is proposed in this paper. The data observations are converted into proper graph representations (polygons). These representative graphs can systematically describe all interrelations between data variables based on Hamiltonian cycles. The data preprocessing and visualization procedure in this paper was inspired by the concept of parallel coordinate plots (PCP) (Inselberg & Dimsdale, 1990), which was introduced as a powerful data visualization and pattern recognition tool in the process industry (Choi et al., 2009; Dunia et al., 2013). In particular, its pattern recognition capability is the genesis of the proposed method. It is based on Hamiltonian cycles as an image generation tool that converts the numeric data into polygons to train the CNNs. Our proposed method shows a higher performance compared to other fault classifiers based on two case studies discussed in this paper; a benchmark dataset in the process industry and a challenging industrial dataset collected from a thermomechanical pulp mill located in Canada.

This paper is divided into six sections. Section 3.3 introduces some related work to the industrial fault classification using DL methods. Section 3.4 presents the proposed method along with the detailed description of its steps. Section 3.5 shows the two case studies used to validate the proposed method. Section 3.6 discusses the results obtained from the proposed method and other fault classifiers for the purpose of performance comparison, remarks on the proposed method, and provides some insights into future work. Section 3.7 concludes the paper.

3.3 Industrial Fault Classification Using Deep Learning Methods

Fault diagnosis is indispensable in engineering processes. The detection and diagnosis of faults are not only important for safety, but also for their environmental and economic impacts and for process maintenance activities as well. Challenges are still faced in fault diagnosis due to the complexity of industrial processes, interactions between the process variables, and lack of practical and accurate fault classification models (Tidriri et al., 2016). Consequently, it is difficult to obtain an adequate model-based diagnostic method in order to identify the relationship among the different parts of the process. Alternatively, data-driven methods can be more practical and more direct.

DL methods have recently experienced rapid development and received significant attention in fault classification due to their outstanding performance. The transition from shallow machine learning (ML) classifiers to DL classifiers can be attributed to certain reasons. First, the availability of an enormous amount of data allows the efficient training of deep neural networks and outperforming the other classical fault classifiers. Second, the advancement and evolution of algorithms and optimization techniques for training deep neural networks results in a better generalization of trained models. Finally, the use of high-performance graphical processing units (GPUs) accelerates the process of training such deep neural networks, thanks to their parallel computing capability (S. Zhang et al., 2019).

Adopting DL in fault classification has certain advantages over other ML classifiers. This is due to the fact that DL methods are characterized by embedded feature extraction. Moreover, these methods are already applied successfully to different problems such as language, vision and time series processing and can be easily adapted to a wide spectrum of fault classification applications (S. Zhang et al., 2019). A comprehensive literature review on these methods is found in (S. Zhang et al., 2019). In what follows, we briefly discuss some of these methods used in industrial fault classification.

Fault diagnosis in gearboxes was proposed in (Gecgel et al., 2019) using DL architectures, which outperform other comparable ML methods. As a chemical process, the Tennessee Eastman process (TEP) was investigated in (Ayubi Rad & Yazdanpanah, 2015) using supervised local multi-layer perceptron (SLMLP) classifiers along with independent component analysis (ICA) for fault detection and diagnosis. CNN is one of the most widely used DL methods in fault diagnosis in many industries due to its high performance. CNNs are used for fault detection and classification in semi-conductor manufacturing processes (Hsu & Liu, 2020; K. B. Lee et al., 2017). They are also used for bearing fault analysis using raw signal data and its frequency spectrograms (X. Chen et al., 2020; Gunerkar et al., 2019; D. Lee et al., 2016). A method for fault detection and diagnosis in the chemical process is proposed in (H. Wu & Zhao, 2018) based on CNNs, providing better classification accuracy. In (Afrasiabi et al., 2019), a method for wind turbine fault diagnosis was investigated using temporal CNNs as a classifier block along with Generative Adversarial Network (GAN) as a feature extraction block. CNNs are used in (Xiang Li et al., 2020; Y. Wang et al., 2020) for fault diagnosis of bearings in rotating machinery.

The question is how the process industry can leverage these advanced DL methods to develop accurate and representative fault classifiers for their complex systems? These industries are overwhelmed by a huge amount of numerical data that represents an important asset and needs to be fully exploited. Therefore, there is a clear need to transform these valuable data into a proper representation that can be used by DL methods to finally develop accurate fault classifiers. This can better model the complex relationships between industrial data variables, which are difficult to get in most cases in the process industries.

This paper introduces a novel method for preprocessing and visualizing data variables and relationships among them. The proposed method represents raw data as images in the form of polygons. Transformation of high-dimensional data into 2D polygon images that systematically and efficiently represent the interrelationships between all data variables using Hamiltonian cycles plays an important role in enhancing the performance of the DL classifiers. The concept of PCP (Inselberg & Dimsdale, 1990) motivates us towards the data preprocessing and visualization procedures in this paper. PCP was introduced as a powerful data visualization technique in several industrial applications (Dunia et al., 2013; Inselberg, 2009; Siirtola & Räihä, 2006). Moreover, PCP is used for visualizing the high-dimensional data and there have been many attempts to discover patterns from the PCP plots (Choi et al., 2009; Hauser et al., 2002; Santamaria et al., 2008; Wong & Bergeron, 1996). This inspires us to exploit the advantages of the PCP plots and to come up with the proposed polygon representation, which is the cornerstone of the proposed method and is presented in detail in the following section.

3.4 Proposed method

The proposed method consists of two main stages: 1) the preprocessing of numeric data, and 2) the training & testing stage. The preprocessing stage involves the polygon construction, and the output of this stage is passed to the learning (training) stage to train the CNN. The trained CNN will be used to test the new data points that were not seen before during the training stage, as shown in Figure 3.1. CNN has the advantage of the embedded feature extraction of images without the need for manual feature engineering of numerical data; this enhances the diagnosis performance of the trained model.

3.4.1 Preprocessing Stage

The preprocessing stage includes several steps. Each will be discussed in detail.

a) Step 1: Polygon Construction

Depending on the number of numerical variables in the training dataset, a polygon with a pre-specified side length is constructed with the number of sides equal to the number of variables. Given a dataset with *n* observations with numerical variables $(X_1, X_2, \ldots, X_m)$, each variable $X_j, j = 1, \ldots, m$ is represented as a polygon side, where m is the total number of variables. The output of this step is the 2-dimensional polygon shape with the coordinates of the midpoints of each polygon side $(\overrightarrow{X_j}, j = 1, \ldots, m)$ along with the unit vectors of each polygon side $(\widehat{X_j}, j = 1, \ldots, m)$. For the sake of explanation, Figure 3.2 shows a polygon constructed with four numerical variables. In addition, a simple numerical example will be illustrated later.

b) Step 2: Calculation of the Polygon Coordinates

The aim of this step is to standardize the numerical variables that have different scales. Many techniques can be used for that purpose, such as mean standardization, median standardization, and normalization. Figure 3.2 shows the mean standardization where the value of observation k for variable X_j (x_{kj}) is subtracted from the corresponding variable mean $\overline{X_j}$ and divided by its corresponding standard deviation δ_j as defined in Equation (1)**.**

$$Z_{kj} = \frac{x_{kj} - \overline{X_j}}{\delta_j}, \qquad j = 1, \ldots, m, k = 1, \ldots, n \qquad (1)$$

where Z_{kj} is the mean standardized value of the observation k for the variable X_j. However, normalization or median standardization methods can be used instead of mean standardization.

The performance of each of these methods can differ according to the nature of the training datasets. After data standardization, a pre-specified side length (L) of the polygon is calculated as two times the maximum value of Z_{kj}.

The midpoint of each polygon side ($\overrightarrow{X_j}$) represents the zero value of each variable. The coordinates of the variable X_j value for the observation k ($\overrightarrow{X_j^k}$) is calculated by adding $\overrightarrow{X_j}$ to the product of Z_{kj} and $\widehat{X_j}$ as shown in Equation (2).

$$\overrightarrow{X_j^k} = \overrightarrow{X_j} + \left(Z_{kj} * \widehat{X_j}\right) \qquad (2)$$

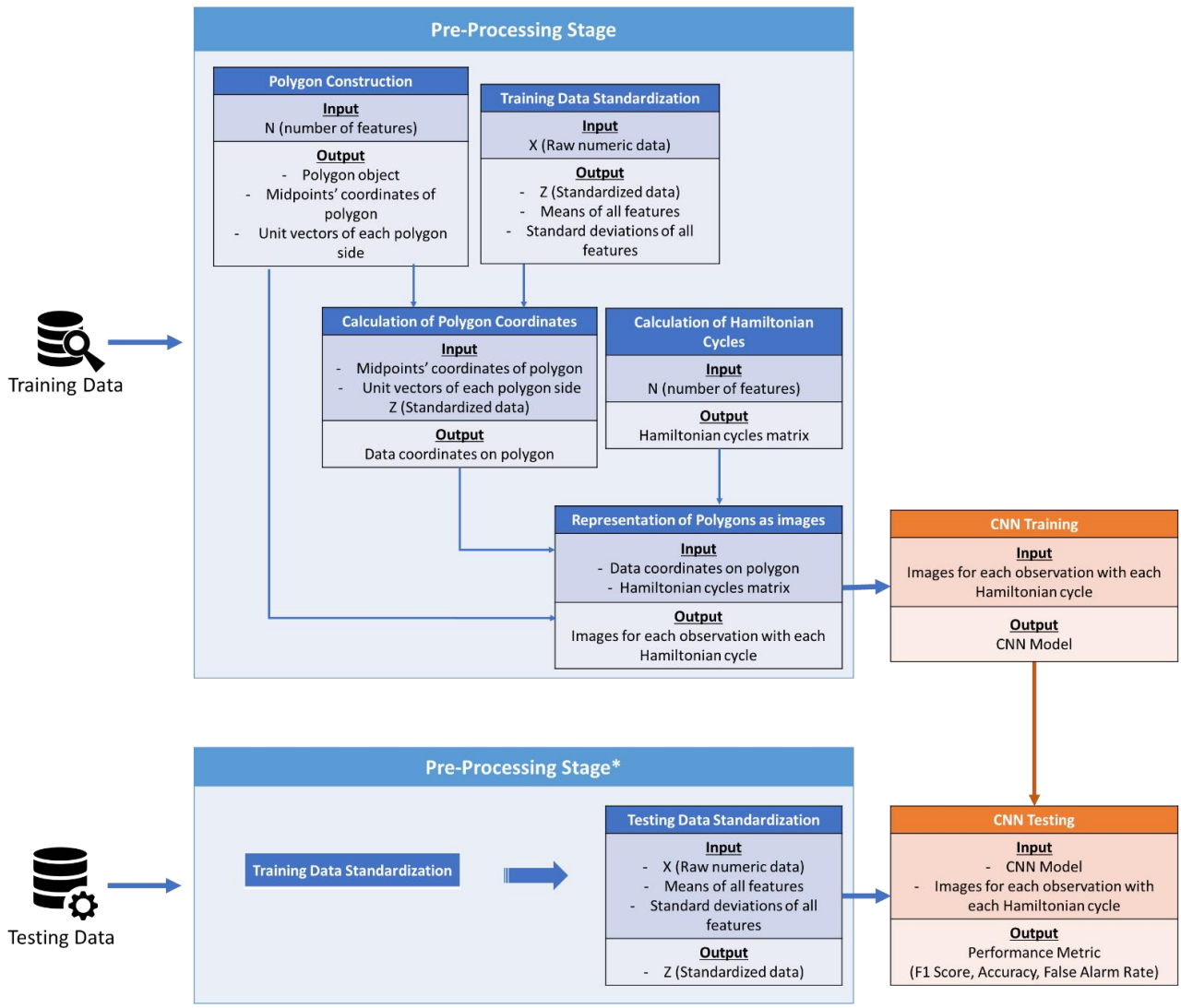
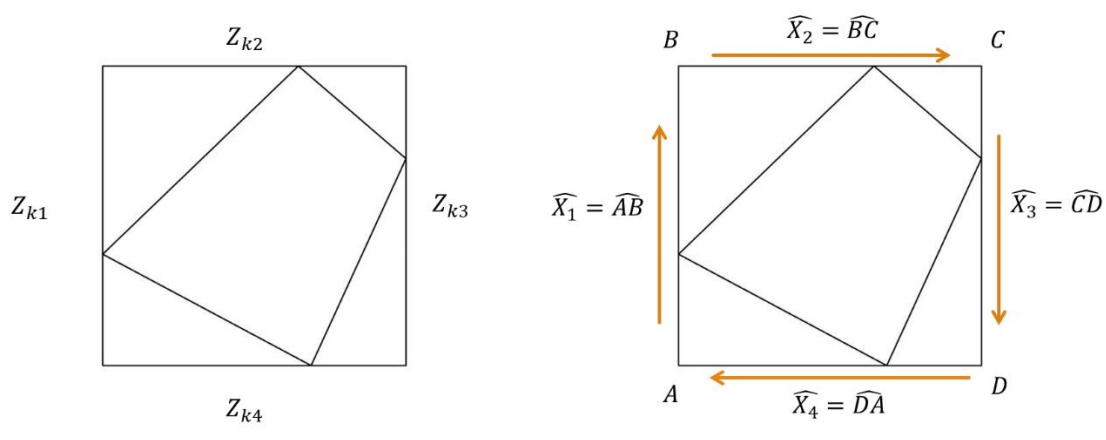


Figure 3.1 Diagram of the proposed method: Polygon Generation and Deep Learning for Classification, *the preprocessing stage for the testing dataset is the same as the training dataset, except for the replacement of the training data standardization with the testing data standardization



(a) (b)

Figure 3.2 – (a) Each polygon side represents a data variable where Z_{kj} is the standardized scalar value of variable X_j for observation k (b) $\widehat{X_j}$ is the unit vector of each polygon side and $\overrightarrow{X_j}$ represents the zero value of each variable as a point vector and $\overrightarrow{X_j^k}$ is the point vector of the variable value for the observation k

To exemplify the previous two steps, a numerical example is used based on the Iris data, a well-known benchmark dataset from the UCI machine learning repository (Bache & Lichman, 2013). In this paper, the four numeric variables of the data are expressed as X_1, X_2, X_3 and X_4. The observation #1 of the dataset is given as:

X_{11}	X_{12}	X_{13}	X_{14}
5.5	3.5	1.3	0.2

The means and standard deviations of the four variables are calculated as:

$$\overline{X_1} = 5.84, \overline{X_2} = 3.06, \overline{X_3} = 3.76, \overline{X_4} = 1.2$$
$$\delta_1 = 0.83, \delta_2 = 0.44, \delta_3 = 1.77, \delta_4 = 0.76$$

The standardized first observation is calculated using Equation (1) as:

Z_{11}	Z_{12}	Z_{13}	Z_{14}
-0.41	1.02	-1.39	-1.31

The standardized observations are converted into images of regular polygons with the number of sides equal to the number of variables. Accordingly, the polygon will be a square in this simple example. The side length L of the square would be two times the maximum value of the standardized data, calculated as $2 \times 3.08 = 6.16$, as shown in Figure 3.3.

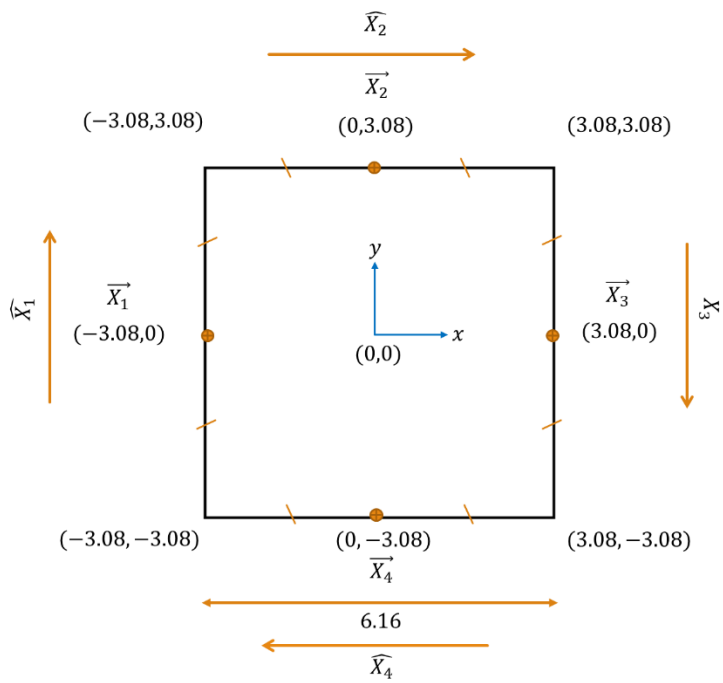


Figure 3.3 - Polygon (square) of a simple numeric dataset of four variables: the iris dataset example

As shown in Figure 3.3, the midpoint of each square side ($\overrightarrow{X_j}$) represents the zero value of its corresponding variable, calculated as $\overrightarrow{X_1} = -3.08\,\hat{q} + 0\hat{l}$, $\overrightarrow{X_2} = 0\hat{q} + 3.08\hat{l}$, $\overrightarrow{X_3} = 3.08\hat{q} + 0\hat{l}$, $\overrightarrow{X_4} = 0\hat{q} - 3.08\hat{l}$, where $\hat{q}$ and $\hat{l}$ are the unit vectors of the x and y directions, respectively.

The unit vectors of the polygon sides ($\widehat{X_j}$) are in a clockwise direction as shown in Figure 3.3, calculated using the square vertices as:

$$\widehat{X_1} = \frac{\left(-3.08\hat{q} + 3.08\hat{l}\right) - \left(-3.08\hat{q} - 3.08\hat{l}\right)}{6.16} = 0\hat{q} + 1\hat{l}$$

$$\widehat{X_2} = \frac{\left(3.08\hat{q} + 3.08\hat{l}\right) - \left(-3.08\hat{q} + 3.08\hat{l}\right)}{6.16} = 1\hat{q} + 0\hat{l}$$

$$\widehat{X_3} = \frac{\left(3.08\hat{q} - 3.08\hat{l}\right) - \left(3.08\hat{q} + 3.08\hat{l}\right)}{6.16} = 0\hat{q} - 1\hat{l}$$

$$\widehat{X_4} = \frac{\left(-3.08\hat{q} - 3.08\hat{l}\right) - \left(3.08\hat{q} - 3.08\hat{l}\right)}{6.16} = -1\hat{q} + 0\hat{l}$$

Equation (2) is used to calculate the point coordinates that represent the observation values on the polygon sides $\overrightarrow{X_j^i}$ as:

$$\overrightarrow{X_1^1} = \overrightarrow{X_1} + \left(Z_{11} * \widehat{X_1}\right) = -3.08\,\hat{q} - 0.41\,\hat{l}, \overrightarrow{X_2^1} = \overrightarrow{X_2} + \left(Z_{12} * \widehat{X_2}\right) = 1.02\,\hat{q} + 3.08\,\hat{l}$$

$$\overrightarrow{X_3^1} = \overrightarrow{X_3} + \left(Z_{13} * \widehat{X_3}\right) = 3.08\,\hat{q} + 1.39\,\hat{l}, \overrightarrow{X_4^1} = \overrightarrow{X_4} + \left(Z_{14} * \widehat{X_4}\right) = 1.31\,\hat{q} - 3.08\,\hat{l}$$

The four points are depicted in Figure 3.4. The same procedure is applied for all observations in the dataset using the same midpoints and unit vectors calculated previously.

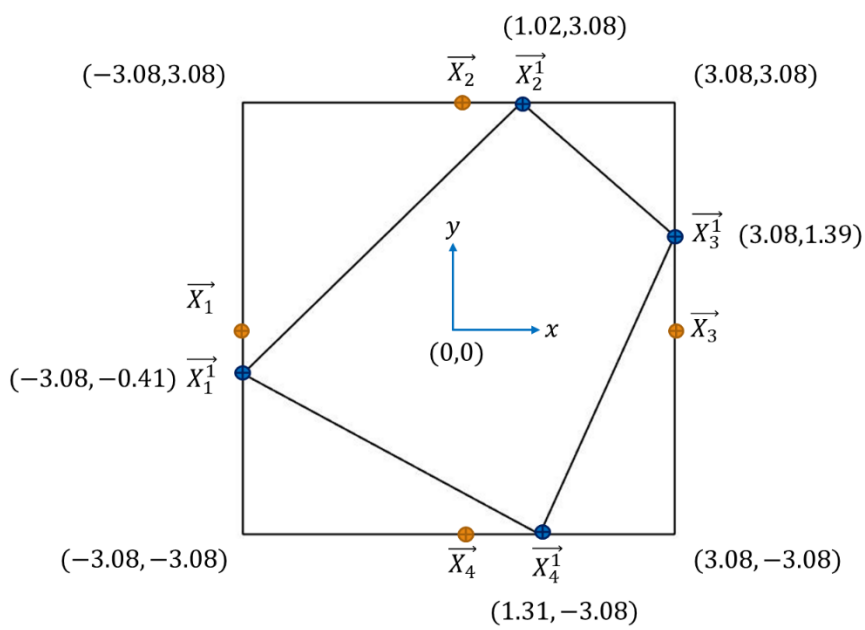


Figure 3.4 - A four-sided polygon with points on its sides that represent the observation values after standardization: the iris dataset example

c) Step 3: Calculation of Hamiltonian Cycles

After calculating the point coordinates of each variable in each data observation, there are different ways to connect these points. In order to represent each observation properly, all different connections are obtained in this step in a systematic way by using Lucas-Walecki Hamiltonian decompositions for N-Complete graphs (Hurley & Oldford, 2010; Wegman, 1990). It is worth mentioning that a complete graph with vertices representing the coordinate points of each observation on different polygon sides (variables) can include all possible connections between these variables. In this step, this complete graph is divided systematically into a finite number of Hamiltonian cycles that pass through all of the vertices only once. The minimum number of permutations of the m vertices (variables) containing the adjacencies for all pairs of vertices is r permutations (Hamiltonian cycles) for an even number $m = 2r$ and for an odd number $m = 2r + 1$. The procedure used to calculate the Hamiltonian cycles is stated in the algorithm in Table 3.1. In case of odd $m = 2r + 1$, the Hamiltonian path matrix would be the same matrix generated with even $m = 2r$ but concatenates an additional column vector with entries of m at the end of the matrix. Figure 3.5 shows the two Hamiltonian cycles of the observation given in the previous numerical example with four variables.

Table 3.1 - Algorithm: Calculating the Hamiltonian path matrix (Hurley & Oldford, 2010)

Input: number of vertices m
H is the Hamiltonian path matrix with r rows and m columns, $m = 2r$ (even m)
or $m = 2r + 1$ (odd m)
$H[1,1] = 0$
for $j = 2$ **to** m **do**
    for $k = 2$ **to** r **do**
        $H[1,j] = (H[1,j-1] + (-1)^j(j-1))mod(m)$
        $H[k,j] = (H[k-1,j] + 1)mod(m)$
    end for
end for

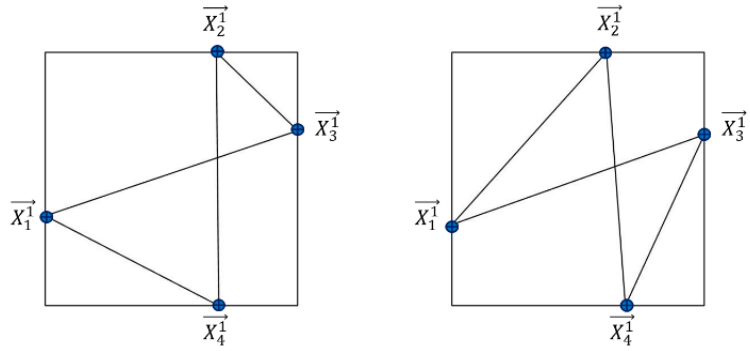


Figure 3.5 - Two polygons (images) for one observation with two different Hamiltonian cycles: the iris dataset example

d) Step 4: Representation of Polygons as Images

After calculating the Hamiltonian cycles and coordinate points of each observation, the coordinate points are connected to follow the Hamiltonian paths calculated previously. Each polygon along with the connections between coordinate points is converted into an image that represents one data observation with only one Hamiltonian cycle. Consequently, each observation is represented as a number of images equal to the number of Hamiltonian cycles that represent all possible connections between data variables. As an example, in the case of a dataset with 5 variables and 1000 observations, there exist 2 Hamiltonian cycles for each observation. Accordingly, there are 2000 images to learn the CNN architecture, as we will see in the next stage.

3.4.2 Learning Stage: CNN training and testing

In this stage, the CNN architecture is used for the training, validation and testing of the images generated from the preprocessing stage mentioned previously. It is worth mentioning that each of the images generated previously is labeled according to the label in the original numeric dataset. As a supervised learning method, the labeled images are used for the CNN training procedure. More specifically, for a classification problem with multiple classes, a numeric observation belonging to a certain class is transformed, using the Hamiltonian cycles, into a number of images that are all assigned the same label of that class. This means that the number of generated images is equal to the number of Hamiltonian cycles.

After training the CNN model with the labeled images, the parameters of each layer (the values of each filter) will be learned. Regarding the testing procedure, numeric testing (unseen) observations are preprocessed similarly as in the training procedure. Each of these testing observations is represented as a number of testing images. The trained CNN model is then used to predict the class label of each testing image. Finally, the predicted class of the numeric testing observation will be the majority voting class of its corresponding images.

All Hamiltonian cycles generated according to the algorithm mentioned in Table 1 have to be used in each observation to properly represent all of the relationships between data variables. When the number of Hamiltonian cycles is divisible by the number of classes, more random cycles are added to avoid random voting. These random cycles represent redundant relationships between variables since the generated Hamiltonian cycles from the algorithm express all relationships between variables. For example, if we have 8 data variables and 2 classes (binary classification problem), then we will have 4 Hamiltonian cycles (4 images per observation). In this case, random voting may be encountered in the case of 2 images assigned as class #1 and the other 2 images assigned as class #2. To avoid such cases, more cycles (randomly generated) are added to the systematically generated Hamiltonian cycles.

3.5 Fault classification case studies

In this subsection, two case studies are presented to validate the proposed classification method. The first case study is the Tennessee Eastman Process (TEP), a well-known problem in the field of process monitoring that uses simulated data. The second case study is a real dataset collected from a reboiler system in the heat recovery network (HRN) in a thermomechanical pulp (TMP) mill located in Canada. In these complex processes, an incipient fault can propagate through many parts and can cause undesired abnormal situations.

3.5.1 Case Study 1: Tennessee Eastman Process (TEP)

The TEP is a benchmark industrial process for the Eastman Chemical Company proposed by Downs and Vogel (Downs & Vogel, 1993), consisting of main units including a reactor, a condenser, a recycle compressor, a liquid-vapor separator and a product stripper. The diagram of the TEP process is shown in Figure 3.6 (Downs & Vogel, 1993). This process produces three liquid

products F, G and H and fed by four reactants A, C, D and E and an inert gas B. As shown in Figure 3.6, the circled elements in the diagram are forty-one sensing devices representing measured variables. Among these measured variables, twenty-two variables are measured continuously including temperatures, pressures, levels, and flows. The remaining 19 measured variables are composition analyses collected from chromatographs and sampled at different times (6 or 15 min). The twelve manipulated variables of the process are the positions of the eleven valves and the reactor agitator speed. The interested readers are referred to (Downs & Vogel, 1993) for more details.

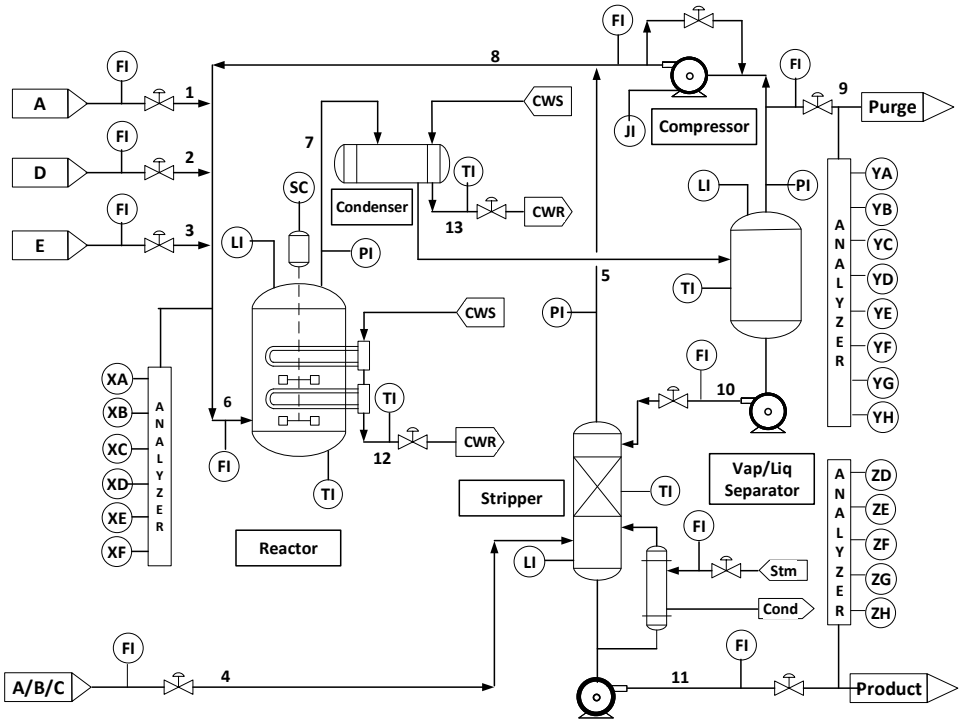


Figure 3.6 - Case Study 1: Tennessee Eastman Process (TEP) (Downs & Vogel, 1993)

Since it is difficult or even impossible to obtain a model-based fault diagnosis method for such complex process, a data-driven diagnostic methods are practical alternatives. The simulation of the process is a straightforward way to generate data representing different process states. A simulator for the TEP process was originally introduced by (Downs & Vogel, 1993). It acts as a real TEP and allows the user to analyze the process under faulty and normal conditions. The simulator has been used by researchers for studying topics such as fault classification, process

optimization, control and others (Bathelt et al., 2015; D'Angelo et al., 2016; Duvall & Riggs, 2000; Golshan et al., 2005; Ragab et al., 2018; Yin et al., 2012).

Stabilization of the TEP process was investigated in the literature using different control strategies (Larsson et al., 2001; Larsson & Skogestad, 2000; McAvoy & Ye, 1994; Ricker, 1996). It is well-known that fault classification of the TEP simulated data is still challenging due to its contamination with different noise sources. Besides, other challenges are faced when classifying faults in the controlled TEP. This is due to the fast reaction of controllers to the process disturbance. This paper addresses the problem of fault classification for the controlled TEP process using the revised and modified TEP simulator found in (Bathelt et al., 2015).

3.5.2 Case Study 2: The Reboiler System of Heat Recovery Network in TMP Mills

The reboiler system is a major equipment in the heat recovery network in TMP mills that are among the most energy-intensive processes. The reboiler is a complex equipment used for clean steam production (Bajpai, 2018). Figure 3.7 shows a simplified diagram of the heat recovery network in TMP mills. A significant amount of energy is applied in TMP process for the sake of wood fibers' separation and fibrillation. However, most of this energy is recovered as dirty steam and only a small amount is used for wood refining. As shown in Figure 3.7, the dirty steam that comes from the TMP lines is the input for the reboiler. The output is the clean steam that goes to low pressure (LP) steam network. Another LP steam from the turbine is fed into the LP steam network, so that the produced clean steam can be used by paper machines' dryers and other LP steam users at the mill.

One of the major abnormal situations in the HRN is the opening of the dirty steam venting valve that is caused by a pressure increase in the dirty steam header. This leads to energy losses into the atmosphere, moreover, the efficiency of the heat recovery system is affected significantly. Extracting an extra LP steam from the turbine can compensate the shortage of the clean steam production, however this is not economically beneficial. Therefore, there is a clear need to efficiently classify the different causes of this abnormal event, and this is the main purpose of applying the proposed method to this real case study. The objective is to lessen and alleviate such negative economic and environmental consequences.

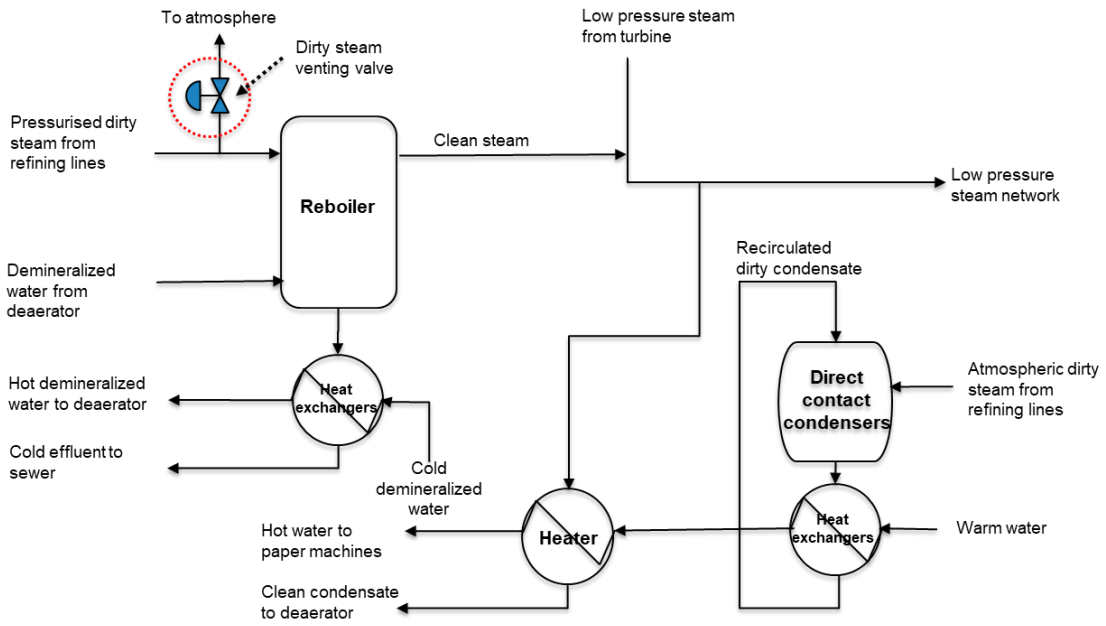


Figure 3.7 - Case Study 2: Reboiler system of Heat Recovery Network (HRN) in TMP mills

The data is collected from the HRN during one year with thirty-minute time intervals including a set of manipulated and measured variables. Data cleaning and preparation including removal of outliers and other non-representative observations was done with the assistance of the process expert and by using the EXPLORE Software (Amazouz, 2015). Another set of variables were also calculated by the expert based on the manipulated and measured variables. Examples of these variables are shown in Table 2.

Table 3.2 - Examples of manipulated, measured, and calculated variables for the HRN

Refiners and heat recovery system operation related variables (manipulated and measured variables)	Calculated variables (variables describing the process behavior)
▪ Power to refiners ▪ Production rates of TMP lines ▪ Dilution water flows at different locations ▪ Pressure levels at different locations ▪ Temperature at different locations	▪ Recovered energy ▪ Specific energy applied to each refiner ▪ Differential Pressure

3.6 Experimental setup & Results

Based on the case studies that were previously presented, the performance of the proposed classification method is compared with shallow ML fault classifiers; Multi-layer Perceptron (MLP), Quadratic Discriminant Analysis (QDA), k-Nearest Neighbors (kNN), Support Vector Machine (SVM). These ML methods have been used extensively in the literature and in practice (Eren, Ince, & Kiranyaz, 2019; Jing & Hou, 2015; King, Feng, & Sutherland, 1995; Ragab, Koujok, et al., 2019; Ragab, Yacout, Ouali, & Osman, 2019; Tian, Morillo, Azarian, & Pecht, 2016). In addition, the proposed method is compared with state-of-art DL algorithms; One Dimensional – CNN (1D-CNN) (Eren et al., 2019; Hasan et al., 2019; Peng et al., 2019), Radial Basis Function Neural Network (RBFNN) (Qi et al., 2015; Qing Zhang et al., 2018; W. Zhou et al., 2019) and Deep Belief Network (DBN) (Z. Chen et al., 2016; Z. Li et al., 2020; S. Y. Shao et al., 2017; Z. Zhang & Zhao, 2017) used for fault diagnosis in industrial processes. These DL fault classifiers are tested without data preprocessing. Figure 3.8 summarizes the main steps of the proposed methodology applied to the case studies that were previously mentioned.

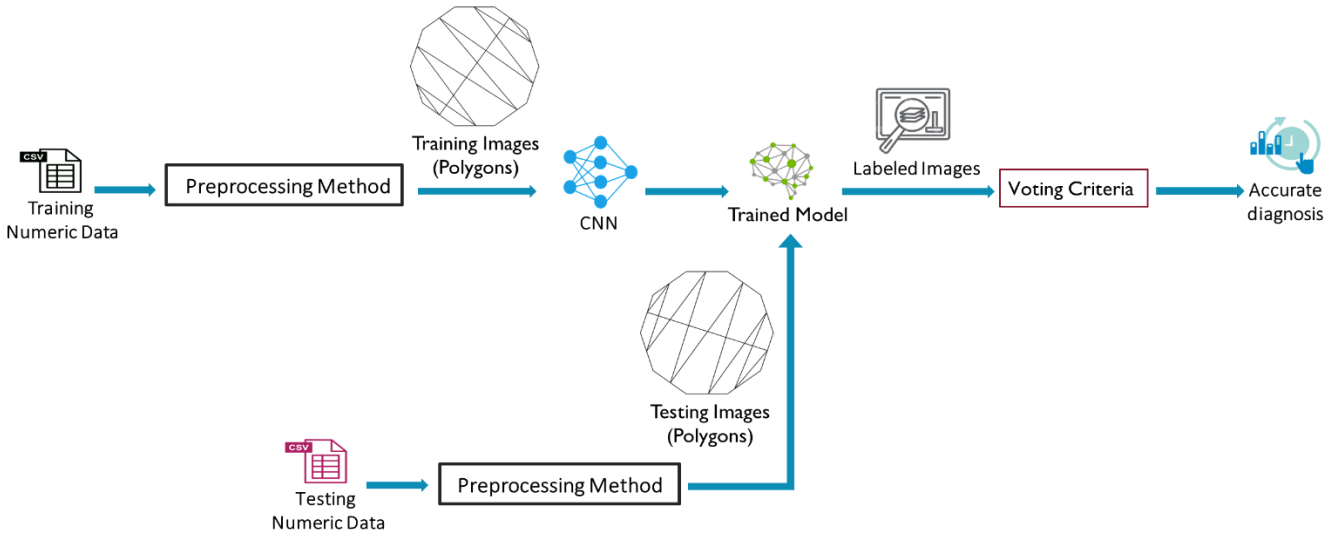


Figure 3.8 - The main steps of the proposed methodology applied to the two fault classification case studies

The images of polygons generated from numerical data were used for training and testing a CNN architecture shown in Table 3.3 with hyperparameters of filter size (r*r), the number of filters (n), batch size and number of epochs. The Keras package (Chollet et al., 2015) with Python 3.7 was used on the computing infrastructure; Intel(R) Core(TM) i7-8750H CPU @2.2 GHz + NVIDIA GeForce GTX 1070 with Max-Q Design.

Table 3.3 - Baseline Convolutional Neural Network (CNN) architecture

Input layer (64*64 grey-scale image)
2D Convolutional layer stage with n filters with size (r*r) and zero-padding in each layer + ReLU layer
2D Convolutional layer stage with n filters with size (r*r) and zero-padding in each layer + ReLU layer
Maxpool (2) with stride 2
2D Convolutional layer stage with 2n filters with size (r*r) and zero-padding in each layer + ReLU layer
Maxpool (2) with stride 2
2D Convolutional layer stage with 4n filters with size (r*r) and zero-padding in each layer + ReLU layer
2D Convolutional layer stage with 4n filters with size (r*r) and zero-padding in each layer + ReLU layer
Fully connected stage with neurons equal number of classes
Softmax

To compare the proposed method with the best results achieved from other ML and DL fault classifiers - MLP, QDA, kNN, SVM, 1D-CNN, RBFNN and DBN - the hyperparameters of these algorithms with specified ranges are optimized using grid search and 5-fold cross validation for each of the two case studies, as shown in Table 3.4. Regarding the reproducibility of the results, a random seed is fixed for training DL models. This is to ensure the consistency of trained models. The F1 Score of each class, accuracy, and false alarm rate (FAR) are used as the performance metrics for testing each classifier. The FAR is calculated using Equations (3) and (4), given in (Lipton et al., 2014; Om & Kundu, 2012) as:

$$FAR = \frac{FP}{FP + TN}, \qquad (3)$$

$$F1 = \frac{2\,TP}{2\,TP + FP + FN}, \qquad (4)$$

where FP, TN, TP and FN are the number of false positives, true negatives, true positives and false negatives, respectively.

Table 3.4 - Range of hyperparameters of each fault classifier

Algorithm	**Hyperparameters**
Proposed Method	n = [4,64], r = (2,3), batch size = [50,200] # epochs = [30,150], activation function = {sigmoid, ReLU}
MLP	# neurons in hidden layer = [10,40] Maximum number of iterations = [1000,5000]
KNN	K = [3,15] Weights = {‘uniform’,’weighted with distance’} Distance = {‘Euclidean’,’Manhattan’}
QDA	Regularization coefficient = (0,0.9]
SVM	C (penalty parameter) = [0.01,10] Kernel = {‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’}
1D-CNN	# filters = [4,32], batch size=[4,32], # epochs = [10,50], activation function = {sigmoid, ReLU}, kernel size = [2.8]
RBFNN	# clusters = [5,30] coefficient of smoothing exponential kernel = [0.1,0.9]
DBN	# epochs = [30,200], activation function = {sigmoid, ReLU}, batchsize = [50,150] , learning rate = [0.001,0.1], number of hidden layers = [2,4], hidden layer sizes = [64,512]

3.6.1 Results of the TEP dataset

In the TEP dataset, among the 53 variables, 48 variables were selected to represent the whole dynamic behavior of the process. We seeded five faults in addition to the normal operations. Accordingly, we have a total of six classes of observations. According to the algorithm shown in

Table 3.1, each labeled observation has 24 different Hamiltonian cycle connections that completely represent all connections between variables. More specifically, a data with 48 variables will result in 24 different Hamiltonian connections (i.e., images) per observation. Each image represents different connection order of the same points on the resulting 48-sided polygon. Accordingly, all these 24 different connections consistently represent all interrelationships between the 48 variables. The predicted class of the unlabeled (testing) observations is the majority voting class of its 24 unlabeled images.

As shown in Table 3.5, the proposed method has achieved the highest F1 score in the normal class and fault #5. It is worth mentioning that fault #5 is considered to be a challenging fault in most related work mentioned in the literature. Most of those algorithms either achieve a very low F1 score for fault #5 or may not even detect it at all. This is due to the fact that the controllers react very rapidly to eliminate the effects of this incipient fault, while the fault causes are still there. From a practical point of view, this a serious issue in most controlled plants in which the processes can be subject to faults that lead to an abnormal operation, while the operators are not able to discover such abnormalities. This confirms that the proposed method could be a promising solution in highly dynamic processes that need accurate fault classifiers with the capability for quick responses. Moreover, the proposed method has achieved the highest F1 score for the normal class, which is an important merit for the fault classifier in assisting the operators with efficiently monitoring the normal operation state. Moreover, the proposed method achieves the highest accuracy and lowest FAR among the other fault classifiers.

Table 3.5 - F1 Scores, Accuracy and False alarm rate (FAR) of each algorithm in TEP dataset

Algorithm		*Normal*	*Fault*	*Fault*	*Fault*	*Fault*	*Fault*	*Accuracy*	*FAR*
Proposed Method		**0.92**	1	1	1	**0.5**	1	**95.83**	**0.17**
ML Fault Classifiers	SVM	0.87	1	0.99	1	0	1	94.32	0.23
	MLP	0.8	1	0.99	1	0.28	1	93.2	0.19
	QDA	0.88	1	1	1	0	1	92.42	0.2
	kNN	0.83	0.99	0.98	1	0	1	92.42	0.25
DL Fault Classifiers	1D-CNN	0.7	0.96	1	0.75	0	0.73	85.3	0.27
	RBFNN	0.86	1	0.98	1	0	1	93.94	0.24
	DBN	0.87	1	0.99	1	0	1	94.3	0.23

It is worth mentioning that for the 48 numeric variables, the size of the input images is set to 64*64 pixels. A CNN architecture with only five convolutional layers is used to detect the difference between each class without the need to build more sophisticated and deeper architectures, such as VGG16 and VGG19 (Simonyan & Zisserman, 2014).

3.6.2 Results of the Reboiler dataset

For the reboiler data, the process operator was able to define the normal operating conditions, the faulty situations, and their causes. The operator has assigned the label for each observation based on a developed cause-and-effect table. The abnormal situation mentioned before, a dirty steam header venting valve opening, has eight different causes documented in the diagnostic knowledge base repository in the mill.

In these experiments, 27 numerical variables and three challenging causes were selected to test the performance of our proposed method. As shown in Table 3.6, our proposed method outperforms the other classifiers in terms of F1 scores, accuracy, and FAR. Despite being a challenging dataset, our method has been proven to be robust and accurate in an industrial use case.

Table 3.6 - F1 Scores, Accuracy and False alarm rate (FAR) of each algorithm in the Reboiler data

Algorithm		*Normal*	*Fault 1*	*Fault 2*	*Fault 3*	*Accuracy (%)*	*FAR*
Proposed Method		**0.97**	**0.82**	**0.89**	**0.86**	**92.74**	**0.053**
ML Fault Classifiers	SVM	0.93	0.74	0.8	0.8	90.82	0.074
	MLP	0.94	0.8	0.78	0.8	91.44	0.064
	QDA	0.96	0.77	0.7	0.71	89.92	0.082
	kNN	0.92	0.5	0.7	0.78	85.71	0.11
DL Fault Classifiers	1D-CNN	0.92	0.4	0.67	0.5	83.33	0.15
	RBFNN	0.94	0.8	0.8	0.8	90.83	0.065
	DBN	0.92	0.7	0.76	0.8	87.5	0.078

These results obtained are useful for the process operators, allowing them to accurately classify the different causes for such an abnormal situation. Consequently, the operators can select the appropriate corrective actions at the right time. These corrective actions have a great impact on the mill's efficiency and the environment as well. This impact is out of the scope of this paper and will be studied later in different work.

3.6.3 Remarks

An important merit of this method is that it uses Hamiltonian cycles for the visual representation of observations, which allows the expression of all interrelationships between data variables. Moreover, in image classification problems using CNNs, a data augmentation technique is used to build robust classification models. On the other hand, in our method, the idea of connecting different variables with Hamiltonian cycles make the polygons generated from a certain observation seem to be rotated versions of just one image. This means that an observation can be seen from different perspectives to deeply extract class-specific representations. Accordingly, an analogy can be observed between the data augmentation technique and generated polygons. Besides, the procedure used for generating polygons can eliminate the need for a huge amount of data for accurate deep learning models.

The proposed method achieves these results in the two case studies without the need for feature selection or extraction. Having representative features that describe the raw data well has become a more important step in deep learning than choosing a suitable algorithm to be used. Instead of manual feature engineering, feature extraction in our methodology is embedded in CNN to select the best features from the data visualization. This is called an “end-to-end learning” technique.

According to the nature of data and its distribution, different preprocessing techniques can be used such as mean standardization, median standardization or Max-min normalization. Another advantage of the proposed method is that the majority voting labeling strategy used for testing raw numeric data may increase the final accuracy of the trained CNN classifier.

Despite having more hyperparameters in DL networks than shallow ML algorithms, efficient hyperparameter optimization methods and the access to GPUs with high computational power can greatly shorten the training time of such complex networks.

3.6.4 Future work

In future work, testing our method on datasets in the process industries with a large number of variables needs to be investigated to further ensure the consistency of the proposed method. One of the possibilities is to test it on challenging datasets collected from Kraft pulp mills. We also may need to extend this study to consider the effect of image resolution, which may have a significant impact on the accuracy of the trained classifiers. Moreover, the CNN architecture can be further optimized using the neural architecture search (NAS) approach that automates the design procedures of DL models. Furthermore, solving unsupervised learning problems may be possible using our method to make use of the unlabeled industrial dataset. Finally, extracting interpretable patterns is an urgent need for many industries. The idea of polygon generation presented in this paper can be an important step towards the interpretability of deep learning models. This is one of our future directions. Besides, the conversion of numerical data into images can be a cornerstone for building a data fusion tool that blends numerical and imagery data.

3.7 Conclusion

This paper proposes a data preprocessing method that converts numeric data into polygons, expressing all of the relationships between data variables based on Hamiltonian cycles. The

procedure used for generating polygons from numerical data is analogous to the data augmentation used in training convolutional neural networks (CNNs). Furthermore, the method can help support end-to-end learning, which can intrinsically extract class-specific features in industrial fault classification applications. It can be concluded that the proposed method has been proven to be a promising tool for accurate fault classification in the process industry. It was validated with two datasets: the first one is a well-known benchmark problem that uses simulated data. The second one is a real industrial dataset collected from a reboiler system in a thermomechanical pulp (TMP) mill located in Canada. The results obtained show that the proposed method outperforms other comparable machine learning (ML) and deep learning (DL) fault classifiers in terms of its higher accuracy and lower false alarm rate (FAR). In future work, the proposed method will be further tested on other datasets in the process industry. Further developments related to the resolution of polygon images will be considered. Moreover, the generated polygons can play a significant role in the interpretation of trained deep learning fault classifiers. The proposed method will be a cornerstone for building a data fusion methodology that blends different sources of heterogeneous data.

3.8 References

Abadi, Mart\'\in, Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., & others. (2016). Tensorflow: A system for large-scale machine learning. *12th $\{$USENIX$\}$ Symposium on Operating Systems Design and Implementation ($\{$OSDI$\}$ 16)*, 265–283.

Abadi, Martin, Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., S. Corrado, G., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Yangqing, J., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/

Afrasiabi, S., Afrasiabi, M., Parang, B., Mohammadi, M., Arefi, M. M., & Rastegar, M. (2019). Wind Turbine Fault Diagnosis with Generative-Temporal Convolutional Neural Network. *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, 1–5.

Aggarwal, K., Kirchmeyer, M., Yadav, P., Keerthi, S. S., & Gallinari, P. (2019). *Benchmarking Regression Methods: A comparison with CGAN*. *May*. http://arxiv.org/abs/1905.12868

Alam, F., Mehmood, R., Katib, I., Albogami, N. N., & Albeshri, A. (2017). Data fusion and IoT for smart ubiquitous environments: A survey. *IEEE Access*, *5*, 9533–9554.

Alpaydin, E. (2010). *Introduction to machine learning, 2nd edn. Adaptive computation and machine learning*. The MIT Press (February 2010).

Amazouz, M. (2015). *Improving Process Operation Using the Power of Advanced Data Analysis*. https://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/canmetenergy/files/pubs/EXPLORE-brochure_ EN.pdf

Anaconda Software Distribution. (2020). In *Anaconda Documentation*. Anaconda Inc. https://docs.anaconda.com/

Andersson, E., & Thollander, P. (2019). Key performance indicators for energy management in the Swedish pulp and paper industry. *Energy Strategy Reviews*, *24*(December 2018), 229–235. https://doi.org/10.1016/j.esr.2019.03.004

Andrew Ng Launches A Campaign For Data-Centric AI. (n.d.). https://www.forbes.com/sites/gilpress/2021/06/16/andrew-ng-launches-a-campaign-for-data-centric-ai/?sh=5dea92f374f5

Ardsomang, T., Hines, J. W., & Upadhyaya, B. R. (2013). Heat exchanger fouling and estimation of remaining useful life. *Annual Conference of the PHM Society*, *5*(1).

Arendt, J. S., & Lorenzo, D. K. (2010). *Evaluating process safety in the chemical industry: A user's guide to quantitative risk analysis* (Vol. 3). John Wiley \& Sons.

Athar, M., Mohd Shariff, A., Buang, A., Shuaib Shaikh, M., & Ishaq Khan, M. (2019). Review of Process Industry Accidents Analysis towards Safety System Improvement and Sustainable Process Design. *Chemical Engineering and Technology*, *42*(3), 524–538. https://doi.org/10.1002/ceat.201800215

Ayubi Rad, M. A., & Yazdanpanah, M. J. (2015). Designing supervised local neural network classifiers based on EM clustering for fault diagnosis of Tennessee Eastman process.

Chemometrics and Intelligent Laboratory Systems, *146*, 149–157. https://doi.org/10.1016/j.chemolab.2015.05.013

Bache, K., & Lichman, M. (2013). *UCI machine learning repository*.

Bai, Y., Xie, J., Wang, D., Zhang, W., & Li, C. (2021). A manufacturing quality prediction model based on AdaBoost-LSTM with rough knowledge. *Computers and Industrial Engineering*, *155*(January), 107227. https://doi.org/10.1016/j.cie.2021.107227

Bajpai, P. (2018). Brief Description of the Pulp and Papermaking Process. In *Biotechnology for pulp and paper processing* (pp. 9–26). Springer.

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., & others. (2012). NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Research*, *41*(D1), D991--D995.

Bashkirova, D., Usman, B., & Saenko, K. (2018). *Unsupervised Video-to-Video Translation*. *Nips*. http://arxiv.org/abs/1806.03698

Bathelt, A., Ricker, N. L., & Jelali, M. (2015). Revision of the tennessee eastman process model. *IFAC-PapersOnLine*, *48*(8), 309–314.

Baturynska, I., & Martinsen, K. (2020). Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms. *Journal of Intelligent Manufacturing*, *32*(1), 179–200. https://doi.org/10.1007/s10845-020-01567-0

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Biermann, C. J. (1996). *Handbook of pulping and papermaking*. Elsevier.

Borovykh, A., Bohte, S., & Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10614 LNCS*, 729–730.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley \& Sons.

Brown, R. G., & Meyer, R. F. (1961). The fundamental theorem of exponential smoothing. *Operations Research*, *9*(5), 673–685.

Bustillo, A., Pimenov, D. Y., Mia, M., & Kapłonek, W. (2020). Machine-learning for automatic prediction of flatness deviation considering the wear of the face mill teeth. *Journal of Intelligent Manufacturing*, *mm*. https://doi.org/10.1007/s10845-020-01645-3

Bustillo, A., Reis, R., Machado, A. R., & Pimenov, D. Y. (2020). Improving the accuracy of machine-learning models with data from machine test repetitions. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01661-3

Camacho, J., Magán-Carrión, R., García-Teodoro, P., & Treinen, J. J. (2016). Networkmetrics: multivariate big data analysis in the context of the internet. *Journal of Chemometrics*, *30*(9), 488–505. https://doi.org/10.1002/cem.2806

Cao, S., Wen, L., Li, X., & Gao, L. (2018). Application of Generative Adversarial Networks for Intelligent Fault Diagnosis. *IEEE International Conference on Automation Science and Engineering*, *2018-Augus*, 711–715. https://doi.org/10.1109/COASE.2018.8560528

Castanedo, F. (2013). A review of data fusion techniques. In *The Scientific World Journal* (Vol. 2013). Hindawi Publishing Corporation. https://doi.org/10.1155/2013/704504

Chamzas, D., Chamzas, C., & Moustakas, K. (2020). cMinMax: A fast algorithm to find the corners of an N-dimensional convex polytope. *ArXiv:2011.14035v2*.

Chen, X., Zhang, B., & Gao, D. (2020). Bearing fault diagnosis base on multi-scale CNN and LSTM model. *Journal of Intelligent Manufacturing*, *December 2019*. https://doi.org/10.1007/s10845-020-01600-2

Chen, Z., Zeng, X., Li, W., & Liao, G. (2016). Machine fault classification using deep belief network. *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, *2016-July*(51475170). https://doi.org/10.1109/I2MTC.2016.7520473

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio,

Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Choi, H., Lee, H., & Kim, H. (2009). Fast detection and visualization of network attacks on parallel coordinates. *Computers & Security*, *28*(5), 276–288.

Chollet, F., & others. (2015). *Keras*. GitHub.

Chollet, F., & others. (2018). Keras: The python deep learning library. *Astrophysics Source Code Library*.

Cocchi, M. (2019). Introduction: Ways and Means to Deal With Data From Multiple Sources. *Data Handling in Science and Technology*, *31*, 1–26. https://doi.org/10.1016/B978-0-444-63984-4.00001-6

D'Angelo, M. F. S. V, Palhares, R. M., Camargos Filho, M. C. O., Maia, R. D., Mendes, J. B., & Ekel, P. Y. (2016). A new fault classification approach applied to Tennessee Eastman benchmark process. *Applied Soft Computing*, *49*, 676–686. https://doi.org/https://doi.org/10.1016/j.asoc.2016.08.040

Data-Centric AI Competition. (n.d.). https://https-deeplearning-ai.github.io/data-centric-comp/

Demir, U., & Unal, G. (2018). Patch-based image inpainting with generative adversarial networks. *ArXiv:1803.07422v1*.

Dong, D., Li, X.-Y., & Sun, F.-Q. (2017). Life prediction of jet engines based on LSTM-recurrent neural networks. *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 1–6.

Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, *17*(3), 245–255.

Dunia, R., Edgar, T. F., & Nixon, M. (2013). Process monitoring using principal components in parallel coordinates. *AIChE Journal*, *59*(2), 445–456.

Durall, R., Chatzimichailidis, A., Labus, P., & Keuper, J. (2020). Combating Mode Collapse in

GAN training: An Empirical Analysis using Hessian Eigenvalues. *ArXiv Preprint ArXiv:2012.09673*.

Duvall, P. M., & Riggs, J. B. (2000). On-line optimization of the Tennessee Eastman challenge problem. *Journal of Process Control*, *10*(1), 19–33. https://doi.org/https://doi.org/10.1016/S0959-1524(99)00041-4

Elhefnawy, M., Ragab, A., & Ouali, M.-S. (2021a). Fault classification in the process industry using polygon generation and deep learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-021-01742-x

Elhefnawy, M., Ragab, A., & Ouali, M. S. (2021b). Fault classification in the process industry using polygon generation and deep learning. *Journal of Intelligent Manufacturing*, *0123456789*. https://doi.org/10.1007/s10845-021-01742-x

Elizabeth Bush Nathan Gillett, E. W. J. F., & others. (2019). *Canada's Changing Climate Report*. https://changingclimate.ca/CCCR2019/

Environment challenges | Climate Action. (n.d.). https://ec.europa.eu/clima/policies/adaptation/how/challenges_en#tab-0-1

Eren, L., Ince, T., & Kiranyaz, S. (2019). A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *Journal of Signal Processing Systems*, *91*(2), 179–189.

Essien, A., & Giannetti, C. (2020). A Deep Learning Model for Smart Manufacturing Using Convolutional LSTM Neural Network Autoencoders. *IEEE Transactions on Industrial Informatics*, *16*(9), 6069–6078. https://doi.org/10.1109/TII.2020.2967556

FP, L. (1996). *Loss Prevention in the Process Industries: Hazard Identification, Assessment and Control*.

Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83–85.

Fu, K., Peng, J., Zhang, H., Wang, X., & Jiang, F. (2020). Image super-resolution based on generative adversarial networks: A brief review. *Computers, Materials and Continua*, *64*(3), 1977–1997. https://doi.org/10.32604/cmc.2020.09882

Fuentes, R., Fuster, B., & Lillo-Bañuls, A. (2016). A three-stage DEA model to evaluate learning-teaching technical efficiency: Key performance indicators and contextual variables. *Expert Systems with Applications*, *48*, 89–99. https://doi.org/10.1016/j.eswa.2015.11.022

Gamboa, J. C. B. (2017). *Deep Learning for Time-Series Analysis*.

Gärtler, M., Khaydarov, V., Klöpper, B., & Urbas, L. (2021). The Machine Learning Life Cycle in Chemical Operations – Status and Open Challenges. *Chemie Ingenieur Technik*, *12*, 1–19. https://doi.org/10.1002/cite.202100134

Ge, Z. (2017). Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemometrics and Intelligent Laboratory Systems*, *171*(September), 16–25. https://doi.org/10.1016/j.chemolab.2017.09.021

Gecgel, O., Ekwaro-Osire, S., Dias, J. P., Serwadda, A., Alemayehu, F. M., & Nispel, A. (2019). Gearbox Fault Diagnostics Using Deep Learning with Simulated Data. *Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management*, 1–8.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471.

Gheisari, M., Wang, G., & Bhuiyan, M. Z. A. (2017). A survey on deep learning in big data. *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, *2*, 173–180.

Golshan, M., boozarjomehry, R. B., & Pishvaie, M. R. (2005). A new approach to real time optimization of the Tennessee Eastman challenge problem. *Chemical Engineering Journal*, *112*(1), 33–44. https://doi.org/https://doi.org/10.1016/j.cej.2005.06.005

Gomez-Cabrero, D., Abugessaisa, I., Maier, D., Teschendorff, A., Merkenschlager, M., Gisel, A., Ballestar, E., Bongcam-Rudloff, E., Conesa, A., & Tegnér, J. (2014). *Data integration in the era of omics: current and future challenges*. BioMed Central.

Goodfellow, I. (2016). NIPS 2016 tutorial: Generative adversarial networks. *ArXiv*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. *Commun. ACM*, *63*(11), 139–144.

https://doi.org/10.1145/3422622

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative Adversarial Networks. *Commun. ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Gunerkar, R. S., Jalan, A. K., & Belgamwar, S. U. (2019). Fault diagnosis of rolling element bearing based on artificial neural network. *Journal of Mechanical Science and Technology*, *33*(2), 505–511. https://doi.org/10.1007/s12206-019-0103-x

Han, Z., Zhao, J., Leung, H., Ma, K. F., & Wang, W. (2021). A Review of Deep Learning Models for Time Series Prediction. *IEEE Sensors Journal*, *21*(6), 7833–7848. https://doi.org/10.1109/JSEN.2019.2923982

Hasan, M. J., Sohaib, M., & Kim, J.-M. (2019). 1D CNN-Based Transfer Learning Model for Bearing Fault Diagnosis Under Variable Working Conditions. In S. Omar, W. S. Haji Suhaili, & S. Phon-Amnuaisuk (Eds.), *Computational Intelligence in Information Systems* (pp. 13–23). Springer International Publishing.

Hauser, H., Ledermann, F., & Doleisch, H. (2002). Angular brushing of extended parallel coordinates. *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, 127–130.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

He, Y., Cowell, L., Diehl, A. D., Mobley, H. L., Peters, B., Ruttenberg, A., Scheuermann, R. H., Brinkman, R. R., Courtot, M., Mungall, C., & others. (2009). VO: vaccine ontology. *The 1st International Conference on Biomedical Ontology (ICBO 2009) Nature Precedings*, *2009*.

Hendrickx, D. M., Aerts, H. J. W. L., Caiment, F., Clark, D., Ebbels, T. M. D., Evelo, C. T., Gmuender, H., Hebels, D. G. A. J., Herwig, R., Hescheler, J., & others. (2014). diXa: a data infrastructure for chemical safety assessment. *Bioinformatics*, *31*(9), 1505–1507.

Heo, S., & Lee, J. H. (2018). Fault detection and classification using artificial neural networks. *IFAC-PapersOnLine*, *51*(18), 470–475.

Himeur, Y., Alsalemi, A., Al-Kababji, A., Bensaali, F., & Amira, A. (2020). Data fusion strategies

for energy efficiency in buildings: Overview, challenges and novel orientations. *Information Fusion*, *64*(June), 99–120. https://doi.org/10.1016/j.inffus.2020.07.003

Hoermann, S., Bach, M., & Dietmayer, K. (2018). Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2056–2063.

Hsu, C. Y., & Liu, W. C. (2020). Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, *0123456789*. https://doi.org/10.1007/s10845-020-01591-0

Huang, J. T., Li, J., & Gong, Y. (2015). An analysis of convolutional neural networks for speech recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2015-Augus*, 4989–4993. https://doi.org/10.1109/ICASSP.2015.7178920

Hurley, C. B., & Oldford, R. W. (2010). Pairwise display of high-dimensional information via eulerian tours and hamiltonian decompositions. *Journal of Computational and Graphical Statistics*, *19*(4), 861–886.

Ian Goodfellow Yoshua Bengio, A. C. (2017). The Deep Learning Book. *MIT Press*, *521*(7553), 785. https://doi.org/10.1016/B978-0-12-391420-0.09987-X

Inselberg, A. (2009). *Parallel coordinates*. Springer.

Inselberg, A., & Dimsdale, B. (1990). Parallel coordinates: a tool for visualizing multi-dimensional geometry. *Proceedings of the 1st Conference on Visualization'90*, 361–378.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 5967–5976. https://doi.org/10.1109/CVPR.2017.632

Jain, A., Smarra, F., Behl, M., & Mangharam, R. (2018). Data-driven model predictive control with regression trees-an application to building energy management. *ACM Transactions on Cyber-Physical Systems*, *2*(1), 1–21. https://doi.org/10.1145/3127023

Jain, S., Seth, G., Paruthi, A., Soni, U., & Kumar, G. (2020). Synthetic data augmentation for

surface defect detection and classification using deep learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01710-x

Jebara, T. (2012). *Machine learning: discriminative and generative* (Vol. 755). Springer Science & Business Media.

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221–231.

Jing, C., & Hou, J. (2015). SVM and PCA based fault classification approaches for complicated industrial process. *Neurocomputing*, *167*, 636–642. https://doi.org/https://doi.org/10.1016/j.neucom.2015.03.082

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9906 LNCS*, 694–711. https://doi.org/10.1007/978-3-319-46475-6_43

Jurkovic, Z., Cukor, G., Brezocnik, M., & Brajkovic, T. (2018). A comparison of machine learning methods for cutting parameters prediction in high speed turning process. *Journal of Intelligent Manufacturing*, *29*(8), 1683–1693. https://doi.org/10.1007/s10845-016-1206-1

Karagkouni, D., Paraskevopoulou, M. D., Chatzopoulos, S., Vlachos, I. S., Tastsoglou, S., Kanellos, I., Papadimitriou, D., Kavakiotis, I., Maniou, S., Skoufos, G., & others. (2017). DIANA-TarBase v8: a decade-long collection of experimentally supported miRNA--gene interactions. *Nucleic Acids Research*, *46*(D1), D239--D245.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2019). Analyzing and improving the image quality of StyleGAN. *ArXiv*, 8110–8119.

Kasuya, E. (2019). On the use of r and r squared in correlation and regression. *Ecological Research*, *34*(1), 235–236. https://doi.org/10.1111/1440-1703.1011

Kedem, B., & Fokianos, K. (2005). *Regression models for time series analysis* (Vol. 488). John Wiley \& Sons.

Khaleghi, B., Khamis, A., Karray, F. O., & Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, *14*(1), 28–44. https://doi.org/10.1016/j.inffus.2011.08.001

King, R. D., Feng, C., & Sutherland, A. (1995). Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence an International Journal*, *9*(3), 289–333.

Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. *Big Data & Society*, *1*(1), 205395171452848. https://doi.org/10.1177/2053951714528481

Kletz, T. A. (1988). *Learning from accidents in industry*. Butterworth-Heinemann.

Kourniotis, S. P., Kiranoudis, C. T., & Markatos, N. C. (2000). Statistical analysis of domino chemical accidents. *Journal of Hazardous Materials*, *71*(1–3), 239–252.

Krämer, S., & Engell, S. (2018). *Resource efficiency of processing plants: monitoring and improvement*. John Wiley & Sons.

Kusiak, A. (2020). Convolutional and generative adversarial neural networks in manufacturing. *International Journal of Production Research*, *58*(5), 1594–1604. https://doi.org/10.1080/00207543.2019.1662133

Lahat, D., Adali, T., & Jutten, C. (2015). Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. *Proceedings of the IEEE*, *103*(9), 1449–1477. https://doi.org/10.1109/JPROC.2015.2460697

Lanzetti, N., Lian, Y. Z., Cortinovis, A., Dominguez, L., Mercangöz, M., & Jones, C. (2019). Recurrent neural network based MPC for process industries. *2019 18th European Control Conference (ECC)*, 1005–1010.

Lapedes, A., & Farber, R. (1987). *Nonlinear signal processing using neural networks: Prediction and system modelling*.

Larsson, T., Hestetun, K., Hovland, E., & Skogestad, S. (2001). Self-optimizing control of a large-scale plant: The Tennessee Eastman process. *Industrial & Engineering Chemistry Research*, *40*(22), 4889–4901.

Larsson, T., & Skogestad, S. (2000). *Plantwide control-A review and a new design procedure*.

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, *6*(4), 239–242.

Lau, B. P. L., Marakkalage, S. H., Zhou, Y., Hassan, N. U., Yuen, C., Zhang, M., & Tan, U. X. (2019). A survey of data fusion in smart city applications. *Information Fusion*, *52*(January), 357–374. https://doi.org/10.1016/j.inffus.2019.05.004

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

LeCun, Y., Bengio, Y., & others. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(10), 1995.

Lee, D., Siu, V., Cruz, R., & Yetman, C. (2016). Convolutional Neural Net and Bearing Fault Analysis. *Int'l Conf. Data Mining*, 194–200. https://doi.org/https://pdfs.semanticscholar.org/6e45/f39b1e50cfd10deaabd1d786f

Lee, K. B., Cheon, S., & Kim, C. O. (2017). A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, *30*(2), 135–142. https://doi.org/10.1109/TSM.2017.2676245

Li, M. J., & Tao, W. Q. (2017). Review of methodologies and polices for evaluation of energy efficiency in high energy-consuming industry. *Applied Energy*, *187*, 203–215. https://doi.org/10.1016/j.apenergy.2016.11.039

Li, P., Chen, Z., & Zhang, J. (2020). *A Survey on Deep Learning for Multimodal Data Fusion*.

Li, Xiang, Zhang, W., Ding, Q., & Sun, J. Q. (2020). Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. *Journal of Intelligent Manufacturing*, *31*(2), 433–452. https://doi.org/10.1007/s10845-018-1456-1

Li, Xin, Liang, Y., Zhao, M., Wang, C., & Jiang, Y. (2019). Few-shot learning with generative adversarial networks based on WOA13 data. *Computers, Materials and Continua*, *60*(3), 1073–1085.

Li, Z., Wang, Y., & Wang, K. (2020). A data-driven method based on deep belief networks for backlash error prediction in machining centers. *Journal of Intelligent Manufacturing*, *31*(7),

1693–1705. https://doi.org/10.1007/s10845-017-1380-9

Lindberg, C. F., Tan, S., Yan, J., & Starfelt, F. (2015). Key Performance Indicators Improve Industrial Performance. *Energy Procedia*, *75*, 1785–1790. https://doi.org/10.1016/j.egypro.2015.07.474

Lipton, Z. C., Elkan, C., & Naryanaswamy, B. (2014). Optimal thresholding of classifiers to maximize F1 measure. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 225–239.

Liu, M. Y., Huang, X., Yu, J., Wang, T. C., & Mallya, A. (2021). Generative Adversarial Networks for Image and Video Synthesis: Algorithms and Applications. *Proceedings of the IEEE*, *109*(5), 839–862. https://doi.org/10.1109/JPROC.2021.3049196

Liu, X., Yin, G., Shao, J., Wang, X., & Li, H. (2019). Learning to predict layout-to-image conditional convolutions for semantic image synthesis. *ArXiv Preprint ArXiv:1910.06809*.

Long, M., Peng, F., & Zhu, Y. (2019). Identifying natural images and computer generated graphics based on binary similarity measures of PRNU. *Multimedia Tools and Applications*, *78*(1), 489–506.

Lv, F., Wen, C., Bao, Z., & Liu, M. (2016). Fault diagnosis based on deep learning. *Proceedings of the American Control Conference*, *2016-July*(2), 6851–6856. https://doi.org/10.1109/ACC.2016.7526751

Mandreoli, F., & Montangero, M. (2019). Dealing With Data Heterogeneity in a Data Fusion Perspective: Models, Methodologies, and Algorithms. In *Data Handling in Science and Technology* (Vol. 31). https://doi.org/10.1016/B978-0-444-63984-4.00009-0

Mart\’\in-Lopo, M. M., Boal, J., & Sánchez-Miralles, Á. (2020). A literature review of IoT energy platforms aimed at end users. *Computer Networks*, *171*, 107101.

Martens, H. (2015). Quantitative Big Data: where chemometrics can contribute. *Journal of Chemometrics*, *29*(11), 563–581. https://doi.org/10.1002/cem.2740

Mathieu, M., Couprie, C., & LeCun, Y. (2016). Deep multi-scale video prediction beyond mean square error. *4th International Conference on Learning Representations, ICLR 2016 -*

Conference Track Proceedings, *2015*, 1–14.

McAvoy, T. J., & Ye, N. (1994). Base control for the Tennessee Eastman problem. *Computers & Chemical Engineering*, *18*(5), 383–413.

McEntyre, J., & Lipman, D. (2001). PubMed: bridging the information gap. *Cmaj*, *164*(9), 1317–1319.

Meng, T., Jing, X., Yan, Z., & Pedrycz, W. (2020). A survey on machine learning for data fusion. *Information Fusion*, *57*(2), 115–129. https://doi.org/10.1016/j.inffus.2019.12.001

Mirza, M., & Osindero, S. (2014). *Conditional Generative Adversarial Nets*. 1–7. http://arxiv.org/abs/1411.1784

Narayanasamy, R., & Padmanabhan, P. (2012). Comparison of regression and artificial neural network model for the prediction of springback during air bending process of interstitial free steel sheet. *Journal of Intelligent Manufacturing*, *23*(3), 357–364. https://doi.org/10.1007/s10845-009-0375-6

National Inventory Report. (2019). *GREENHOUSE GAS SOURCES AND SINKS IN CANADA CANADA'S SUBMISSION TO THE UNITED NATIONS FRAMEWORK CONVENTION ON CLIMATE CHANGE Executive Summary*.

Neurohive. (n.d.). *FaceStyleGAN: GAN Network Generates Style Portraits in Snapchat*. https://neurohive.io/en/news/facestylegan-gan-network-generates-style-portraits-in-snapchat/

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems*, 841–848.

Olmschenk, G., Zhu, Z., & Tang, H. (2019). Generalizing semi-supervised generative adversarial networks to regression using feature contrasting. *Computer Vision and Image Understanding*, *186*(June), 1–12. https://doi.org/10.1016/j.cviu.2019.06.004

Olsina, L., & Martin, M. de los A. (2004). Ontology for software metrics and indicators. *J. Web Eng.*, *2*(4), 262–281.

Om, H., & Kundu, A. (2012). A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, 131–136.

Pan, R. (2010). H olt--W inters Exponential Smoothing. *Wiley Encyclopedia of Operations Research and Management Science*.

Park, T., Liu, M.-Y., Wang, T.-C., & Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2337–2346.

Parmenter, D. (2020). *Key perfomance indicator developing, implementing and using winning KPIs*. John Wiley & Sons.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, D., Liu, Z., Wang, H., Qin, Y., & Jia, L. (2019). A novel deeper one-dimensional CNN with residual learning for fault diagnosis of wheelset bearings in high-speed trains. *IEEE Access*, *7*, 10278–12093. https://doi.org/10.1109/ACCESS.2018.2888842

Petzold, B., Roggendorf, M., Rowshankish, K., & Sporleder, C. (2020). Designing data governance that delivers value. *Mc Kinsey Digital*, *June*. https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/designing-data-governance-that-delivers-value

Qi, X., Yuan, Z., & Han, X. (2015). Diagnosis of misalignment faults by tacholess order tracking analysis and RBF networks. *Neurocomputing*, *169*, 439–448. https://doi.org/10.1016/j.neucom.2014.09.088

Qiu, X., Zhang, L., Ren, Y., Suganthan, P., & Amaratunga, G. (2014). Ensemble deep learning for regression and time series forecasting. *IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - CIEL 2014: 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning, Proceedings*. https://doi.org/10.1109/CIEL.2014.7015739

Ragab, A., El-Koujok, M., Poulin, B., Amazouz, M., & Yacout, S. (2018). Fault diagnosis in

industrial chemical processes using interpretable patterns based on Logical Analysis of Data. *Expert Systems with Applications*, *95*, 368–383. https://doi.org/https://doi.org/10.1016/j.eswa.2017.11.045

Ragab, A., Koujok, M. El, Ghezzaz, H., Amazouz, M., Ouali, M.-S., & Yacout, S. (2019). Deep understanding in industrial processes by complementing human expertise with interpretable patterns of machine learning. *Expert Systems with Applications*, *122*, 388–405. https://doi.org/https://doi.org/10.1016/j.eswa.2019.01.011

Ragab, A., Ouali, M. S., Yacout, S., & Osman, H. (2016). Remaining useful life prediction using prognostic methodology based on logical analysis of data and Kaplan–Meier estimation. *Journal of Intelligent Manufacturing*, *27*(5), 943–958. https://doi.org/10.1007/s10845-014-0926-3

Ragab, A., Yacout, S., Ouali, M. S., & Osman, H. (2019). Prognostics of multiple failure modes in rotating machinery using a pattern-based classifier and cumulative incidence functions. *Journal of Intelligent Manufacturing*, *30*(1), 255–274. https://doi.org/10.1007/s10845-016-1244-8

Ricker, N. L. (1996). Decentralized control of the Tennessee Eastman challenge process. *Journal of Process Control*, *6*(4), 205–221.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., & Waldman-Brown, A. (2019). Tackling climate change with machine learning. *ArXiv Preprint ArXiv:1906.05433*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015* (pp. 234–241). Springer International Publishing.

Sakpal, M. (2021). *12 Actions to Improve Your Data Quality*. https://www.gartner.com/smarterwithgartner/how-to-improve-your-data-quality

Sanderson, C., & Gruen, R. (2006). *Analytical models for decision-making*. McGraw-Hill Education (UK).

Santamaria, R., Therón, R., & Quintales, L. (2008). A visual analytics approach for understanding biclustering results from microarray data. *BMC Bioinformatics*, *9*(1), 247.

Schat, E., van de Schoot, R., Kouw, W. M., Veen, D., & Mendrik, A. M. (2020). The data representativeness criterion: Predicting the performance of supervised classification based on data set similarity. *PLoS ONE*, *15*(8 August), 1–16. https://doi.org/10.1371/journal.pone.0237009

Schmidt, C., Li, W., Thiede, S., Kornfeld, B., Kara, S., & Herrmann, C. (2016). Implementing Key Performance Indicators for Energy Efficiency in Manufacturing. *Procedia CIRP*, *57*, 758–763. https://doi.org/10.1016/j.procir.2016.11.131

Shao, S., Wang, P., & Yan, R. (2019). Generative adversarial networks for data augmentation in machine fault diagnosis. *Computers in Industry*, *106*, 85–93. https://doi.org/10.1016/j.compind.2019.01.001

Shao, S. Y., Sun, W. J., Yan, R. Q., Wang, P., & Gao, R. X. (2017). A Deep Learning Approach for Fault Diagnosis of Induction Motors in Manufacturing. *Chinese Journal of Mechanical Engineering (English Edition)*, *30*(6), 1347–1356. https://doi.org/10.1007/s10033-017-0189-y

Shen, Y., Yang, F., Habibullah, M. S., Ahmed, J., Das, A. K., Zhou, Y., & Ho, C. L. (2020). Predicting tool wear size across multi-cutting conditions using advanced machine learning techniques. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01625-7

Siirtola, H., & Räihä, K.-J. (2006). Interacting with parallel coordinates. *Interacting with Computers*, *18*(6), 1278–1309.

Simmonds, A., Sandilands, P., & Van Ekert, L. (2004). An ontology for network security attacks. *Asian Applied Computing Conference*, 317–323.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556*.

Soualhi, M., El Koujok, M., Nguyen, K. T. P., Medjaher, K., Ragab, A., Ghezzaz, H., Amazouz, M., & Ouali, M.-S. (2021). Adaptive prognostics in a controlled energy conversion process

based on long-and short-term predictors. *Applied Energy*, *283*, 116049.

Sridharan, M. (2017). *Using governance to achieve data quality objectives*. https://www.bloomberg.com/professional/blog/using-governance-achieve-data-quality-objectives/

Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015). Unsupervised learning of video representations using lstms. *International Conference on Machine Learning*, 843–852.

Talbot, D., & Boiral, O. (2013). Can we trust corporates GHG inventories? An investigation among Canada's large final emitters. *Energy Policy*, *63*, 1075–1085.

Tang, H., Qi, X., Xu, D., Torr, P. H. S., & Sebe, N. (2020). Edge guided GANs with semantic preserving for semantic image synthesis. *ArXiv Preprint ArXiv:2003.13898*.

Telea, A. C. (2007). Data Visualization: Principles and practice. In *Data Visualization: Principles and Practice*. https://doi.org/10.1201/b10679

Tian, J., Morillo, C., Azarian, M. H., & Pecht, M. (2016). Motor Bearing Fault Detection Using Spectral Kurtosis-Based Feature Extraction Coupled With K-Nearest Neighbor Distance Analysis. *IEEE Transactions on Industrial Electronics*, *63*(3), 1793–1803. https://doi.org/10.1109/TIE.2015.2509913

Tidriri, K., Chatti, N., Verron, S., & Tiplica, T. (2016). Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual Reviews in Control*, *42*, 63–81.

Uguz, S., & Ipek, O. (2021). Prediction of the parameters affecting the performance of compact heat exchangers with an innovative design using machine learning techniques. *Journal of Intelligent Manufacturing*, *2013*. https://doi.org/10.1007/s10845-020-01729-0

Uysal, M. P., & Sogut, M. Z. (2017). An integrated research for architecture-based energy management in sustainable airports. *Energy*, *140*, 1387–1397.

Vakkilainen, E., & others. (2005). *Kraft recovery boilers--Principles and practice*.

Vasara, P. (2001). Scandinavia: Through different eyes: environmental issues in Scandinavia and North America. *Tappi Journal*, *84*(6), 46–49.

Vlachos, I. P. (2014). A hierarchical model of the impact of RFID practices on retail supply chain performance. *Expert Systems with Applications*, *41*(1), 5–15. https://doi.org/10.1016/j.eswa.2013.07.006

Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Generating videos with scene dynamics. *Advances in Neural Information Processing Systems*, *29*, 613–621.

Wald, L. (1999). *Definitions and terms of reference in data fusion*.

Wang, Jin, Yang, Y., Wang, T., Sherratt, R. S., & Zhang, J. (2020). Big data service architecture: a survey. *Journal of Internet Technology*, *21*(2), 393–405.

Wang, Jinjiang, Yan, J., Li, C., Gao, R. X., & Zhao, R. (2019). Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction. *Computers in Industry*, *111*, 1–14. https://doi.org/10.1016/j.compind.2019.06.001

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8798–8807.

Wang, Y., Zhou, J., Zheng, L., & Gogu, C. (2020). An end-to-end fault diagnostics method based on convolutional neural network for rotating machinery with multiple case studies. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01671-1

Wang, Z., & Bovik, A. C. (2002). A universal image quality index. *IEEE Signal Processing Letters*, *9*(3), 81–84.

Wegman, E. J. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, *85*(411), 664–675.

White, F. E. (1991). JDL, data fusion lexicon. *Technical Panel for C*, *3*, 15.

Wilke, C. O. (2019). Fundamentals of Data Visualization. In *Serial Mentor*.

Wong, P. C., & Bergeron, R. D. (1996). Multiresolution multidimensional wavelet brushing. *Proceedings of Seventh Annual IEEE Visualization'96*, 141–148.

Wu, A. (n.d.). *A Chat with Andrew on MLOps: From Model-Centric to Data-Centric AI. 2021*.

Wu, H., & Zhao, J. (2018). Deep convolutional neural network model based chemical process fault

diagnosis. *Computers and Chemical Engineering*, *115*, 185–197. https://doi.org/10.1016/j.compchemeng.2018.04.009

Wu, J., Zhang, C., Xue, T., Freeman, W. T., & Tenenbaum, J. B. (2016). Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. *Advances in Neural Information Processing Systems*, *Nips*, 82–90.

Xia, C., Pan, Z., Polden, J., Li, H., Xu, Y., & Chen, S. (2021). Modelling and prediction of surface roughness in wire arc additive manufacturing using machine learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01725-4

Xu, Y., Pei, J., & Lai, L. (2017). Deep Learning Based Regression and Multiclass Models for Acute Oral Toxicity Prediction with Automatic Chemical Feature Extraction. *Journal of Chemical Information and Modeling*, *57*(11), 2672–2685. https://doi.org/10.1021/acs.jcim.7b00244

Yin, S., Ding, S. X., Haghani, A., Hao, H., & Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control*, *22*(9), 1567–1581. https://doi.org/https://doi.org/10.1016/j.jprocont.2012.06.009

Yourtechdiet. (n.d.). *Top Tools in Generative Adversarial Networks*. https://www.yourtechdiet.com/blogs/generative-adversarial-networks-tools/

Yuan, X., Li, L., Shardt, Y. A. W., Wang, Y., & Yang, C. (2021). Deep Learning with Spatiotemporal Attention-Based LSTM for Industrial Soft Sensor Model Development. *IEEE Transactions on Industrial Electronics*, *68*(5), 4404–4414. https://doi.org/10.1109/TIE.2020.2984443

Zagrebina, S. A., Mokhov, V. G., & Tsimbol, V. I. (2019). Electrical energy consumption prediction is based on the recurrent neural network. *Procedia Computer Science*, *150*, 340–346.

Zakharov, E., Shysheya, A., Burkov, E., & Lempitsky, V. (2019). Few-shot adversarial learning of realistic neural talking head models. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 9458–9467. https://doi.org/10.1109/ICCV.2019.00955

Zhang, D., Li, Q., Yang, G., Li, L., & Sun, X. (2017). Detection of image seam carving by using

weber local descriptor and local binary patterns. *Journal of Information Security and Applications*, *36*, 135–144.

Zhang, J., Zhong, S., Wang, T., Chao, H.-C., & Wang, J. (2020). Blockchain-based systems and applications: a survey. *Journal of Internet Technology*, *21*(1), 1–14.

Zhang, Qing, Gao, J., Dong, H., & Mao, Y. (2018). WPD and DE/BBO-RBFNN for solution of rolling bearing fault diagnosis. *Neurocomputing*, *312*, 27–33. https://doi.org/10.1016/j.neucom.2018.05.014

Zhang, Qingchen, Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, *42*, 146–157.

Zhang, S., Zhang, S., Wang, B., & Habetler, T. G. (2019). Machine learning and deep learning algorithms for bearing fault diagnostics-a comprehensive review. *ArXiv Preprint ArXiv:1901.08247*.

Zhang, W., Li, X., Jia, X. D., Ma, H., Luo, Z., & Li, X. (2020). Machinery fault diagnosis with imbalanced data using deep generative adversarial networks. *Measurement: Journal of the International Measurement Confederation*, *152*. https://doi.org/10.1016/j.measurement.2019.107377

Zhang, Z., & Zhao, J. (2017). A deep belief network based fault diagnosis model for complex chemical processes. *Computers and Chemical Engineering*, *107*, 395–407. https://doi.org/10.1016/j.compchemeng.2017.02.041

Zhao, D., Ivanov, M., Wang, Y., & Du, W. (2020). Welding quality evaluation of resistance spot welding based on a hybrid approach. *Journal of Intelligent Manufacturing*, *0123456789*. https://doi.org/10.1007/s10845-020-01627-5

Zhao, M., Liu, X., Yao, X., & He, K. (2020). Better visual image super-resolution with Laplacian pyramid of generative adversarial networks. *CMC-COMPUTERS MATERIALS \& CONTINUA*, *64*(3), 1601–1614.

Zheng, H., Liao, H., Chen, L., Xiong, W., Chen, T., & Luo, J. (2020). Example-guided image synthesis across arbitrary scenes using masked spatial-channel attention and self-supervision. *ArXiv Preprint ArXiv:2004.10024*.

Zhou, P., Ang, B. W., & Zhou, D. Q. (2012). Measuring economy-wide energy efficiency performance: A parametric frontier approach. *Applied Energy*, *90*(1), 196–200. https://doi.org/10.1016/j.apenergy.2011.02.025

Zhou, W., Li, X., Yi, J., & He, H. (2019). A Novel UKF-RBF Method Based on Adaptive Noise Factor for Fault Diagnosis in Pumping Unit. *IEEE Transactions on Industrial Informatics*, *15*(3), 1415–1424. https://doi.org/10.1109/TII.2018.2839062

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2223–2232.

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., & Shechtman, E. (2017). Multimodal Image-to-Image Translation by Enforcing Bi-Cycle Consistency. *Advances in Neural Information Processing Systems*, 465–476.

Zhu, L., Johnsson, C., Mejvik, J., Varisco, M., & Schiraldi, M. (2018). Key performance indicators for manufacturing operations management in the process industry. *IEEE International Conference on Industrial Engineering and Engineering Management*, *2017-Decem*, 969–973. https://doi.org/10.1109/IEEM.2017.8290036

Zitnik, M., Nguyen, F., Wang, B., Leskovec, J., Goldenberg, A., & Hoffman, M. M. (2019). Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, *50*(September 2018), 71–91. https://doi.org/10.1016/j.inffus.2018.09.012

CHAPTER 4 ARTICLE 2: MULTI-OUTPUT REGRESSION USING POLYGON GENERATION AND CONDITIONAL GENERATIVE ADVERSARIAL NETWORKS

Mohamed Tarek Mohamed Elhefnawy, Mohamed-Salah Ouali, Ahmed Ragab

Submitted to: EXPERT SYSTEMS WITH APPLICATIONS

Manuscript Number: ESWA-D-21-01672

4.1 Abstract

This paper proposes an innovative multi-output regression method that processes and converts the numeric data variables into representative images (polygons) to build accurate predictive models in industrial applications with several dependent variables (responses). In this method, polygon images are generated from both the inputs and outputs of numeric data. The images representing the data inputs are then translated into those representing the outputs by training a conditional generative adversarial network (cGAN). The output images of the trained cGAN are then mapped into the outputs to get back the predicted numeric values. The advantage of the proposed method is that it makes use of the breakthrough of deep generative modeling to learn the true distribution of complex data, which is difficult to determine in many industrial applications. This is attributed to the fact that the generated polygons express all interrelationships between the data variables in the form of trustworthy representational images used to train the cGAN model. The performance of the proposed method was validated successfully using a complex industrial dataset acquired from a black liquor recovery boiler (BLRB) in a Kraft pulp & paper mill located in Canada. Three key performance indicators (KPIs), one economic and two environmental, are used as regression outputs in the BLRB dataset. The results of the proposed method demonstrate better performance than other comparable machine learning regression methods.

SYMBOL	DEFINITION
$\boldsymbol{X_j}$	Variable j
$\boldsymbol{Y_h}$	Output h
$\hat{\boldsymbol{q}}$	The unit vector of x direction
$\hat{\boldsymbol{l}}$	The unit vector of y direction
$\overline{\boldsymbol{X_j}}$	Mean of variable j
$\overline{\boldsymbol{Y_h}}$	Mean of output h
$\boldsymbol{\delta_j}$	Standard deviation of variable j
$\boldsymbol{\delta_h}$	Standard deviation of output h
$\boldsymbol{x_{kj}}$	Value of variable j for observation k
$\boldsymbol{y_{kh}}$	Value of output h for observation k
$\boldsymbol{Z_{kj}}$	Standardized value of variable j for observation k
$\boldsymbol{Z_{kh}}$	Standardized value of output h for observation k
$\widehat{\boldsymbol{X_j}}$	Unit vector of the polygon side representing variable j
$\widehat{\boldsymbol{Y_h}}$	Unit vector of the polygon side representing output h
$\overrightarrow{\boldsymbol{X_j}}$	Point coordinates of the zero standardized value of the variable X_j
$\overrightarrow{\boldsymbol{Y_h}}$	Point coordinates of the zero standardized value of the output Y_h
$\overrightarrow{\boldsymbol{X_j^k}}$	Point coordinates of the standardized values of observation k for variable X_j
$\overrightarrow{\boldsymbol{Y_h^k}}$	Point coordinates of the standardized values of observation k for output Y_h
$\boldsymbol{\alpha_j}$	The underestimation parameter of each KPI_j in the penalty function
$\boldsymbol{\beta_j}$	The overestimation parameter of each KPI_j in the penalty function
$\boldsymbol{L_{acc}^j(t_k)}$	The penalty function of each KPI_j at instance t_k
$\boldsymbol{L_{AP}^j}$	The average penalty function of each KPI_j
$\boldsymbol{L_{model}}$	The total average penalty function of each regression model

4.2 Introduction

Process industries are among the industrial systems that contribute to a large amount of harmful greenhouse gas (GHG) emissions (Talbot & Boiral, 2013). These emissions badly affect the environment and cause many serious issues, such as global warming, air pollution and climate change. In terms of energy costs, process industries are the largest energy-consuming sectors of the industry's total delivered energy (M. J. Li & Tao, 2017). Examples of such processes are found in the oil & gas industry, steel & iron production, cement industry, chemical processes, and pulp & paper mills. There are many reasons for GHG emissions and the consumption of such excessive energy. Among the reasons is the inefficient monitoring and control of the operation of such industrial processes. The performance of these processes is monitored through a set of key

performance indicators (KPIs) that refer to the overall system health state. Inefficient monitoring of KPIs leads to high-frequency abnormal/ faulty situations, poor adaptability to system disturbances that lead to undesired downtime, and unnecessary, excessive maintenance activities (Andersson & Thollander, 2019).

Therefore, there is an urgent need to develop an efficient monitoring system that accurately predicts these KPIs. This would provide the process expert with better knowledge to manage abnormal operations and to reduce any related environmental and economic losses in the process by taking the most appropriate actions (Rolnick et al., 2019).

Several procedures should be followed in the definition and selection of KPIs in industrial systems (Parmenter, 2020; Schmidt et al., 2016). In fact, this is an exhaustive task that involves experts examining several aspects, including energy efficiency, environment, economy, safety and others (Krämer & Engell, 2018). Several studies have been conducted on the definition, selection and benchmarking of KPIs in the industrial and standardization context (Lindberg et al., 2015; Schmidt et al., 2016; L. Zhu et al., 2018).

The prediction of interacting KPIs poses several challenges for many industries in which the aim is to maximize the global values of their assets. Model based regression methods were commonly used in the literature to predict KPIs in industrial systems. A method called stochastic frontier analysis (SFA) is adopted in (P. Zhou et al., 2012) to estimate the energy-efficiency index. A hierarchical analytical model is used in (Vlachos, 2014) to estimate retail supply chain performance. The KPIs are estimated in (Fuentes et al., 2016) using a non-parametric method called multiple-stage data envelopment analysis (DEA). However, these prediction methods rely on assumptions that are heavily based on human expertise and the application at hand. It is difficult, expensive, and time-consuming to obtain such detailed models for a wide spectrum of industrial processes with hundreds and even thousands of interacting variables. This hinders the application of this type of KPI prediction method to efficiently monitor such processes (Sanderson & Gruen, 2006; Tidriri et al., 2016).

Fortunately, process industries are rich with data collected from the enormous number of sensors equipped everywhere in the plants. The resulting numerical data is an important asset to build multi-output regression methods that are proven to be more accurate compared to the model-

based methods (A. Jain et al., 2018). In the literature, machine learning (ML) regression methods, such as support vector regression (SVR), random forest (RF), multi-layer perceptron (MLP), adaptive neuro-fuzzy inference system (ANFIS), gradient boosting regression (GBR) and linear regression are used to predict the KPIs of various industrial systems (Jurkovic et al., 2018; Shen et al., 2020; Uguz & Ipek, 2021; Xia et al., 2021; D. Zhao et al., 2020). These methods are based on the concept of learning from data to represent the complex relationships between the interacting variables and their KPIs. However, most of the existing ML regression methods make unrealistic assumptions about the data distributions. These methods show some limitations in fully capturing and expressing the true distribution of the non-stationary process data, as a result of several factors, including fast system dynamics and others.

Deep learning (DL) is proposed by several researchers in playing a key role in tackling regression problems (Qiu et al., 2014; Xu et al., 2017). An important field of DL is the generative DL modeling, which provides the ability to capture and represent high-dimensional distributions in a wide variety of engineering domains (Goodfellow, 2016). The most common example is the generative adversarial networks (GANs) that have proven their high effectiveness, particularly in computer vision problems (Goodfellow et al., 2020). This is one of the breakthroughs in artificial intelligence; recent advances have demonstrated how it is possible for GANs to learn the true distribution of complex industrial data (Kusiak, 2020). This motivates us to make use of GANs and combine it with an efficient preprocessing method based on polygon generation (PG), proposed in (Elhefnawy et al., 2021a), which has shown promising results in mapping numerical observations into polygon images. In this paper, we used a specific type of GANs, called conditional (cGAN), which was proposed in (Isola et al., 2017) for image-to-image translation. In the proposed PGcGAN method, we adopt the Pix2Pix model recommended in that work to convert the polygon images that represent the data input variables into another set of polygon images that represent the numeric outputs (i.e., the KPIs). An image processing procedure is followed to convert the images representing the KPIs back to numerical values. The PGcGAN method makes use of both the powerful representation of the PG technique and the breakthrough of the deep generative modeling. By combining these two approaches, an accurate and robust multi-output regression technique is developed by representing the data as polygons using the PG technique, then efficiently mapping these polygons (inputs) into numerical values (outputs) through the cGAN. This aims at

maximizing the knowledge extracted from the industrial data. The proposed method is validated successfully using a real case study representing a piece of critical equipment at a Kraft pulp & paper mill located in Canada: the black liquor recovery boiler (BLRB). The results obtained demonstrate that the proposed method outperforms other common ML regression methods in terms of KPIs prediction accuracy.

The rest of this paper is organized as follows. Section 4.3 provides a background on polygon generation, generative modeling, GAN, cGAN and image-to-image translation, in addition to some related work in computer vision and industrial processes. Section 4.4 presents the proposed method along with illustrations of its steps. Section 4.5 shows the real case study (BLRB) used to validate the proposed method and the experimental setup. Section 4.6 discusses the results for performance comparison, remarks on the proposed method and some insights for future work. Finally, Section 4.7 concludes the paper.

4.3 Background & Related Work

This section discusses the background and related work of the two main blocks used in the proposed method to tackle the problem of KPI prediction in the industry. The first one is the polygon generation method, which converts the numeric data into images, and the second one is the cGAN, which translates images representing data inputs into other images representing numeric outputs.

4.3.1 Polygon Generation from Numerical Data

The data preprocessing method proposed in (Elhefnawy et al., 2021a) along with the convolutional neural networks (CNNs) are used to classify challenging faults in a reboiler system in a thermomechanical pulp (TMP) mill. It outperforms other common machine learning and deep learning fault classifiers. The proposed preprocessing method converts numerical data into images of polygons. In this paper, we are motivated to adopt the polygon generation technique to be used for multi-output regression problems that are found in industrial applications with KPIs that have complex and ill-defined distributions. The steps for polygon generation are explained in a comprehensive way via an illustrative example in (Elhefnawy et al., 2021a).

In what follows, we recall and summarize these steps through another example with five numeric input variables and three outputs. As shown in Figure 4.1, each polygon side in the figure represents a numeric data input variable**.** The standardized values Z_{kj} of observation k are calculated using Eq. (1) for the variable X_j, where $j = 1,2,..,5$ in this example.

$$Z_{kj} = \frac{x_{kj} - \overline{X_j}}{\delta_j} \qquad (1)$$

Where x_{kj} is the numeric value of observation k for the variable X_j, $\overline{X_j}$ and δ_j are their mean value and standard deviation, respectively. As shown in Figure 4.1, the point coordinates (in orange) on each side of the polygon represent the standardized values Z_{kj}, calculated using Eq. (2).

$$\overrightarrow{X_j^k} = \overrightarrow{X_j} + \left(Z_{kj} * \widehat{X_j}\right) \qquad (2)$$

where $\overrightarrow{X_j}$ represents the point coordinates (in blue) of the zero standardized value of the variable X_j and $\widehat{X_j}$ represents the unit vector of each polygon side. Table 4.1 shows the values calculated using Equations 1 and 2 for the toy example shown in Figure 4.1.

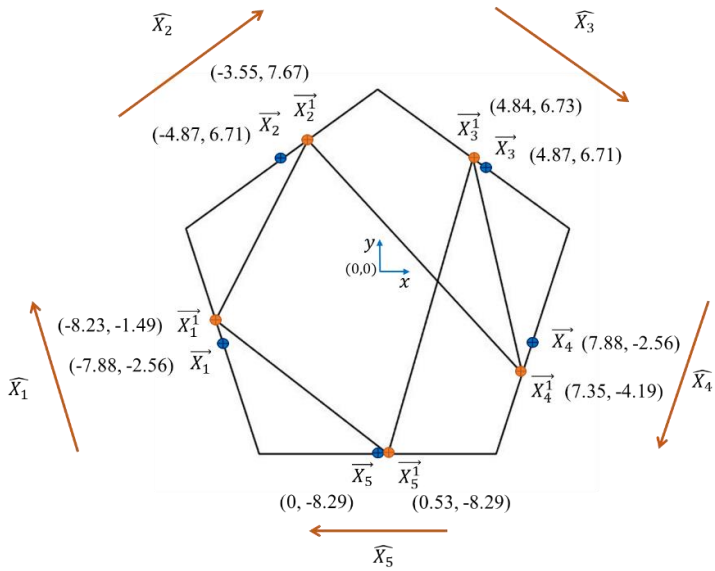


Figure 4.1 - A polygon generated from a numeric observation of five data variables using the method proposed in (Elhefnawy et al., 2021a), where $\overrightarrow{X_j^k}$ represents the point coordinates of standardized values of observation k for each variable X_j, $\overrightarrow{X_j}$ represents the point coordinates of the zero standardized value of the variable X_j and $\widehat{X_j}$ represents the unit vector of each polygon side. All variables are numbered in a clockwise direction.

Table 4.1 - Calculations of point coordinates $\overrightarrow{X_j^1}$ on the sides of the polygon for a numeric observation with five variables shown in Figure 1, where $\hat{q}$ and $\hat{l}$ are the unit vectors of x and y directions, respectively

j	x_{1j}	$\overline{X_j}$	δ_j	Z_{1j}	$\widehat{X_j}$	$\overrightarrow{X_j}$	$\overrightarrow{X_j^1}$
1	69.44	68.54	0.8	1.13	$-0.31\,\hat{q} + 0.95\,\hat{l}$	$-7.88\,\hat{q} - 2.56\,\hat{l}$	$-8.23\,\hat{q} - 1.49\,\hat{l}$
2	136.95	118.12	11.48	1.64	$0.81\,\hat{q} + 0.59\,\hat{l}$	$-4.87\,\hat{q} + 6.71\,\hat{l}$	$-3.55\,\hat{q} + 7.67\,\hat{l}$
3	860.21	860.36	3.83	-0.04	$0.81\,\hat{q} - 0.59\,\hat{l}$	$4.87\,\hat{q} + 6.71\,\hat{l}$	$4.84\,\hat{q} + 6.73\,\hat{l}$
4	16.8	11.42	3.13	1.72	$-0.31\,\hat{q} - 0.95\,\hat{l}$	$7.88\,\hat{q} - 2.56\,\hat{l}$	$7.35\,\hat{q} - 4.19\,\hat{l}$
5	4081.08	4120.38	74.16	-0.53	$-1\,\hat{q} + 0\,\hat{l}$	$0\,\hat{q} - 8.29\,\hat{l}$	$0.53\,\hat{q} - 8.29\,\hat{l}$

Similarly, the polygon generation procedure is applied to the data outputs. Figure 4.2 shows the three outputs, where each polygon side represents the output Y_h, where $h = 1,2,3$. Table 4.2 shows the calculations of the point coordinates (in orange) $\overrightarrow{Y_h^k}$ of the standardized outputs Z_{kh}, where y_{kh} is the numeric value of observation k for the output Y_h, $\overline{Y_h}$ and δ_h are their mean value and standard deviation, respectively, $\overrightarrow{Y_h}$ represents the point coordinates (in blue) of the zero standardized value of the output Y_h and $\widehat{Y_h}$ represents the unit vector of each polygon side.

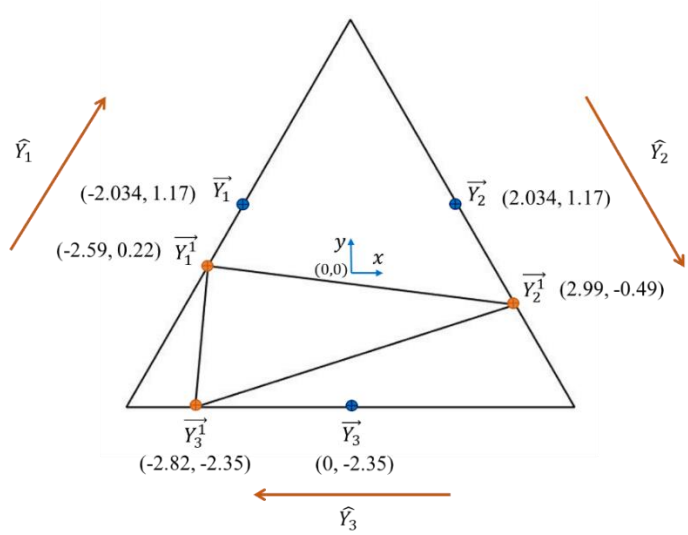


Figure 4.2 - A polygon generated from a numeric observation of three outputs using the method proposed in (Elhefnawy et al., 2021a), where $\overrightarrow{Y_h^k}$ represents the point coordinates of standardized values of observation k for each output Y_h, $\overrightarrow{Y_h}$ represents the point coordinates of the zero standardized value of the output Y_h and $\widehat{Y_h}$ represents the unit vector of each polygon side. All outputs are numbered in clockwise direction.

Table 4.2 - Calculations of point coordinates $\overrightarrow{Y_h^1}$ on polygon sides for a numeric observation with three outputs shown in Figure 2, where $\hat{q}$ and $\hat{l}$ are the unit vectors of the x and y directions, respectively

h	y_{1h}	$\overline{Y_h}$	δ_h	Z_{1h}	$\widehat{Y_h}$	$\overrightarrow{Y_h}$	$\overrightarrow{Y_h^1}$
1	3.28	3.46	0.16	-1.10	$-2.034\,\hat{q} + 1{,}17\,\hat{l}$	$0.5\,\hat{q} + 0.87\,\hat{l}$	$-2.59\,\hat{q} + 0.22\,\hat{l}$
2	94.56	23.88	36.81	1.92	$2.034\,\hat{q} + 1{,}17\,\hat{l}$	$0.5\,\hat{q} - 0.87\,\hat{l}$	$2.99\,\hat{q} - 0.49\,\hat{l}$
3	80.74	15.17	23.25	2.82	$0\,\hat{q} - 2.35\,\hat{l}$	$-1\,\hat{q} + 0\,\hat{l}$	$-2.82\,\hat{q} - 2.35\,\hat{l}$

To recap, Figure 4.1 and Figure 4.2 show one possible connection between the points on polygon sides for one numeric observation in the data. This means that these polygons can represent different connections of the points ordered on their sides. All possible connections between the points are found using Hamiltonian cycles according to the number of polygon sides (Elhefnawy et al., 2021a). The method generates these polygon images using the algorithm in (Hurley & Oldford, 2010; Wegman, 1990) for each numerical observation. Accordingly, all interrelations between data variables and outputs in each numeric observation are represented through a number of representative polygon images.

4.3.2 Generative Modeling: The Generative Adversarial Networks (GANs)

Discriminative models are used to distinguish between classes in cases of classification problems or to predict continuous outputs in cases of regression problems. Mathematically, a discriminant model (function) tries to predict output Y using a set of variables X or estimate the probability $P(Y|X)$ of the output Y given X. On the other hand, generative modeling tries to capture the probability distribution $P(X|Y)$ of X given Y (Jebara, 2012).

GANs are first introduced as generative modeling technique in (Goodfellow et al., 2014) to implicitly learn the underlying data distribution. GANs are composed of two main components: generator and discriminator, as shown in Figure 4.3. They are two neural networks that are trained

in an alternating fashion. The generator acts as a decoder that tries to produce realistic images where its input is simply a random noise (fake images). The role of the discriminator is to simultaneously distinguish between these fake images and a set of real images. The generator and discriminator compete at the same time and that is why they are adversarial. Over time, the generator can succeed to reconstruct fake images that are indistinguishable from the real ones. This process continues until the discriminator is no longer needed.

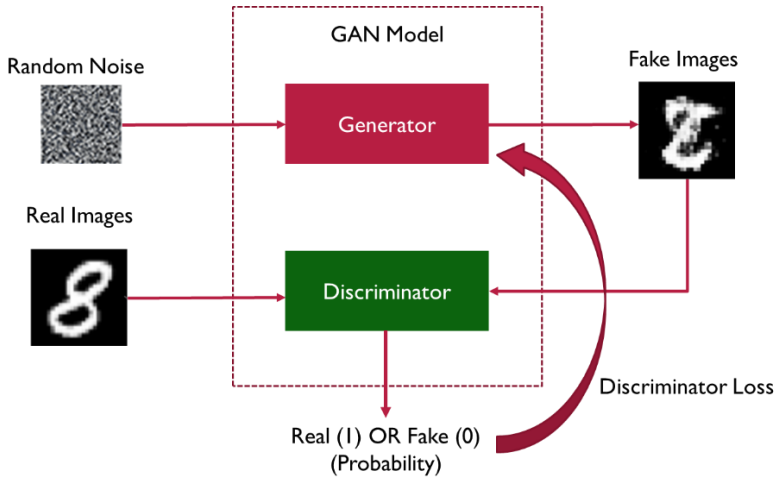


Figure 4.3 - A simplified schematic of Generative Adversarial Network (GAN)

More specifically, during the training of a discriminator, the generator produces fake images, then the discriminator can learn to distinguish between the real and fake images, while in this case the generator training is freezed. During the training of a generator, the discriminator decides whether the images are real or fake and, based on its output (discriminative losses), the generator can determine which direction it should go in to learn how to produce more realistic images. In this case, the discriminator training is freezed. The generator and discriminator should always improve each other. If there is a superior generator, it would generate 100% real images, and then there is no way to improve the discriminator. If there is a superior discriminator, it would inspect all generated images as 100% fake, and then there would be no way to improve the generator. The latter case is more common than the former because the job of the discriminator is much easier than that of the generator. This is because the generator has to model the entire space, whereas the discriminator just figures out whether the images are real or fake.

In the computer vision domain, GANs have made significant progress in terms of generating realistic images, such as the images of human faces and animals in (Karras et al., 2019). Additionally, GANs are able to animate portraits (e.g. *The Mona Lisa*) using the motion of real persons' faces (Zakharov et al., 2019). Moreover, they can be used for obtaining high-resolution image using a low-resolution single image or a set of low-resolution images (Fu et al., 2020; M. Zhao et al., 2020). Furthermore, 3D-GANs can generate 3D objects as in (J. Wu et al., 2016). Prominent companies have started using GANs for different tasks; for instance, Google is using them for text generation along with images, IBM is using them for data augmentation to generate synthetic examples in cases where there is little available data. Despite of the need of large datasets for training GANs, recent works are proposed for few-shot learning using GANs (Xin Li et al., 2019). Moreover, TikTok and Snapchat are using GANs for creative image filters that anyone can use nowadays (Neurohive; Yourtechdiet)

In the industrial context, GANs have recently been used in intelligent fault diagnosis to address the issue of the limited and imbalanced data (Cao et al., 2018; W. Zhang et al., 2020). To overcome the challenge of the time-consuming and costly acquisition of large annotated data, process industries take advantage of GANs to play a key role in data augmentation (S. Jain et al., 2020; S. Shao et al., 2019). A regression method is proposed in (Olmschenk et al., 2019) in computer vision problems such as age estimation and crowd counting from single images using semi-supervised GANs. Another regression method is proposed in (Aggarwal et al., 2019) to use conditional GANs in problems with single output datasets. In this paper, we use the conditional GAN for our multi-output regression method.

4.3.3 Conditional GAN & Image-to-image Translation

The conditional GAN (cGAN) is a significant contribution in generative modeling. They were first introduced in (Mirza & Osindero, 2014). They are used for image translation from one domain to another (Park et al., 2019). Unlike the unconditional GANs, cGANs allow an example to be generated from a specific class from a given dataset. As a result, with conditional generation, one can train the GAN with datasets of labels from different classes, while in the case of unconditional generation, no labels are needed. The input to the generator in a cGAN is actually a concatenated vector comprising the noise in addition to the encoded one-hot class

information (Park et al., 2019). The discriminator, in a similar way as the GAN, takes the examples, but these examples are paired with class information to determine whether they are either real or fake representations of a particular class. The classes are encoded based on one-hot matrices, in addition to image channels.

Image-to-image translation or paired image translation is an application of cGANs where an input image is conditioned to get another direct output image. The Pix2Pix is an example of the cGAN architectures adapted to be used for this paired image translation (Isola et al., 2017). Unlike the traditional cGAN, the entire image is fed as an input instead of the class vector. A simple schematic diagram of the Pix2Pix architecture that translates satellite images into Google Map images is shown in Figure 4.4. The generator in the figure is an encoder-decoder architecture trained using pairs of images to generate fake Google Map images based on real satellite images. Given a pair of images, the discriminator tries to figure out whether this pair is a fake or real, and then provides feedback to the generator to better synthesize more realistic images. According to (Isola et al., 2017; Mathieu et al., 2016), random noise is not used in the Pix2Pix architecture, since that noise vector does not make a huge difference in the generator's output. This is because there is a paired output image that the generator is trying to synthesize.

Given its successful application to image translation, in this paper, we propose combining the cGAN and polygon generation to solve multi-output regression problems. The proposed method is presented in detail in the next section.

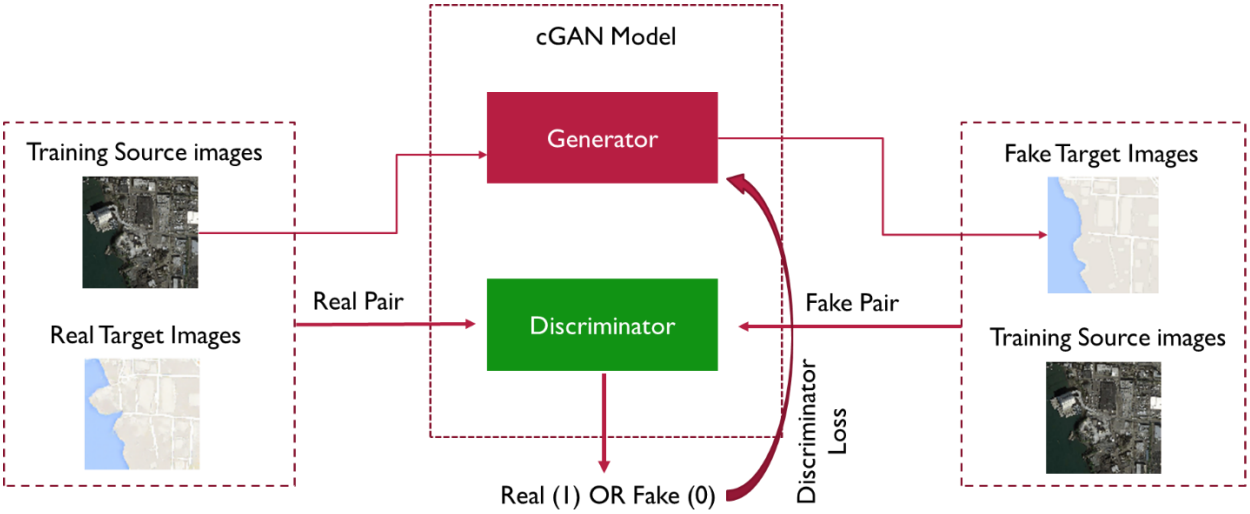


Figure 4.4 - A schematic diagram of the cGAN to translate satellite images into Google Map images

4.4 Proposed Method: Multi-Output Regression Using Polygon Generation and cGAN

The proposed PGcGAN method is comprised of two phases; training and testing, as shown in Figure 4.5. The main purpose of the training phase is to build a generator model based on the representative images of data variables (inputs) and KPIs (outputs), created using polygon generation. In the testing phase, the trained generator model can synthesize the images for the numeric data outputs (KPIs). The two phases are illustrated in the following subsections.

4.4.1 Training Phase: Building Generator Model

Our proposed multi-output regression method is targeting numeric data with several numerical outputs. It is a supervised learning method that is fed with a labeled training data to build a prediction model for the new unseen observations. As shown in Figure 4.5, training data with n numerical input variables and m numerical outputs is split into two separate tables, one includes all input variables $(X_1, X_2, \dots, X_n)$ and the other includes only the numerical outputs $(Y_1, Y_2, \dots, Y_m)$. The polygon generation method illustrated before is applied to both tables separately. Accordingly, the proposed method has the advantage of expressing all interrelationships between numerical variables in the form of polygon images (training source images). Besides, all interrelationships

between the numerical outputs are represented through another set of polygon images (training target images). In this phase, pairs of images are formed, where each pair includes a polygon image that represents the numerical variables of a certain observation, and the second image represents its corresponding outputs.

These pairs of images are fed as input to the Pix2Pix generator (cGAN) model. The generator tries to convert the set of polygon images representing the numerical variables (training source images) into the other set of polygon images representing the numerical outputs (training target images) as shown in Figure 4.6. The discriminator tries to figure out if the generated pair is fake or real and it keeps competing with the generator until the latter synthesizes sharp and realistic images that are so close to the training target images. The cGAN model tries to approximate the true distribution of the input variables and their relationships with the numerical outputs by extracting the correspondence between the two sets of polygon images. The output of the cGAN is the trained generator that is used later in the testing phase.

4.4.2 Testing Phase: Mapping Images into Numeric Outputs

As shown in Figure 4.6, given the testing numerical data, the polygon generation method is applied to the input variables to form a new set of polygon images (testing source images). The trained generator, which is the output of the training phase, is used to synthesize polygon images that represent the predicted numeric outputs. In order to map these synthesized images into numerical values, an image processing procedure is needed to get back the predicted numerical outputs. Figure 4.7 summarizes the image processing steps. First, the cMinMax algorithm proposed in (Chamzas et al., 2020) is used to find the corners of the polygon (a triangle in Figure 4.7 represents an example of three outputs). These corners are common in all generated testing images. The midpoints of polygon sides that represent the mean value of numerical outputs are determined through the corner points. The other points on each polygon side represent different values of each corresponding numerical output. The pixel value of each point on each polygon side is summed up with the pixel values of its surrounding pixels. The point with maximum summation represents the standardized numerical value of its corresponding output. By using the polygon generation technique, the mapping of polygon images into numerical values is performed as an easy step using

both the midpoints of polygon sides (representing the mean values of the numerical outputs) and the points that represent the standardized values of their corresponding outputs.

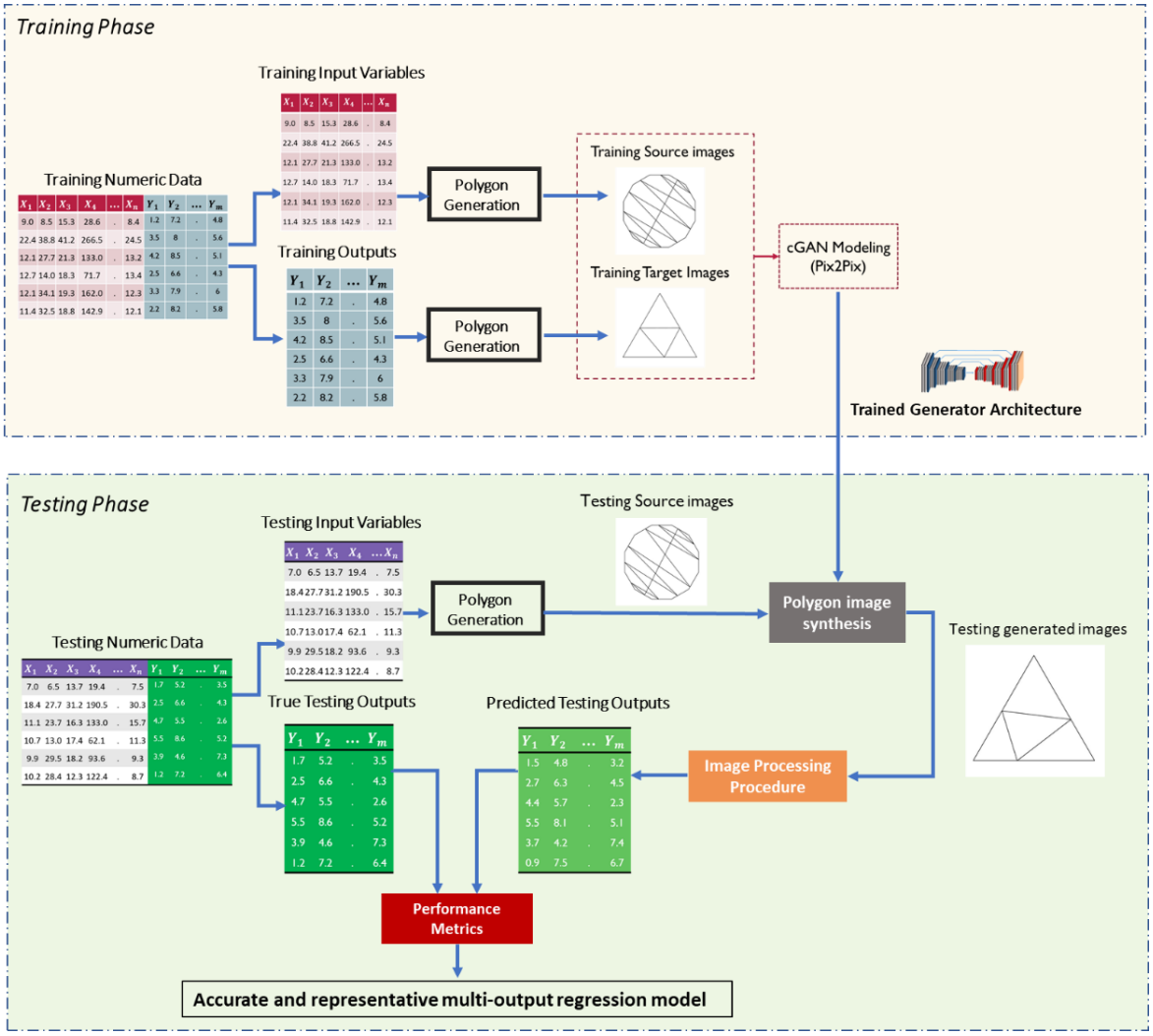


Figure 4.5 - A detailed schematic diagram of the proposed method (PGcGAN)

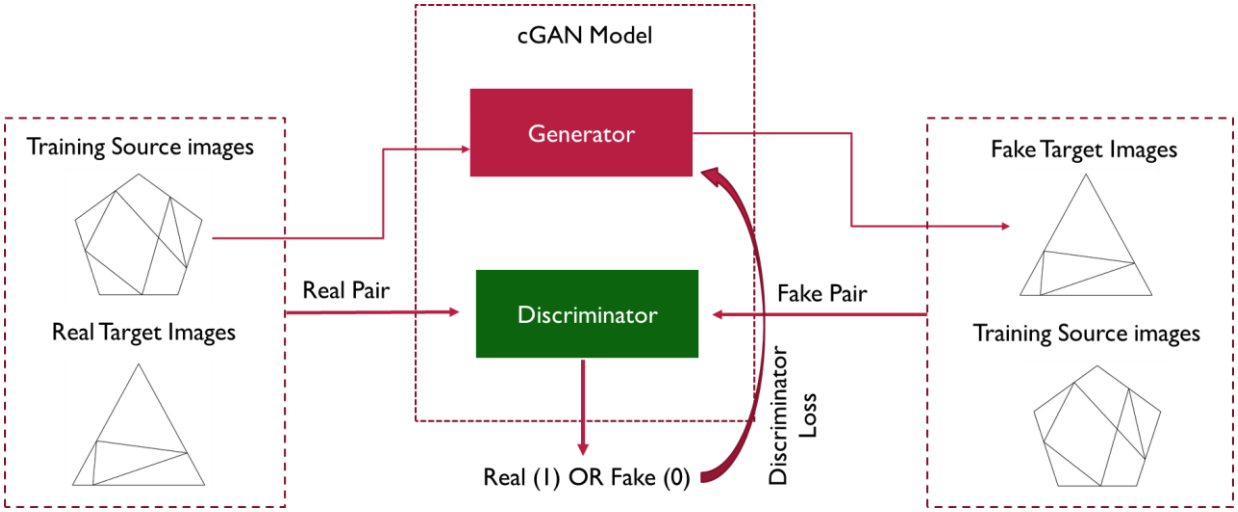


Figure 4.6 - Training of the cGAN model to convert polygon images representing input variables into another set of polygon images representing the numerical outputs

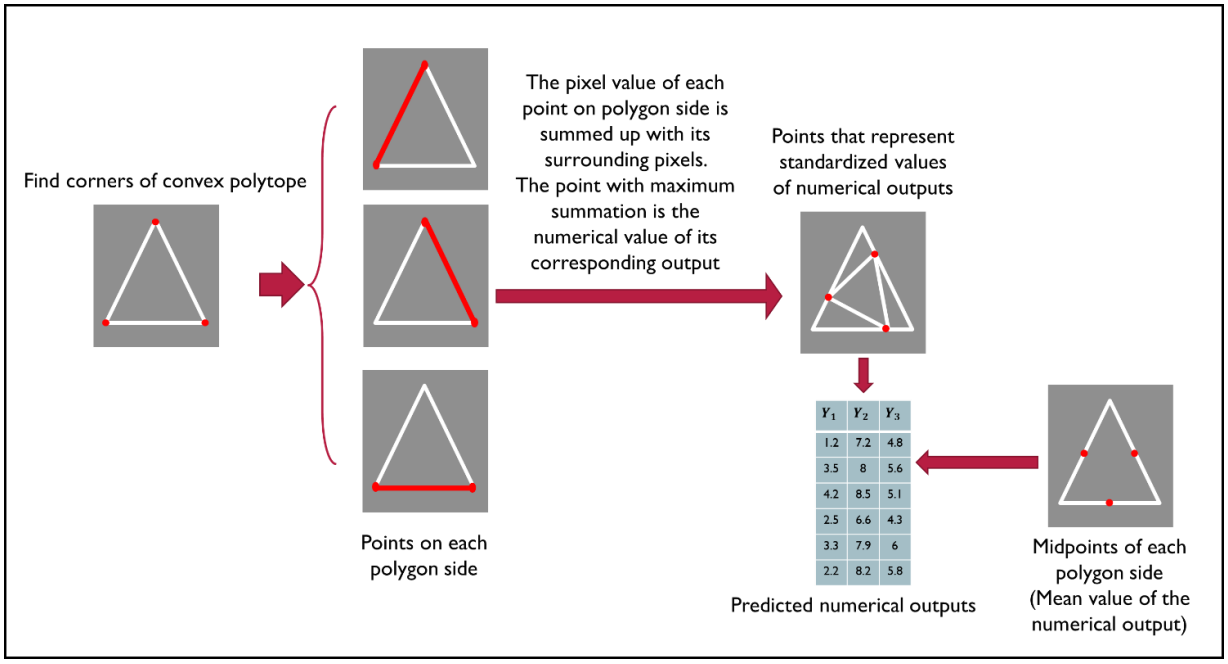


Figure 4.7 - A schematic diagram summarizes the image processing steps for mapping polygon images into numerical output values

4.5 Case Study: Black Liquor Recovery Boiler (BLRB) in Kraft Pulp & Paper Mills

The proposed regression method is validated based on real data with complex distribution collected from major equipment in the pulp and paper industry. This section provides detail on this equipment and the collected data, along with an explanation of the KPIs.

4.5.1 BLRB: Operation and KPIs

The BLRB is an important piece of equipment in Kraft Pulp & Paper mills. It is important to preserve the high-efficiency of the BLRB process operation (Ragab et al., 2018). The BLRB mainly produces steam through the combustion of the black liquor that comes from the pulping process. It also recovers several chemicals that are reused in the process. A schematic diagram of the BLRB is shown in Figure 4.8. One of the economic KPIs that reflects the efficiency of the BLRB is steam production divided by black liquor flow (SP/BLF). A high SP/BLF indicates high performance. The SP/BLF is affected by many factors, such as combustion air in the furnace, the superheater and the inlet water temperature, and the percentage of solids in the black liquor. Therefore, it is necessary to maximize this KPI and keep it as high as possible.

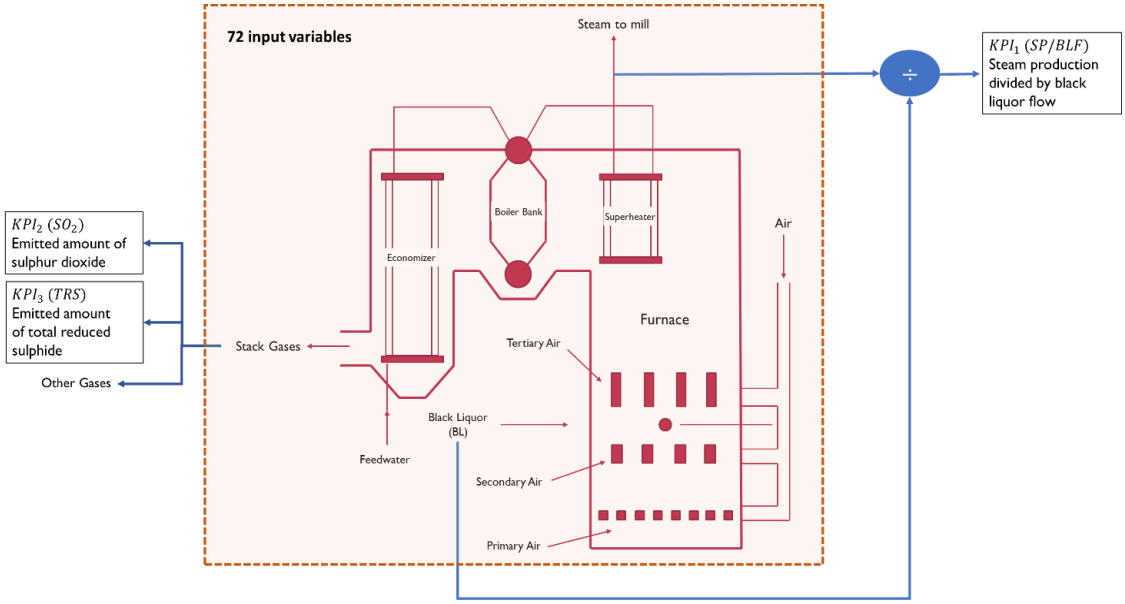


Figure 4.8 - A schematic diagram of the Black Liquor Recovery Boiler (BLRB) (Vakkilainen & others, 2005)

Moreover, the BLRB is one of the largest producers of harmful emissions in a pulp mill. This is one of the reasons why the pulp and paper industry suffers from several environmental issues (Vasara, 2001). Emissions from a BLRB into the air, such as sulphur dioxide (SO_2) and total reduced sulphide (TRS), depend on the process conditions, the type of equipment and the personnel who run the unit with high level process control optimization. The proper operation of the BLRB can help minimize these environmental impacts. Accordingly, two KPIs are included in the BLRB dataset to reflect the effects of the BLRB on the environment. They represent the emitted amounts of SO_2 and TRS.

Therefore, there is a clear need to better predict the three KPIs, whether economic or environmental, to maximize the business value and to reduce and alleviate any environmental consequences through proper monitoring of the BLRB process.

4.5.2 Experimental setup & Results

The BLRB dataset of 75951 observations was collected from the mill historian with a sampling time of 5 mins. The dataset comprises a total of 92 manipulated and measured variables; 72 variables were selected by the process expert to represent the behavior of the BLRB process. Examples of these variables are shown in Table 4.3. The expert cleaned and prepared the data by removing the outliers and other non-representative observations using the software EXPLORE (Amazouz, 2015).

Table 4.3 - Examples of manipulated, measured variables and KPIs for the BLRB

Manipulated and Measured variables	KPIs
▪ Vapor pressure atomization (kPa) ▪ Carbon Monoxide (CO) emissions (ppm) ▪ Pressure levels at different locations (kPa) ▪ Temperature at different locations (^{0}C) ▪ Primary, secondary, and tertiary air ratios (%) ▪ Total air flow (m^3/min) ▪ Differential Pressure (kPa) ▪ Solid Recovery (TM/j)	▪ KPI1 (SP/BLF): steam production divided by black liquor flow ▪ KPI2 (SO_2): emitted amount of sulphur dioxide (ppm) ▪ KPI3 (TRS): emitted amount of total reduced sulphide (%)

The performance of the proposed method is compared with other ML regressors; Multi-layer Perceptron (MLP), Decision Tree (DT), Random Forest. These ML regressors have been used extensively in the literature and in practice (Baturynska & Martinsen, 2020; Bustillo, Pimenov, et al., 2020; Bustillo, Reis, et al., 2020; Narayanasamy & Padmanabhan, 2012; Ragab, Yacout, et al., 2019). As mentioned previously, the cGANs are composed mainly of two blocks: generator and discriminator. To build both the generator and discriminator models, we used the architecture of the Pix2Pix model used in (Isola et al., 2017).

More specifically, the Pix2Pix generator is an encoder-decoder called the U-Net architecture introduced in (Ronneberger et al., 2015). As shown in Figure 4.9, the U-Net is composed of eight encoder and eight decoder blocks. Each encoder block has a convolutional layer, batch normalization layer and leaky rectified linear unit (ReLU) activation layer. Each decoder block has a transpose convolutional layer, batch normalization layer and ReLU activation layer. As shown in the figure, the middle layer is a bottleneck where all important information in an input image is compressed into vector spaces, representing a set of high-level features. These features are then decoded into another output image with the same resolution. Since it is easy for these networks to overfit to the training image pairs, the U-Net introduces skip connections from the encoder to decoder, allowing for the encoder information to be concatenated into the decoder with the same resolution before going into each convolutional block in the decoder, as shown in Figure 4.9.

The discriminator network in the Pix2Pix is the PatchGAN architecture introduced in (Demir & Unal, 2018). This architecture gives output in the form of a matrix of classification probabilities instead of a single output. It takes a patch of an image at a time and assigns it one value out of the entire classification matrix. Each entry in this matrix has a value between 0 and 1, where 0 refers to a fake patch and 1 refers to a real one, as shown in Figure 4.10. In this case study, the PatchGAN considers 70 by 70 patches in each image.

In this paper, the Anaconda package ("Anaconda Software Distribution," 2020) is used to coordinate different ML and DL installed packages such as TensorFlow (Martin Abadi et al., 2015), Sci-kit learn (Pedregosa et al., 2011) and Keras (Chollet & others, 2018) with Python 3.7. Implementation, training and testing the Pix2Pix model and the other baseline DL algorithms are

done using the computing infrastructure; Processor: Intel(R) Core(TM) i7-8750H CPU @2.2 GHz + GPU: NVIDIA GeForce GTX 1070 with Max-Q Design and RAM: 16 GB.

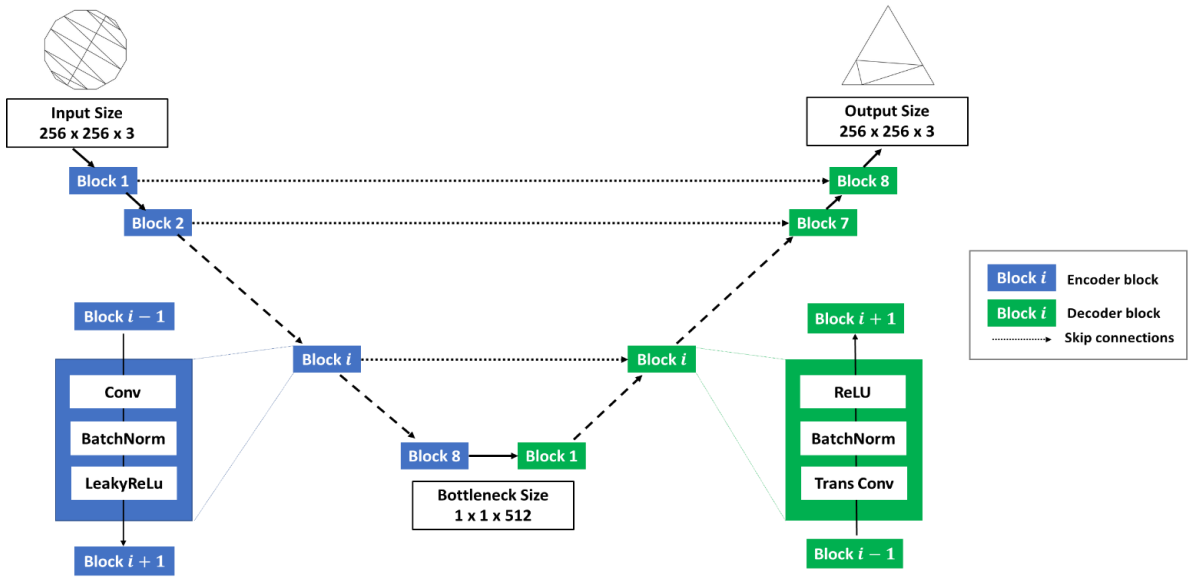


Figure 4.9 - The U-Net architecture: Generator of Pix2Pix model (Ronneberger et al., 2015)

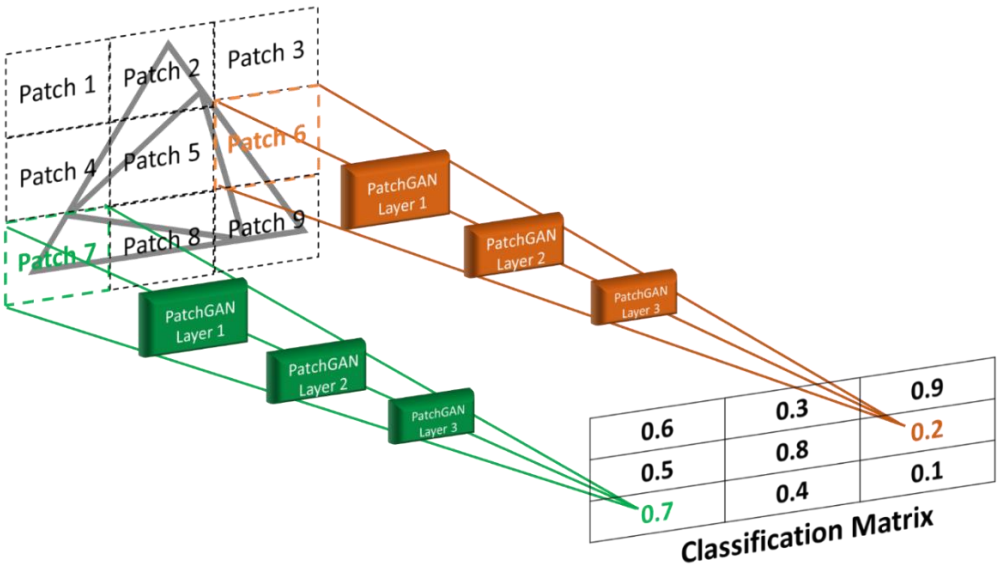


Figure 4.10 - PatchGAN: The discriminator of Pix2Pix model (Demir & Unal, 2018)

4.6 Results, Discussion & Future Work

To validate the proposed method, the results obtained are compared with the results achieved from other ML regressors; the MLP, DT and Random Forest. Moreover, the results of the PGcGAN are compared to the cGAN without using the polygon images to demonstrate the effectiveness of the polygon generation technique. To achieve the best results from these regression algorithms, their hyperparameters are optimized using a grid search and 5-fold cross validation. The ranges of optimized parameters for the BLRB case study are shown in Table 4.4. For the reproducibility of the results, a random seed is fixed to ensure the consistency of trained models. The R-squared values (R^2) and root mean square error ($RMSE$) of each KPI are used as the performance metrics to test each regressor (Bustillo, Reis, et al., 2020; Kasuya, 2019).

Table 4.4 - Range of hyperparameters of each regressor

Algorithm	Hyperparameters
Proposed Method	# filters in conv layer = [4,64], filter size = (2,3) batch size = [50,200] # epochs = [30,150],
MLP	# neurons in hidden layer = [10,40] Maximum number of iterations = [1000,5000]
DT	Function to measure quality of split = {“Mean square error”,”poisson deviance criterion”} Split strategy = {“best”,”random”} maximum depth = [5,12] # features to consider when looking for the best split ={”#features”,”sqrt(#features)”,”log2(#features)”}
RF	Function to measure quality of split = {“Mean square error”,”poisson deviance criterion”} # trees (estimators) = [500,2000], maximum depth = [5,12] # features to consider when looking for the best split ={”#features”,”sqrt(#features)”,”log2(#features)”}

Moreover, a penalty function is calculated for each regression model as a validation criterion. This function takes into consideration the underestimation and overestimation of each

KPI. The penalty function $L_{acc}^{j}(t_k)$ of KPI_j , $j = 1,2,3$ at instance t_k is defined as shown in Eq. (3).

$$L_{acc}^{j}(t_k) = \begin{cases} \alpha_j u(j)\left(KPI_j(t_k) - \widehat{KPI_j}(t_k)\right) & , \widehat{KPI_j}(t_k)u(j) < KPI_j(t_k)u(j) \\ 0 & , \widehat{KPI_j}(t_k) = KPI_j(t_k) \\ \beta_j u(j)\left(\widehat{KPI_j}(t_k) - KPI_j(t_k)\right) & , \widehat{KPI_j}(t_k)u(j) > KPI_j(t_k)u(j) \end{cases} \quad (3)$$

where, α_j and β_j are the underestimation and overestimation parameters for each KPI_j respectively and $\widehat{KPI_j}(t_k)$ and $KPI_j(t_k)$ are the predicted and true values of KPI_j at instance t_k respectively. The term $u(j)$ has a value of 1 or -1 depending on the predicted KPI_j. $u(1) = 1$ (the overestimation of KPI_1 is penalized more than the underestimation), while $u(2) = u(3) = -1$ (the underestimation of KPI_2 and KPI_3 is penalized more than the overestimation) These parameters were assigned according to the economic and environmental importance of each KPI as confirmed by the process expert. Accrodingly, in this paper, the values of α_j and β_j are assigned as shown in Table 4.5.

Table 4.5 - Underestimation (α_j) and overestimation (β_j) parameters for each KPI_j in the penalty function as assigned by the process expert

	$j = 1$	$j = 2$	$j = 3$
α_j	0.1	0.1	0.1
β_j	0.15	0.2	0.5

The average penalty score for each KPI is calculated as shown in Eq. (4).

$$L_{AP}^{j} = \frac{1}{N}\sum_{k=1}^{N} L_{acc}^{j}(t_k), \quad (4)$$

where N is the total number of time steps. The total average penalty score for each regression model is calculated as shown in Eq. (5).

$$L_{model} = \frac{1}{3}\sum_{j=1}^{3} L_{AP}^{j}, \quad (5)$$

4.6.1 Results

As illustrated in section 4.5, a total of three KPIs are used for this multi-output regression case study, the steam production divided by black liquor flow (SP/BLF), the emitted amount of SO_2 and the emitted amount of TRS. Based on the polygon generation approach, each data observation has 36 different Hamiltonian cycle connections (36 different polygon images) that completely represent all interrelationships between the input variables, while one Hamiltonian cycle (one polygon image) represents the three numeric outputs. The size of the input images is set to 256 x 256.

As shown in Tables Table 4.6 and Table 4.7, the proposed method (cGAN) has achieved the highest R-squared value and lowest $RMSE$ in the three KPIs and the lowest total average penalty score compared to the other ML prediction models. The numbers in bold indicate the best results obtained. It is observed that the proposed PGcGAN outperformed the cGAN without using the polygon generation technique. This demonstrates the importance of this technique as a powerful data representation to maximize the global value of the industrial data for achieving better model performance. It can be observed that there is a significant improvement in the prediction accuracy of the KPI_3 (TRS) compared to the other methods. This confirms that the proposed method represents a promising solution in the direction of maximizing the steam production and reducing the environmental impacts that are caused by such harmful TRS emissions. These results are validated and confirmed by the process expert and are shown to be very beneficial for the mill operator, as he or she can monitor the process more efficiently and select the best action to take at the right time.

Figure 4.11,Figure 4.12 and Figure 4.13 visualize the performance of the proposed method in comparison with the different methods. They show the predicted values of the three KPIs over time, where each figure compares the proposed method with the true values and a baseline regressor.

Table 4.6 - R-squared and Root Mean Square Error (RMSE) values of each algorithm in BLRB dataset

Algorithm	$KPI_1\ (SP/BLF)$		$KPI_2\ (SO_2)$		$KPI_3\ (TRS)$	
	R^2	$RMSE$	R^2	$RMSE$	R^2	$RMSE$
Proposed Method (PGcGAN)	**0.88**	**0.041**	**0.64**	**9.48**	**0.90**	**3.65**
cGAN	0.86	0.045	0.48	11.55	0.48	8.21
MLP	0.85	0.046	0.46	11.63	0.56	7.83
RF	0.74	0.061	0.54	10.74	0.61	7.38
DT	0.81	0.052	0.25	13.78	0.31	9.79

Table 4.7 - Total average penalty scores for each algorithm in BLRB dataset

Algorithm	PGcGAN	cGAN	MLP	DT	RF
Total average penalty score	**0.0304**	0.0347	0.0379	0.0382	0.0315

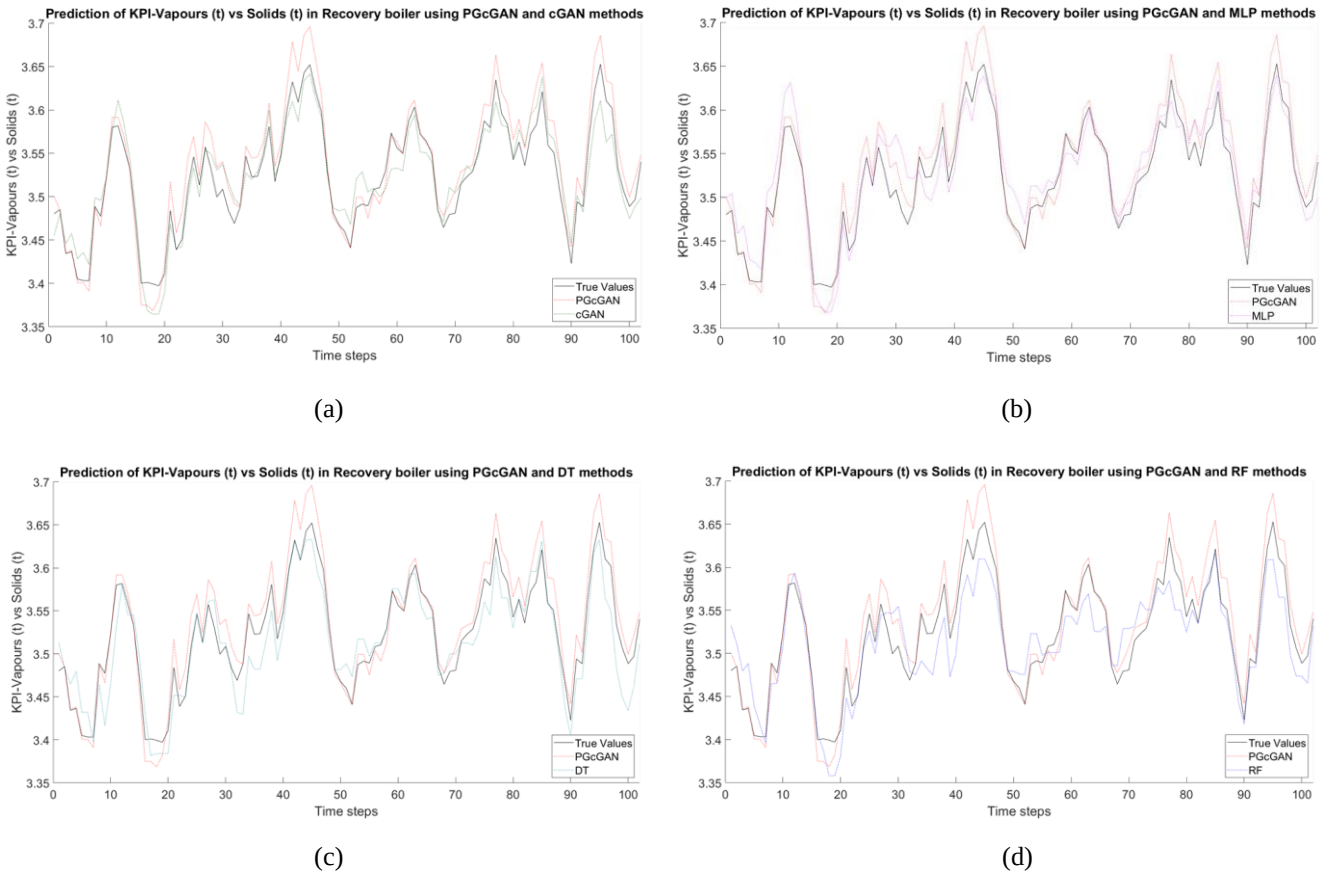


Figure 4.11 - Prediction of KPI_1 (steam production divided by black liquor flow (SP/BLF)) using the proposed method (PGcGAN) and other ML regression models (a) cGAN (b) MLP (c) DT (d) RF

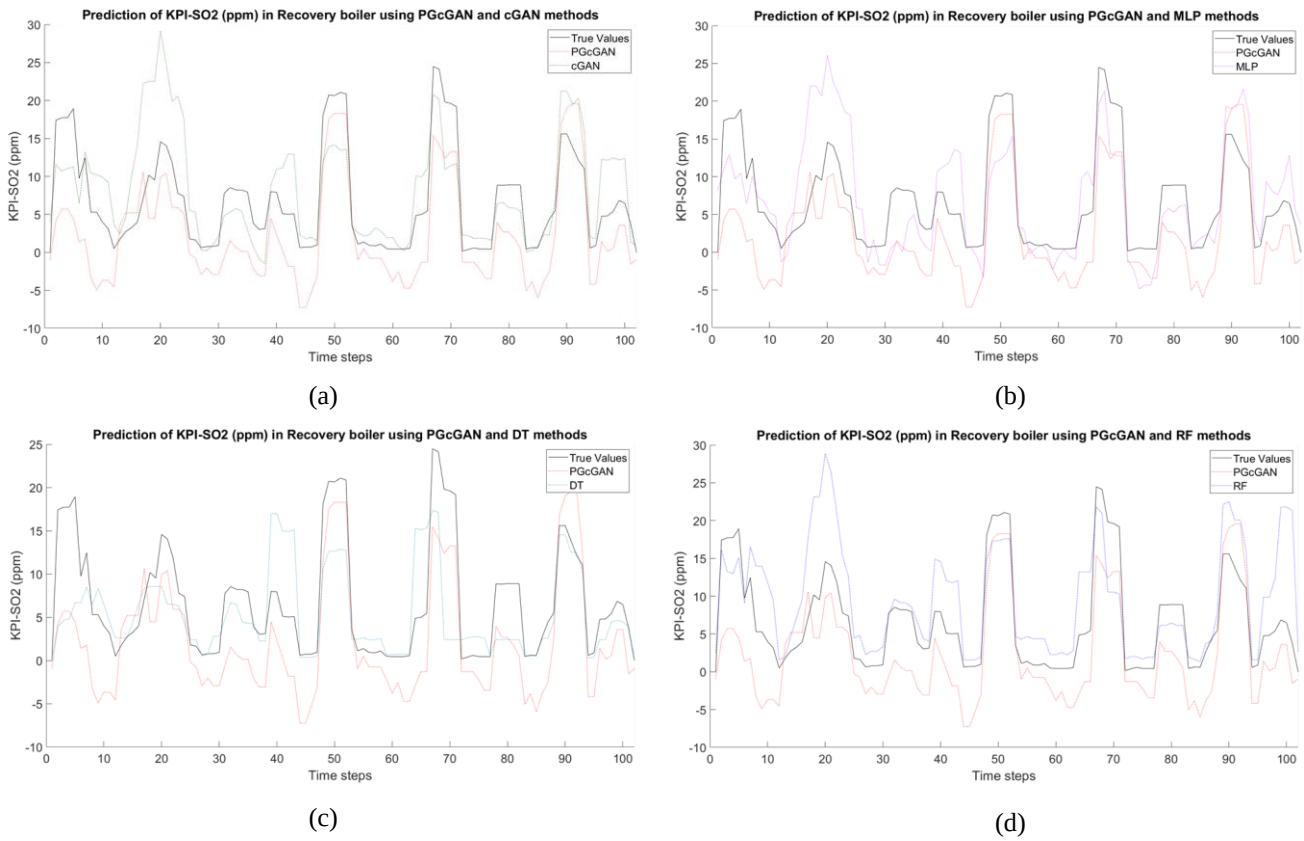


Figure 4.12 - Prediction of KPI_2 (emitted sulphur dioxide (SO_2)) using the proposed method (PGcGAN) and other ML regression models (a) cGAN (b) MLP (c) DT (d) RF

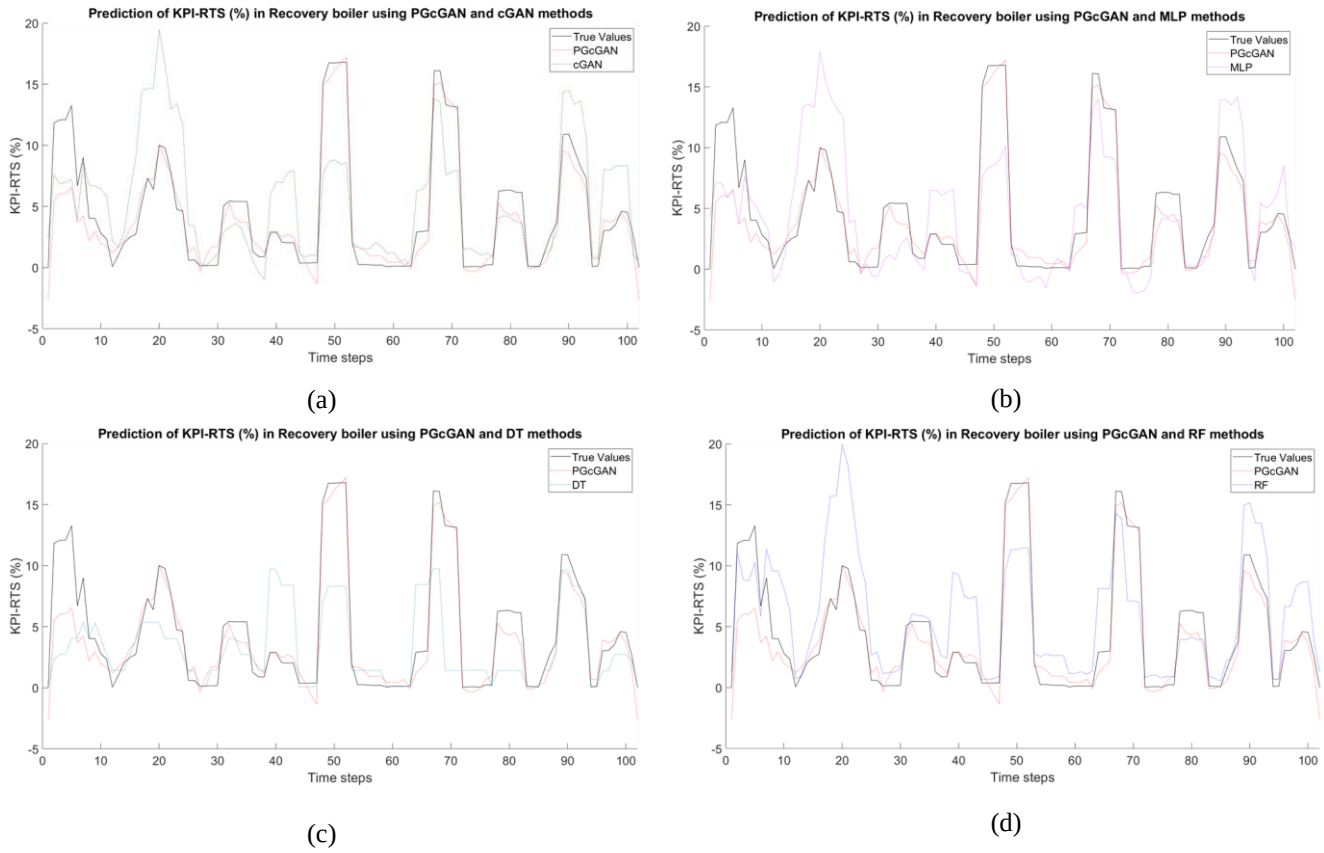


Figure 4.13 - Prediction of KPI_3 (total reduced sulphide (TRS)), the proposed method (PGcGAN) and other ML regression models (a) cGAN (b) MLP (c) DT (d) RF

4.6.2 Discussion and Future Work

To recap, the implementation of the proposed PGcGAN method can be easily applied in an industrial context. As shown in Figure 4.14, the historical data is acquired from the plant monitored, including input variables and KPIs. The figure summarizes the possible implementation of the proposed method as a generic and flexible method applied to predict KPIs in various applications in the industry other than BLRB. In an analogy with the training and testing phases presented in Section 4.4 of this paper, the historical data is converted into polygons (source and target images) and then used to train a generative model (offline). The new observations are collected and converted into other polygons (online). These polygon images are exploited by the trained generative model and then translated into other images that are processed to predict the KPIs. In addition to its high prediction performance, one of the most important remarks that the authors

would like to share with readers is that the online phase of the proposed method is not computationally demanding. From an industrial perspective, in which an accurate and fast prediction is a desired characteristic, this is a potential benefit.

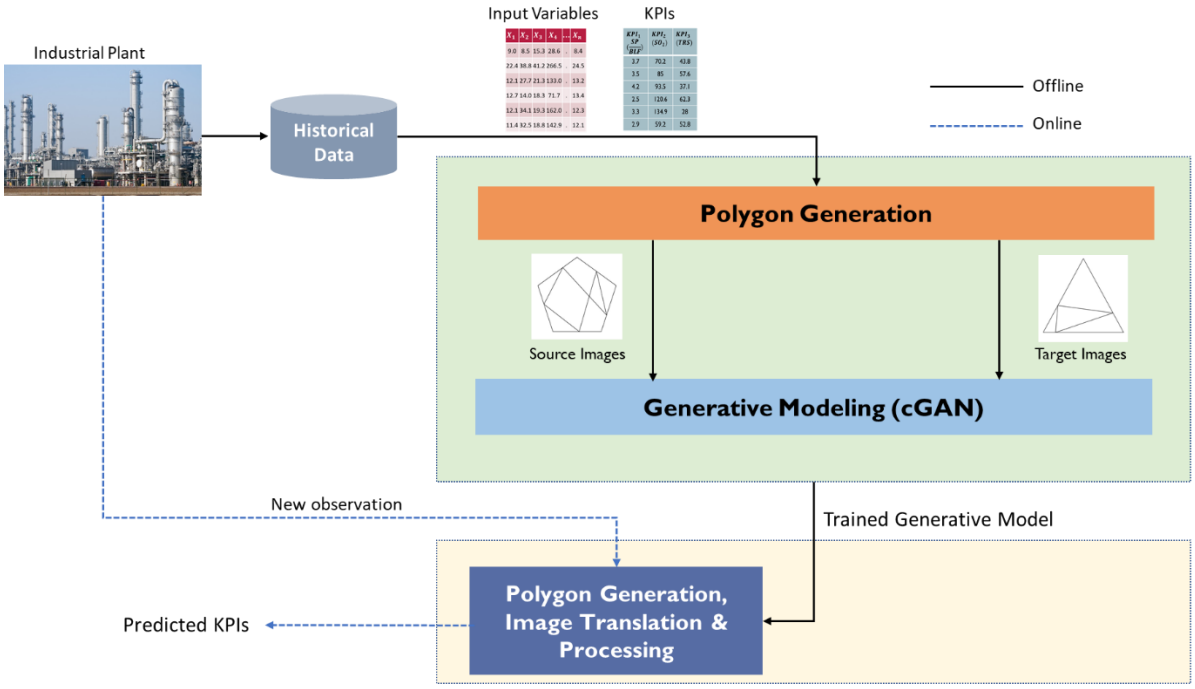


Figure 4.14 - A schematic diagram summarizing the proposed method (PGcGAN) for predicting KPIs in industrial plants

Here, we have some important remarks to share with researchers and practitioners. PGcGAN is a result of combining both deep generative modeling (cGAN) and a powerful data representation (polygon generation). An important distinction of the proposed method is that it represents all relationships between input variables through a set of polygon images. It does the same with the numeric outputs through another set of polygon images. This is very useful in helping obtain the true distribution of the complex datasets that are found in most industrial applications. This is attributed to the fact that the polygon generation method can represent all interrelationships between the input variables or the numeric outputs using different Hamiltonian cycles. Accordingly, this increases the quality of industrial data that significantly impacts the model

performance. Furthermore, polygon generation has the advantage of data augmentation by representing each observation through multiple images with different orders of connection.

We take advantage of the breakthrough of cGANs as an accurate, deep generative modeling method in computer vision problems to generate sharp images that represent the numeric outputs well. The relationships between data inputs and numeric outputs are represented through one of the powerful cGAN models (the Pix2Pix) by translating two different sets of polygon images. In this work, it is worth noting that image processing procedures for mapping the output images into numerical values is neither computationally exhausting nor time consuming. It is worth mentioning that no manual feature engineering has been done in this proposed method in which the feature engineering is embedded inside the cGAN architecture as an "end-to-end learning" procedure.

The proposed method opens the door to solve many regression problems in various fields, not only in the process industry. Big data, IoT, image processing and other fields are rich with numerical and imagery data that can be exploited through the PGcGAN with the aid of a suitable computational infrastructure to build robust regression models (Long et al., 2019; Jin Wang et al., 2020; D. Zhang et al., 2017; J. Zhang et al., 2020). Furthermore, using polygon generation for both classification and regression problems can be a building block for a data fusion tool that can merge both imagery and numerical data. Testing our proposed method on other complex datasets with different data distribution in the process industries should be investigated. Beside the BLRB as a complex equipment in the process industries, we are targeting other types of equipment such as heat exchangers that are commonly found in the pulp & paper industry and oil & gas refineries. We will also consider the effect of the polygon image resolution and its accuracy, which may have a significant impact on the performance of our regression method. Therefore, addressing the accuracy of the generated polygons using a model trustworthiness metric is one of our future research directions. Tackling the interpretability of such deep networks to extract meaningful rules for a process expert is one of our future directions.

4.7 Conclusion

This paper proposes a novel multi-output regression method that combines two techniques: polygon generation and conditional generative adversarial networks (cGAN). The numeric observations are converted into a set of polygon images using polygon generation. An image-to-

image translation process has been done using one of the powerful cGAN models for converting polygon images representing the input variables into other set of images, representing outputs. The proposed method takes advantage of the breakthrough of deep learning (DL) in generative modeling to capture the true data distribution, which is difficult to determine in complex process industries. The method is validated using a real industrial dataset collected from a black liquor recovery boiler (BLRB) in a Kraft pulp and paper mill located in Canada. The results obtained show that the proposed method outperforms other machine learning (ML) regressors when predicting three important key performance indicators (KPIs); one economical and two environmental. Accordingly, the proposed method has proven to be a promising tool for multi-output regression in many industrial applications. Furthermore, the visualization of the polygon generation technique can play a role in the interpretation of such DL networks. Moreover, using the polygon generation technique in different supervised learning problems will open the door for a data fusion methodology that could combine different sources of heterogeneous data. These are two of our future research directions. More datasets from different industries for testing the proposed method will be considered as one of our future research directions.

4.8 References

Abadi, Mart\'\in, Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., & others. (2016). Tensorflow: A system for large-scale machine learning. *12th $\{$USENIX$\}$ Symposium on Operating Systems Design and Implementation ($\{$OSDI$\}$ 16)*, 265–283.

Abadi, Martin, Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., S. Corrado, G., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Yangqing, J., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/

Afrasiabi, S., Afrasiabi, M., Parang, B., Mohammadi, M., Arefi, M. M., & Rastegar, M. (2019). Wind Turbine Fault Diagnosis with Generative-Temporal Convolutional Neural Network. *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, 1–5.

Aggarwal, K., Kirchmeyer, M., Yadav, P., Keerthi, S. S., & Gallinari, P. (2019). *Benchmarking Regression Methods: A comparison with CGAN*. *May*. http://arxiv.org/abs/1905.12868

Alam, F., Mehmood, R., Katib, I., Albogami, N. N., & Albeshri, A. (2017). Data fusion and IoT for smart ubiquitous environments: A survey. *IEEE Access*, *5*, 9533–9554.

Alpaydin, E. (2010). *Introduction to machine learning, 2nd edn. Adaptive computation and machine learning*. The MIT Press (February 2010).

Amazouz, M. (2015). *Improving Process Operation Using the Power of Advanced Data Analysis*. https://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/canmetenergy/files/pubs/EXPLORE-brochure_ EN.pdf

Anaconda Software Distribution. (2020). In *Anaconda Documentation*. Anaconda Inc. https://docs.anaconda.com/

Andersson, E., & Thollander, P. (2019). Key performance indicators for energy management in the Swedish pulp and paper industry. *Energy Strategy Reviews*, *24*(December 2018), 229–235. https://doi.org/10.1016/j.esr.2019.03.004

Andrew Ng Launches A Campaign For Data-Centric AI. (n.d.). https://www.forbes.com/sites/gilpress/2021/06/16/andrew-ng-launches-a-campaign-for-data-centric-ai/?sh=5dea92f374f5

Ardsomang, T., Hines, J. W., & Upadhyaya, B. R. (2013). Heat exchanger fouling and estimation of remaining useful life. *Annual Conference of the PHM Society*, *5*(1).

Arendt, J. S., & Lorenzo, D. K. (2010). *Evaluating process safety in the chemical industry: A user's guide to quantitative risk analysis* (Vol. 3). John Wiley \& Sons.

Athar, M., Mohd Shariff, A., Buang, A., Shuaib Shaikh, M., & Ishaq Khan, M. (2019). Review of Process Industry Accidents Analysis towards Safety System Improvement and Sustainable Process Design. *Chemical Engineering and Technology*, *42*(3), 524–538. https://doi.org/10.1002/ceat.201800215

Ayubi Rad, M. A., & Yazdanpanah, M. J. (2015). Designing supervised local neural network classifiers based on EM clustering for fault diagnosis of Tennessee Eastman process.

Chemometrics and Intelligent Laboratory Systems, *146*, 149–157. https://doi.org/10.1016/j.chemolab.2015.05.013

Bache, K., & Lichman, M. (2013). *UCI machine learning repository*.

Bai, Y., Xie, J., Wang, D., Zhang, W., & Li, C. (2021). A manufacturing quality prediction model based on AdaBoost-LSTM with rough knowledge. *Computers and Industrial Engineering*, *155*(January), 107227. https://doi.org/10.1016/j.cie.2021.107227

Bajpai, P. (2018). Brief Description of the Pulp and Papermaking Process. In *Biotechnology for pulp and paper processing* (pp. 9–26). Springer.

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., & others. (2012). NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Research*, *41*(D1), D991--D995.

Bashkirova, D., Usman, B., & Saenko, K. (2018). *Unsupervised Video-to-Video Translation*. *Nips*. http://arxiv.org/abs/1806.03698

Bathelt, A., Ricker, N. L., & Jelali, M. (2015). Revision of the tennessee eastman process model. *IFAC-PapersOnLine*, *48*(8), 309–314.

Baturynska, I., & Martinsen, K. (2020). Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms. *Journal of Intelligent Manufacturing*, *32*(1), 179–200. https://doi.org/10.1007/s10845-020-01567-0

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Biermann, C. J. (1996). *Handbook of pulping and papermaking*. Elsevier.

Borovykh, A., Bohte, S., & Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10614 LNCS*, 729–730.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley \& Sons.

Brown, R. G., & Meyer, R. F. (1961). The fundamental theorem of exponential smoothing. *Operations Research*, *9*(5), 673–685.

Bustillo, A., Pimenov, D. Y., Mia, M., & Kapłonek, W. (2020). Machine-learning for automatic prediction of flatness deviation considering the wear of the face mill teeth. *Journal of Intelligent Manufacturing*, *mm*. https://doi.org/10.1007/s10845-020-01645-3

Bustillo, A., Reis, R., Machado, A. R., & Pimenov, D. Y. (2020). Improving the accuracy of machine-learning models with data from machine test repetitions. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01661-3

Camacho, J., Magán-Carrión, R., García-Teodoro, P., & Treinen, J. J. (2016). Networkmetrics: multivariate big data analysis in the context of the internet. *Journal of Chemometrics*, *30*(9), 488–505. https://doi.org/10.1002/cem.2806

Cao, S., Wen, L., Li, X., & Gao, L. (2018). Application of Generative Adversarial Networks for Intelligent Fault Diagnosis. *IEEE International Conference on Automation Science and Engineering*, *2018-Augus*, 711–715. https://doi.org/10.1109/COASE.2018.8560528

Castanedo, F. (2013). A review of data fusion techniques. In *The Scientific World Journal* (Vol. 2013). Hindawi Publishing Corporation. https://doi.org/10.1155/2013/704504

Chamzas, D., Chamzas, C., & Moustakas, K. (2020). cMinMax: A fast algorithm to find the corners of an N-dimensional convex polytope. *ArXiv:2011.14035v2*.

Chen, X., Zhang, B., & Gao, D. (2020). Bearing fault diagnosis base on multi-scale CNN and LSTM model. *Journal of Intelligent Manufacturing*, *December 2019*. https://doi.org/10.1007/s10845-020-01600-2

Chen, Z., Zeng, X., Li, W., & Liao, G. (2016). Machine fault classification using deep belief network. *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, *2016-July*(51475170). https://doi.org/10.1109/I2MTC.2016.7520473

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio,

Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Choi, H., Lee, H., & Kim, H. (2009). Fast detection and visualization of network attacks on parallel coordinates. *Computers & Security*, *28*(5), 276–288.

Chollet, F., & others. (2015). *Keras*. GitHub.

Chollet, F., & others. (2018). Keras: The python deep learning library. *Astrophysics Source Code Library*.

Cocchi, M. (2019). Introduction: Ways and Means to Deal With Data From Multiple Sources. *Data Handling in Science and Technology*, *31*, 1–26. https://doi.org/10.1016/B978-0-444-63984-4.00001-6

D'Angelo, M. F. S. V, Palhares, R. M., Camargos Filho, M. C. O., Maia, R. D., Mendes, J. B., & Ekel, P. Y. (2016). A new fault classification approach applied to Tennessee Eastman benchmark process. *Applied Soft Computing*, *49*, 676–686. https://doi.org/https://doi.org/10.1016/j.asoc.2016.08.040

Data-Centric AI Competition. (n.d.). https://https-deeplearning-ai.github.io/data-centric-comp/

Demir, U., & Unal, G. (2018). Patch-based image inpainting with generative adversarial networks. *ArXiv:1803.07422v1*.

Dong, D., Li, X.-Y., & Sun, F.-Q. (2017). Life prediction of jet engines based on LSTM-recurrent neural networks. *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 1–6.

Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, *17*(3), 245–255.

Dunia, R., Edgar, T. F., & Nixon, M. (2013). Process monitoring using principal components in parallel coordinates. *AIChE Journal*, *59*(2), 445–456.

Durall, R., Chatzimichailidis, A., Labus, P., & Keuper, J. (2020). Combating Mode Collapse in

GAN training: An Empirical Analysis using Hessian Eigenvalues. *ArXiv Preprint ArXiv:2012.09673*.

Duvall, P. M., & Riggs, J. B. (2000). On-line optimization of the Tennessee Eastman challenge problem. *Journal of Process Control*, *10*(1), 19–33. https://doi.org/https://doi.org/10.1016/S0959-1524(99)00041-4

Elhefnawy, M., Ragab, A., & Ouali, M.-S. (2021a). Fault classification in the process industry using polygon generation and deep learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-021-01742-x

Elhefnawy, M., Ragab, A., & Ouali, M. S. (2021b). Fault classification in the process industry using polygon generation and deep learning. *Journal of Intelligent Manufacturing*, *0123456789*. https://doi.org/10.1007/s10845-021-01742-x

Elizabeth Bush Nathan Gillett, E. W. J. F., & others. (2019). *Canada's Changing Climate Report*. https://changingclimate.ca/CCCR2019/

Environment challenges | Climate Action. (n.d.). https://ec.europa.eu/clima/policies/adaptation/how/challenges_en#tab-0-1

Eren, L., Ince, T., & Kiranyaz, S. (2019). A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *Journal of Signal Processing Systems*, *91*(2), 179–189.

Essien, A., & Giannetti, C. (2020). A Deep Learning Model for Smart Manufacturing Using Convolutional LSTM Neural Network Autoencoders. *IEEE Transactions on Industrial Informatics*, *16*(9), 6069–6078. https://doi.org/10.1109/TII.2020.2967556

FP, L. (1996). *Loss Prevention in the Process Industries: Hazard Identification, Assessment and Control*.

Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83–85.

Fu, K., Peng, J., Zhang, H., Wang, X., & Jiang, F. (2020). Image super-resolution based on generative adversarial networks: A brief review. *Computers, Materials and Continua*, *64*(3), 1977–1997. https://doi.org/10.32604/cmc.2020.09882

Fuentes, R., Fuster, B., & Lillo-Bañuls, A. (2016). A three-stage DEA model to evaluate learning-teaching technical efficiency: Key performance indicators and contextual variables. *Expert Systems with Applications*, *48*, 89–99. https://doi.org/10.1016/j.eswa.2015.11.022

Gamboa, J. C. B. (2017). *Deep Learning for Time-Series Analysis*.

Gärtler, M., Khaydarov, V., Klöpper, B., & Urbas, L. (2021). The Machine Learning Life Cycle in Chemical Operations – Status and Open Challenges. *Chemie Ingenieur Technik*, *12*, 1–19. https://doi.org/10.1002/cite.202100134

Ge, Z. (2017). Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemometrics and Intelligent Laboratory Systems*, *171*(September), 16–25. https://doi.org/10.1016/j.chemolab.2017.09.021

Gecgel, O., Ekwaro-Osire, S., Dias, J. P., Serwadda, A., Alemayehu, F. M., & Nispel, A. (2019). Gearbox Fault Diagnostics Using Deep Learning with Simulated Data. *Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management*, 1–8.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471.

Gheisari, M., Wang, G., & Bhuiyan, M. Z. A. (2017). A survey on deep learning in big data. *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, *2*, 173–180.

Golshan, M., boozarjomehry, R. B., & Pishvaie, M. R. (2005). A new approach to real time optimization of the Tennessee Eastman challenge problem. *Chemical Engineering Journal*, *112*(1), 33–44. https://doi.org/https://doi.org/10.1016/j.cej.2005.06.005

Gomez-Cabrero, D., Abugessaisa, I., Maier, D., Teschendorff, A., Merkenschlager, M., Gisel, A., Ballestar, E., Bongcam-Rudloff, E., Conesa, A., & Tegnér, J. (2014). *Data integration in the era of omics: current and future challenges*. BioMed Central.

Goodfellow, I. (2016). NIPS 2016 tutorial: Generative adversarial networks. *ArXiv*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. *Commun. ACM*, *63*(11), 139–144.

https://doi.org/10.1145/3422622

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative Adversarial Networks. *Commun. ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Gunerkar, R. S., Jalan, A. K., & Belgamwar, S. U. (2019). Fault diagnosis of rolling element bearing based on artificial neural network. *Journal of Mechanical Science and Technology*, *33*(2), 505–511. https://doi.org/10.1007/s12206-019-0103-x

Han, Z., Zhao, J., Leung, H., Ma, K. F., & Wang, W. (2021). A Review of Deep Learning Models for Time Series Prediction. *IEEE Sensors Journal*, *21*(6), 7833–7848. https://doi.org/10.1109/JSEN.2019.2923982

Hasan, M. J., Sohaib, M., & Kim, J.-M. (2019). 1D CNN-Based Transfer Learning Model for Bearing Fault Diagnosis Under Variable Working Conditions. In S. Omar, W. S. Haji Suhaili, & S. Phon-Amnuaisuk (Eds.), *Computational Intelligence in Information Systems* (pp. 13–23). Springer International Publishing.

Hauser, H., Ledermann, F., & Doleisch, H. (2002). Angular brushing of extended parallel coordinates. *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, 127–130.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

He, Y., Cowell, L., Diehl, A. D., Mobley, H. L., Peters, B., Ruttenberg, A., Scheuermann, R. H., Brinkman, R. R., Courtot, M., Mungall, C., & others. (2009). VO: vaccine ontology. *The 1st International Conference on Biomedical Ontology (ICBO 2009) Nature Precedings*, *2009*.

Hendrickx, D. M., Aerts, H. J. W. L., Caiment, F., Clark, D., Ebbels, T. M. D., Evelo, C. T., Gmuender, H., Hebels, D. G. A. J., Herwig, R., Hescheler, J., & others. (2014). diXa: a data infrastructure for chemical safety assessment. *Bioinformatics*, *31*(9), 1505–1507.

Heo, S., & Lee, J. H. (2018). Fault detection and classification using artificial neural networks. *IFAC-PapersOnLine*, *51*(18), 470–475.

Himeur, Y., Alsalemi, A., Al-Kababji, A., Bensaali, F., & Amira, A. (2020). Data fusion strategies

for energy efficiency in buildings: Overview, challenges and novel orientations. *Information Fusion*, *64*(June), 99–120. https://doi.org/10.1016/j.inffus.2020.07.003

Hoermann, S., Bach, M., & Dietmayer, K. (2018). Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2056–2063.

Hsu, C. Y., & Liu, W. C. (2020). Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, *0123456789*. https://doi.org/10.1007/s10845-020-01591-0

Huang, J. T., Li, J., & Gong, Y. (2015). An analysis of convolutional neural networks for speech recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2015-Augus*, 4989–4993. https://doi.org/10.1109/ICASSP.2015.7178920

Hurley, C. B., & Oldford, R. W. (2010). Pairwise display of high-dimensional information via eulerian tours and hamiltonian decompositions. *Journal of Computational and Graphical Statistics*, *19*(4), 861–886.

Ian Goodfellow Yoshua Bengio, A. C. (2017). The Deep Learning Book. *MIT Press*, *521*(7553), 785. https://doi.org/10.1016/B978-0-12-391420-0.09987-X

Inselberg, A. (2009). *Parallel coordinates*. Springer.

Inselberg, A., & Dimsdale, B. (1990). Parallel coordinates: a tool for visualizing multi-dimensional geometry. *Proceedings of the 1st Conference on Visualization'90*, 361–378.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 5967–5976. https://doi.org/10.1109/CVPR.2017.632

Jain, A., Smarra, F., Behl, M., & Mangharam, R. (2018). Data-driven model predictive control with regression trees-an application to building energy management. *ACM Transactions on Cyber-Physical Systems*, *2*(1), 1–21. https://doi.org/10.1145/3127023

Jain, S., Seth, G., Paruthi, A., Soni, U., & Kumar, G. (2020). Synthetic data augmentation for

surface defect detection and classification using deep learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01710-x

Jebara, T. (2012). *Machine learning: discriminative and generative* (Vol. 755). Springer Science & Business Media.

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221–231.

Jing, C., & Hou, J. (2015). SVM and PCA based fault classification approaches for complicated industrial process. *Neurocomputing*, *167*, 636–642. https://doi.org/https://doi.org/10.1016/j.neucom.2015.03.082

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9906 LNCS*, 694–711. https://doi.org/10.1007/978-3-319-46475-6_43

Jurkovic, Z., Cukor, G., Brezocnik, M., & Brajkovic, T. (2018). A comparison of machine learning methods for cutting parameters prediction in high speed turning process. *Journal of Intelligent Manufacturing*, *29*(8), 1683–1693. https://doi.org/10.1007/s10845-016-1206-1

Karagkouni, D., Paraskevopoulou, M. D., Chatzopoulos, S., Vlachos, I. S., Tastsoglou, S., Kanellos, I., Papadimitriou, D., Kavakiotis, I., Maniou, S., Skoufos, G., & others. (2017). DIANA-TarBase v8: a decade-long collection of experimentally supported miRNA--gene interactions. *Nucleic Acids Research*, *46*(D1), D239--D245.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2019). Analyzing and improving the image quality of StyleGAN. *ArXiv*, 8110–8119.

Kasuya, E. (2019). On the use of r and r squared in correlation and regression. *Ecological Research*, *34*(1), 235–236. https://doi.org/10.1111/1440-1703.1011

Kedem, B., & Fokianos, K. (2005). *Regression models for time series analysis* (Vol. 488). John Wiley \& Sons.

Khaleghi, B., Khamis, A., Karray, F. O., & Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, *14*(1), 28–44. https://doi.org/10.1016/j.inffus.2011.08.001

King, R. D., Feng, C., & Sutherland, A. (1995). Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence an International Journal*, *9*(3), 289–333.

Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. *Big Data & Society*, *1*(1), 205395171452848. https://doi.org/10.1177/2053951714528481

Kletz, T. A. (1988). *Learning from accidents in industry*. Butterworth-Heinemann.

Kourniotis, S. P., Kiranoudis, C. T., & Markatos, N. C. (2000). Statistical analysis of domino chemical accidents. *Journal of Hazardous Materials*, *71*(1–3), 239–252.

Krämer, S., & Engell, S. (2018). *Resource efficiency of processing plants: monitoring and improvement*. John Wiley & Sons.

Kusiak, A. (2020). Convolutional and generative adversarial neural networks in manufacturing. *International Journal of Production Research*, *58*(5), 1594–1604. https://doi.org/10.1080/00207543.2019.1662133

Lahat, D., Adali, T., & Jutten, C. (2015). Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. *Proceedings of the IEEE*, *103*(9), 1449–1477. https://doi.org/10.1109/JPROC.2015.2460697

Lanzetti, N., Lian, Y. Z., Cortinovis, A., Dominguez, L., Mercangöz, M., & Jones, C. (2019). Recurrent neural network based MPC for process industries. *2019 18th European Control Conference (ECC)*, 1005–1010.

Lapedes, A., & Farber, R. (1987). *Nonlinear signal processing using neural networks: Prediction and system modelling*.

Larsson, T., Hestetun, K., Hovland, E., & Skogestad, S. (2001). Self-optimizing control of a large-scale plant: The Tennessee Eastman process. *Industrial & Engineering Chemistry Research*, *40*(22), 4889–4901.

Larsson, T., & Skogestad, S. (2000). *Plantwide control-A review and a new design procedure*.

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, *6*(4), 239–242.

Lau, B. P. L., Marakkalage, S. H., Zhou, Y., Hassan, N. U., Yuen, C., Zhang, M., & Tan, U. X. (2019). A survey of data fusion in smart city applications. *Information Fusion*, *52*(January), 357–374. https://doi.org/10.1016/j.inffus.2019.05.004

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

LeCun, Y., Bengio, Y., & others. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(10), 1995.

Lee, D., Siu, V., Cruz, R., & Yetman, C. (2016). Convolutional Neural Net and Bearing Fault Analysis. *Int'l Conf. Data Mining*, 194–200. https://doi.org/https://pdfs.semanticscholar.org/6e45/f39b1e50cfd10deaabd1d786f

Lee, K. B., Cheon, S., & Kim, C. O. (2017). A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, *30*(2), 135–142. https://doi.org/10.1109/TSM.2017.2676245

Li, M. J., & Tao, W. Q. (2017). Review of methodologies and polices for evaluation of energy efficiency in high energy-consuming industry. *Applied Energy*, *187*, 203–215. https://doi.org/10.1016/j.apenergy.2016.11.039

Li, P., Chen, Z., & Zhang, J. (2020). *A Survey on Deep Learning for Multimodal Data Fusion*.

Li, Xiang, Zhang, W., Ding, Q., & Sun, J. Q. (2020). Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. *Journal of Intelligent Manufacturing*, *31*(2), 433–452. https://doi.org/10.1007/s10845-018-1456-1

Li, Xin, Liang, Y., Zhao, M., Wang, C., & Jiang, Y. (2019). Few-shot learning with generative adversarial networks based on WOA13 data. *Computers, Materials and Continua*, *60*(3), 1073–1085.

Li, Z., Wang, Y., & Wang, K. (2020). A data-driven method based on deep belief networks for backlash error prediction in machining centers. *Journal of Intelligent Manufacturing*, *31*(7),

1693–1705. https://doi.org/10.1007/s10845-017-1380-9

Lindberg, C. F., Tan, S., Yan, J., & Starfelt, F. (2015). Key Performance Indicators Improve Industrial Performance. *Energy Procedia*, *75*, 1785–1790. https://doi.org/10.1016/j.egypro.2015.07.474

Lipton, Z. C., Elkan, C., & Naryanaswamy, B. (2014). Optimal thresholding of classifiers to maximize F1 measure. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 225–239.

Liu, M. Y., Huang, X., Yu, J., Wang, T. C., & Mallya, A. (2021). Generative Adversarial Networks for Image and Video Synthesis: Algorithms and Applications. *Proceedings of the IEEE*, *109*(5), 839–862. https://doi.org/10.1109/JPROC.2021.3049196

Liu, X., Yin, G., Shao, J., Wang, X., & Li, H. (2019). Learning to predict layout-to-image conditional convolutions for semantic image synthesis. *ArXiv Preprint ArXiv:1910.06809*.

Long, M., Peng, F., & Zhu, Y. (2019). Identifying natural images and computer generated graphics based on binary similarity measures of PRNU. *Multimedia Tools and Applications*, *78*(1), 489–506.

Lv, F., Wen, C., Bao, Z., & Liu, M. (2016). Fault diagnosis based on deep learning. *Proceedings of the American Control Conference*, *2016-July*(2), 6851–6856. https://doi.org/10.1109/ACC.2016.7526751

Mandreoli, F., & Montangero, M. (2019). Dealing With Data Heterogeneity in a Data Fusion Perspective: Models, Methodologies, and Algorithms. In *Data Handling in Science and Technology* (Vol. 31). https://doi.org/10.1016/B978-0-444-63984-4.00009-0

Mart\’\in-Lopo, M. M., Boal, J., & Sánchez-Miralles, Á. (2020). A literature review of IoT energy platforms aimed at end users. *Computer Networks*, *171*, 107101.

Martens, H. (2015). Quantitative Big Data: where chemometrics can contribute. *Journal of Chemometrics*, *29*(11), 563–581. https://doi.org/10.1002/cem.2740

Mathieu, M., Couprie, C., & LeCun, Y. (2016). Deep multi-scale video prediction beyond mean square error. *4th International Conference on Learning Representations, ICLR 2016 -*

Conference Track Proceedings, *2015*, 1–14.

McAvoy, T. J., & Ye, N. (1994). Base control for the Tennessee Eastman problem. *Computers & Chemical Engineering*, *18*(5), 383–413.

McEntyre, J., & Lipman, D. (2001). PubMed: bridging the information gap. *Cmaj*, *164*(9), 1317–1319.

Meng, T., Jing, X., Yan, Z., & Pedrycz, W. (2020). A survey on machine learning for data fusion. *Information Fusion*, *57*(2), 115–129. https://doi.org/10.1016/j.inffus.2019.12.001

Mirza, M., & Osindero, S. (2014). *Conditional Generative Adversarial Nets*. 1–7. http://arxiv.org/abs/1411.1784

Narayanasamy, R., & Padmanabhan, P. (2012). Comparison of regression and artificial neural network model for the prediction of springback during air bending process of interstitial free steel sheet. *Journal of Intelligent Manufacturing*, *23*(3), 357–364. https://doi.org/10.1007/s10845-009-0375-6

National Inventory Report. (2019). *GREENHOUSE GAS SOURCES AND SINKS IN CANADA CANADA'S SUBMISSION TO THE UNITED NATIONS FRAMEWORK CONVENTION ON CLIMATE CHANGE Executive Summary*.

Neurohive. (n.d.). *FaceStyleGAN: GAN Network Generates Style Portraits in Snapchat*. https://neurohive.io/en/news/facestylegan-gan-network-generates-style-portraits-in-snapchat/

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems*, 841–848.

Olmschenk, G., Zhu, Z., & Tang, H. (2019). Generalizing semi-supervised generative adversarial networks to regression using feature contrasting. *Computer Vision and Image Understanding*, *186*(June), 1–12. https://doi.org/10.1016/j.cviu.2019.06.004

Olsina, L., & Martin, M. de los A. (2004). Ontology for software metrics and indicators. *J. Web Eng.*, *2*(4), 262–281.

Om, H., & Kundu, A. (2012). A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, 131–136.

Pan, R. (2010). H olt--W inters Exponential Smoothing. *Wiley Encyclopedia of Operations Research and Management Science*.

Park, T., Liu, M.-Y., Wang, T.-C., & Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2337–2346.

Parmenter, D. (2020). *Key perfomance indicator developing, implementing and using winning KPIs*. John Wiley & Sons.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, D., Liu, Z., Wang, H., Qin, Y., & Jia, L. (2019). A novel deeper one-dimensional CNN with residual learning for fault diagnosis of wheelset bearings in high-speed trains. *IEEE Access*, *7*, 10278–12093. https://doi.org/10.1109/ACCESS.2018.2888842

Petzold, B., Roggendorf, M., Rowshankish, K., & Sporleder, C. (2020). Designing data governance that delivers value. *Mc Kinsey Digital*, *June*. https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/designing-data-governance-that-delivers-value

Qi, X., Yuan, Z., & Han, X. (2015). Diagnosis of misalignment faults by tacholess order tracking analysis and RBF networks. *Neurocomputing*, *169*, 439–448. https://doi.org/10.1016/j.neucom.2014.09.088

Qiu, X., Zhang, L., Ren, Y., Suganthan, P., & Amaratunga, G. (2014). Ensemble deep learning for regression and time series forecasting. *IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - CIEL 2014: 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning, Proceedings*. https://doi.org/10.1109/CIEL.2014.7015739

Ragab, A., El-Koujok, M., Poulin, B., Amazouz, M., & Yacout, S. (2018). Fault diagnosis in

industrial chemical processes using interpretable patterns based on Logical Analysis of Data. *Expert Systems with Applications*, *95*, 368–383. https://doi.org/https://doi.org/10.1016/j.eswa.2017.11.045

Ragab, A., Koujok, M. El, Ghezzaz, H., Amazouz, M., Ouali, M.-S., & Yacout, S. (2019). Deep understanding in industrial processes by complementing human expertise with interpretable patterns of machine learning. *Expert Systems with Applications*, *122*, 388–405. https://doi.org/https://doi.org/10.1016/j.eswa.2019.01.011

Ragab, A., Ouali, M. S., Yacout, S., & Osman, H. (2016). Remaining useful life prediction using prognostic methodology based on logical analysis of data and Kaplan–Meier estimation. *Journal of Intelligent Manufacturing*, *27*(5), 943–958. https://doi.org/10.1007/s10845-014-0926-3

Ragab, A., Yacout, S., Ouali, M. S., & Osman, H. (2019). Prognostics of multiple failure modes in rotating machinery using a pattern-based classifier and cumulative incidence functions. *Journal of Intelligent Manufacturing*, *30*(1), 255–274. https://doi.org/10.1007/s10845-016-1244-8

Ricker, N. L. (1996). Decentralized control of the Tennessee Eastman challenge process. *Journal of Process Control*, *6*(4), 205–221.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., & Waldman-Brown, A. (2019). Tackling climate change with machine learning. *ArXiv Preprint ArXiv:1906.05433*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015* (pp. 234–241). Springer International Publishing.

Sakpal, M. (2021). *12 Actions to Improve Your Data Quality*. https://www.gartner.com/smarterwithgartner/how-to-improve-your-data-quality

Sanderson, C., & Gruen, R. (2006). *Analytical models for decision-making*. McGraw-Hill Education (UK).

Santamaria, R., Therón, R., & Quintales, L. (2008). A visual analytics approach for understanding biclustering results from microarray data. *BMC Bioinformatics*, *9*(1), 247.

Schat, E., van de Schoot, R., Kouw, W. M., Veen, D., & Mendrik, A. M. (2020). The data representativeness criterion: Predicting the performance of supervised classification based on data set similarity. *PLoS ONE*, *15*(8 August), 1–16. https://doi.org/10.1371/journal.pone.0237009

Schmidt, C., Li, W., Thiede, S., Kornfeld, B., Kara, S., & Herrmann, C. (2016). Implementing Key Performance Indicators for Energy Efficiency in Manufacturing. *Procedia CIRP*, *57*, 758–763. https://doi.org/10.1016/j.procir.2016.11.131

Shao, S., Wang, P., & Yan, R. (2019). Generative adversarial networks for data augmentation in machine fault diagnosis. *Computers in Industry*, *106*, 85–93. https://doi.org/10.1016/j.compind.2019.01.001

Shao, S. Y., Sun, W. J., Yan, R. Q., Wang, P., & Gao, R. X. (2017). A Deep Learning Approach for Fault Diagnosis of Induction Motors in Manufacturing. *Chinese Journal of Mechanical Engineering (English Edition)*, *30*(6), 1347–1356. https://doi.org/10.1007/s10033-017-0189-y

Shen, Y., Yang, F., Habibullah, M. S., Ahmed, J., Das, A. K., Zhou, Y., & Ho, C. L. (2020). Predicting tool wear size across multi-cutting conditions using advanced machine learning techniques. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01625-7

Siirtola, H., & Räihä, K.-J. (2006). Interacting with parallel coordinates. *Interacting with Computers*, *18*(6), 1278–1309.

Simmonds, A., Sandilands, P., & Van Ekert, L. (2004). An ontology for network security attacks. *Asian Applied Computing Conference*, 317–323.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556*.

Soualhi, M., El Koujok, M., Nguyen, K. T. P., Medjaher, K., Ragab, A., Ghezzaz, H., Amazouz, M., & Ouali, M.-S. (2021). Adaptive prognostics in a controlled energy conversion process

based on long-and short-term predictors. *Applied Energy*, *283*, 116049.

Sridharan, M. (2017). *Using governance to achieve data quality objectives*. https://www.bloomberg.com/professional/blog/using-governance-achieve-data-quality-objectives/

Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015). Unsupervised learning of video representations using lstms. *International Conference on Machine Learning*, 843–852.

Talbot, D., & Boiral, O. (2013). Can we trust corporates GHG inventories? An investigation among Canada's large final emitters. *Energy Policy*, *63*, 1075–1085.

Tang, H., Qi, X., Xu, D., Torr, P. H. S., & Sebe, N. (2020). Edge guided GANs with semantic preserving for semantic image synthesis. *ArXiv Preprint ArXiv:2003.13898*.

Telea, A. C. (2007). Data Visualization: Principles and practice. In *Data Visualization: Principles and Practice*. https://doi.org/10.1201/b10679

Tian, J., Morillo, C., Azarian, M. H., & Pecht, M. (2016). Motor Bearing Fault Detection Using Spectral Kurtosis-Based Feature Extraction Coupled With K-Nearest Neighbor Distance Analysis. *IEEE Transactions on Industrial Electronics*, *63*(3), 1793–1803. https://doi.org/10.1109/TIE.2015.2509913

Tidriri, K., Chatti, N., Verron, S., & Tiplica, T. (2016). Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual Reviews in Control*, *42*, 63–81.

Uguz, S., & Ipek, O. (2021). Prediction of the parameters affecting the performance of compact heat exchangers with an innovative design using machine learning techniques. *Journal of Intelligent Manufacturing*, *2013*. https://doi.org/10.1007/s10845-020-01729-0

Uysal, M. P., & Sogut, M. Z. (2017). An integrated research for architecture-based energy management in sustainable airports. *Energy*, *140*, 1387–1397.

Vakkilainen, E., & others. (2005). *Kraft recovery boilers--Principles and practice*.

Vasara, P. (2001). Scandinavia: Through different eyes: environmental issues in Scandinavia and North America. *Tappi Journal*, *84*(6), 46–49.

Vlachos, I. P. (2014). A hierarchical model of the impact of RFID practices on retail supply chain performance. *Expert Systems with Applications*, *41*(1), 5–15. https://doi.org/10.1016/j.eswa.2013.07.006

Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Generating videos with scene dynamics. *Advances in Neural Information Processing Systems*, *29*, 613–621.

Wald, L. (1999). *Definitions and terms of reference in data fusion*.

Wang, Jin, Yang, Y., Wang, T., Sherratt, R. S., & Zhang, J. (2020). Big data service architecture: a survey. *Journal of Internet Technology*, *21*(2), 393–405.

Wang, Jinjiang, Yan, J., Li, C., Gao, R. X., & Zhao, R. (2019). Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction. *Computers in Industry*, *111*, 1–14. https://doi.org/10.1016/j.compind.2019.06.001

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8798–8807.

Wang, Y., Zhou, J., Zheng, L., & Gogu, C. (2020). An end-to-end fault diagnostics method based on convolutional neural network for rotating machinery with multiple case studies. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01671-1

Wang, Z., & Bovik, A. C. (2002). A universal image quality index. *IEEE Signal Processing Letters*, *9*(3), 81–84.

Wegman, E. J. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, *85*(411), 664–675.

White, F. E. (1991). JDL, data fusion lexicon. *Technical Panel for C*, *3*, 15.

Wilke, C. O. (2019). Fundamentals of Data Visualization. In *Serial Mentor*.

Wong, P. C., & Bergeron, R. D. (1996). Multiresolution multidimensional wavelet brushing. *Proceedings of Seventh Annual IEEE Visualization'96*, 141–148.

Wu, A. (n.d.). *A Chat with Andrew on MLOps: From Model-Centric to Data-Centric AI. 2021*.

Wu, H., & Zhao, J. (2018). Deep convolutional neural network model based chemical process fault

diagnosis. *Computers and Chemical Engineering*, *115*, 185–197. https://doi.org/10.1016/j.compchemeng.2018.04.009

Wu, J., Zhang, C., Xue, T., Freeman, W. T., & Tenenbaum, J. B. (2016). Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. *Advances in Neural Information Processing Systems*, *Nips*, 82–90.

Xia, C., Pan, Z., Polden, J., Li, H., Xu, Y., & Chen, S. (2021). Modelling and prediction of surface roughness in wire arc additive manufacturing using machine learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01725-4

Xu, Y., Pei, J., & Lai, L. (2017). Deep Learning Based Regression and Multiclass Models for Acute Oral Toxicity Prediction with Automatic Chemical Feature Extraction. *Journal of Chemical Information and Modeling*, *57*(11), 2672–2685. https://doi.org/10.1021/acs.jcim.7b00244

Yin, S., Ding, S. X., Haghani, A., Hao, H., & Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control*, *22*(9), 1567–1581. https://doi.org/https://doi.org/10.1016/j.jprocont.2012.06.009

Yourtechdiet. (n.d.). *Top Tools in Generative Adversarial Networks*. https://www.yourtechdiet.com/blogs/generative-adversarial-networks-tools/

Yuan, X., Li, L., Shardt, Y. A. W., Wang, Y., & Yang, C. (2021). Deep Learning with Spatiotemporal Attention-Based LSTM for Industrial Soft Sensor Model Development. *IEEE Transactions on Industrial Electronics*, *68*(5), 4404–4414. https://doi.org/10.1109/TIE.2020.2984443

Zagrebina, S. A., Mokhov, V. G., & Tsimbol, V. I. (2019). Electrical energy consumption prediction is based on the recurrent neural network. *Procedia Computer Science*, *150*, 340–346.

Zakharov, E., Shysheya, A., Burkov, E., & Lempitsky, V. (2019). Few-shot adversarial learning of realistic neural talking head models. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 9458–9467. https://doi.org/10.1109/ICCV.2019.00955

Zhang, D., Li, Q., Yang, G., Li, L., & Sun, X. (2017). Detection of image seam carving by using

weber local descriptor and local binary patterns. *Journal of Information Security and Applications*, *36*, 135–144.

Zhang, J., Zhong, S., Wang, T., Chao, H.-C., & Wang, J. (2020). Blockchain-based systems and applications: a survey. *Journal of Internet Technology*, *21*(1), 1–14.

Zhang, Qing, Gao, J., Dong, H., & Mao, Y. (2018). WPD and DE/BBO-RBFNN for solution of rolling bearing fault diagnosis. *Neurocomputing*, *312*, 27–33. https://doi.org/10.1016/j.neucom.2018.05.014

Zhang, Qingchen, Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, *42*, 146–157.

Zhang, S., Zhang, S., Wang, B., & Habetler, T. G. (2019). Machine learning and deep learning algorithms for bearing fault diagnostics-a comprehensive review. *ArXiv Preprint ArXiv:1901.08247*.

Zhang, W., Li, X., Jia, X. D., Ma, H., Luo, Z., & Li, X. (2020). Machinery fault diagnosis with imbalanced data using deep generative adversarial networks. *Measurement: Journal of the International Measurement Confederation*, *152*. https://doi.org/10.1016/j.measurement.2019.107377

Zhang, Z., & Zhao, J. (2017). A deep belief network based fault diagnosis model for complex chemical processes. *Computers and Chemical Engineering*, *107*, 395–407. https://doi.org/10.1016/j.compchemeng.2017.02.041

Zhao, D., Ivanov, M., Wang, Y., & Du, W. (2020). Welding quality evaluation of resistance spot welding based on a hybrid approach. *Journal of Intelligent Manufacturing*, *0123456789*. https://doi.org/10.1007/s10845-020-01627-5

Zhao, M., Liu, X., Yao, X., & He, K. (2020). Better visual image super-resolution with Laplacian pyramid of generative adversarial networks. *CMC-COMPUTERS MATERIALS \& CONTINUA*, *64*(3), 1601–1614.

Zheng, H., Liao, H., Chen, L., Xiong, W., Chen, T., & Luo, J. (2020). Example-guided image synthesis across arbitrary scenes using masked spatial-channel attention and self-supervision. *ArXiv Preprint ArXiv:2004.10024*.

Zhou, P., Ang, B. W., & Zhou, D. Q. (2012). Measuring economy-wide energy efficiency performance: A parametric frontier approach. *Applied Energy*, *90*(1), 196–200. https://doi.org/10.1016/j.apenergy.2011.02.025

Zhou, W., Li, X., Yi, J., & He, H. (2019). A Novel UKF-RBF Method Based on Adaptive Noise Factor for Fault Diagnosis in Pumping Unit. *IEEE Transactions on Industrial Informatics*, *15*(3), 1415–1424. https://doi.org/10.1109/TII.2018.2839062

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2223–2232.

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., & Shechtman, E. (2017). Multimodal Image-to-Image Translation by Enforcing Bi-Cycle Consistency. *Advances in Neural Information Processing Systems*, 465–476.

Zhu, L., Johnsson, C., Mejvik, J., Varisco, M., & Schiraldi, M. (2018). Key performance indicators for manufacturing operations management in the process industry. *IEEE International Conference on Industrial Engineering and Engineering Management*, *2017-Decem*, 969–973. https://doi.org/10.1109/IEEM.2017.8290036

Zitnik, M., Nguyen, F., Wang, B., Leskovec, J., Goldenberg, A., & Hoffman, M. M. (2019). Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, *50*(September 2018), 71–91. https://doi.org/10.1016/j.inffus.2018.09.012

CHAPTER 5 ARTICLE 3: POLYGON GENERATION AND VIDEO-TO-VIDEO TRANSLATION FOR TIME-SERIES PREDICTION IN INDUSTRIAL SYSTEMS

Mohamed Tarek Mohamed Elhefnawy, Ahmed Ragab, Mohamed-Salah Ouali

Submitted to: COMPUTERS IN INDUSTRY

Manuscript Number: COMIND-D-21-01558

5.1 Abstract

This paper proposes an innovative method for time-series prediction in industrial systems characterized by highly dynamic non-linear operations. The proposed method can capture the true distributions of the inputs and outputs of such systems and map these distributions using polygon generation and video-to-video translation techniques. More specifically, the time-series data are represented as polygon streams (videos), then the video-to-video translation is used to transform the input polygon streams into the output ones. This transformation is tuned based on a model trustworthiness metric for optimal video synthesis. Finally, an image processing procedure is used for mapping the output polygon streams back to time-series outputs. The proposed method is based on cycle-consistent generative adversarial networks as an unsupervised approach. This does not need the heavy involvement of the human expert who devotes much effort to label the complex industrial data. The performance of the proposed method was validated successfully using a challenging industrial dataset collected from a piece of complex equipment in a heat exchanger network of a Canadian pulp mill. The results obtained demonstrate better performance of the proposed method than other comparable time-series prediction models.

5.2 Introduction

Climate change is one of the most important challenges that is urgent to be tackled due to its dangerous effects on different natural aspects (Environment challenges | Climate Action). Increase of greenhouse gas (GHG) emissions in the atmosphere is one of the main reasons for this climate change challenge worldwide. The use of fossil fuels in heavy industries have primarily led to such emissions. In Canada, GHG emissions increased form 600 mega tonnes of carbon dioxide equivalent (Mt CO_2 eq.) in 1990 to 730 Mt CO_2 eq. in 2019 (increase by 21.4%) (National Inventory Report, 2019). According to that report, oil and gas industry (26%) and transport (25%) are the primary causes of such growth of Canada's emissions. Among the reasons for the GHG emissions and excessive energy consumption of such industries are the inefficient monitoring and control of such complex and highly dynamic processes. In such processes, a set of key performance indicators (KPIs) are used for monitoring their health state.

SYMBOL	DEFINITION
$\boldsymbol{X_j}$	j^{th} input variable
$\boldsymbol{Y_h}$	h^{th} output
$\hat{\boldsymbol{q}}, \hat{\boldsymbol{l}}$	The unit vector of x and y directions respectively
$\overline{\boldsymbol{X_j}}$	Mean of jth input variable
$\overline{\boldsymbol{Y_h}}$	Mean of hth output
$\boldsymbol{\delta_j}$	Standard deviation of jth input variable
$\boldsymbol{\delta_h}$	Standard deviation of hth output
$\boldsymbol{x_{kj}}$	Value of jth input variable for k^{th} observation
$\boldsymbol{y_{kh}}$	Value of hth output for k^{th} observation
$\boldsymbol{Z_{kj}}$	Standardized value of jth input variable for k^{th} observation
$\boldsymbol{Z_{kh}}$	Standardized value of hth output for k^{th} observation
$\widehat{\boldsymbol{X_j}}$	Unit vector of the polygon side representing j^{th} input variable
$\widehat{\boldsymbol{Y_h}}$	Unit vector of the polygon side representing h^{th} output
$\overrightarrow{\boldsymbol{X_j}}$	Point coordinates of the zero standardized value of the variable X_j
$\overrightarrow{\boldsymbol{Y_h}}$	Point coordinates of the zero standardized value of the output Y_h
$\overrightarrow{\boldsymbol{X_j^k}}$	Point coordinates of the standardized values of kth observation for variable X_j
$\overrightarrow{\boldsymbol{Y_h^k}}$	Point coordinates of the standardized values of kth observation for output Y_h
$\boldsymbol{G_{B/A}}$	Generator of the GAN architecture translating observations from domain B to domain A
$\boldsymbol{G_{A/B}}$	Generator of the GAN architecture translating observations from domain A to domain B
$\boldsymbol{D_A}$	Discriminator of the GAN architecture for observations generated from domain A
$\boldsymbol{D_B}$	Discriminator of the GAN architecture for observations generated from domain B
$\boldsymbol{L_{GAN}}$	Adversarial loss of the GAN architecture
$\boldsymbol{L_{cyc}}$	Cycle consistency loss of the GAN architecture
$\boldsymbol{\mathcal{L}}$	Total loss of the GAN architecture
$\boldsymbol{Q}$	Universal image quality index
$\bar{\boldsymbol{x}}$	Mean of the pixel values of the frames of the polygon stream representing the outputs
$\bar{\boldsymbol{y}}$	Mean of the pixel values of the frames of the synthesized polygon stream using the GAN model
$\boldsymbol{\sigma_x}$	Standard deviation of the pixel values of the frames of the polygon stream representing the
$\boldsymbol{\sigma_y}$	Standard deviation of the pixel values of the frames of the synthesized polygon stream using the
$\boldsymbol{\alpha_j}$	The underestimation parameter of each KPI_j in the penalty function
$\boldsymbol{\beta_j}$	The overestimation parameter of each KPI_j in the penalty function
$\boldsymbol{L^j(t_k)}$	The penalty function of each KPI_j at instance t_k
$\boldsymbol{L^j_{AP}}$	The average penalty function of each KPI_j
$\boldsymbol{L_{model}}$	The total average penalty function of each time-series regression model

Inefficient control of these KPIs results in various environmental and economic impacts in terms of harmful emissions, excessive maintenance, and unexpected downtime (Andersson & Thollander, 2019). Therefore, developing an accurate prediction model for these KPIs is an urgent need for the sake of accurate KPIs monitoring and optimization that help mitigate such environmental impacts and economic losses (Rolnick et al., 2019).

Most of heavy industrial systems are characterized by highly nonlinear and dynamic operation which make monitoring and prediction of their KPIs more challenging. These nonlinear processes are hard to model and predict their unexpected responses based on the expert knowledge alone. The system response is continuously changing using the same inputs at different time instants. Moreover, the superposition principle cannot apply, and therefore dealing with multiple input variables is a tedious task. Fortunately, these industrial systems are equipped with numerous number of sensors that acquire huge amount of data of different types. One of the major data sources available in such industrial systems is the time-series data. This time-series data acts as an important opportunity to build accurate data-driven models using machine learning (ML) techniques. Data-driven modeling shows promising solution compared to classical analytical techniques such as autoregressive integrated moving average (ARIMA), simple exponential smoothing (SES), Holt Winter's exponential smoothing (HWES) (Box et al., 2015; Brown & Meyer, 1961; Kedem & Fokianos, 2005; Pan, 2010) which are not effective in case of highly dynamic complex systems with several interacted components. However, most of machine learning techniques such as artificial neural network (ANN), decision trees and support vector regression (SVR) used for time-series prediction in the industry made assumptions and cannot capture the actual distribution of data of such non-stationary dynamic processes (Alpaydin, 2010; Franklin, 2005; Lapedes & Farber, 1987). This may result in inaccurate performance of these models and hinder their deployment in such cases.

Another challenge in processing the industrial data is the labeling phase. Correct labeling of the industrial data in alignment with the input variables is an indispensable need for training and testing various data-driven modeling techniques. Unfortunately, labeling of this type of data is a tedious process that heavily involves the human experience which is rare. Even though with existence of such expertise, the labeling process may not be done in an appropriate way that leads to inefficient model building. Fortunately, the deep learning (DL) approach offers an opportunity to tackle the above mentioned limitations. It has been proven that DL achieves better predictive performance compared to other classical ML predictors (LeCun et al., 2015; Lv et al., 2016; Goodfellow, Bengio, 2017).

For accurate time-series prediction, there is a need to learn a mapping function that converts the input time-series variables into the targeted outputs (KPIs in the industrial context). In other

words, a data distribution matching problem needs to be solved aiming to train a model such that the conditional distribution of the predicted KPIs given the input variables resembles that of real KPIs. Conditional generative modeling can be a promising approach for solving this type of problems (M. Y. Liu et al., 2021). One of the state-of-art DL conditional generative modeling techniques is the conditional generative adversarial networks (cGANs) (Goodfellow et al., 2014; Isola et al., 2017) that are used for data augmentation, mapping of images or videos from one domain to another, creating image filters and others.

The distribution matching can be facilitated through a better data representation (Schat et al., 2020). In fact, data representation pathway is the optimal approach for practitioners and researchers in the DL field. These researchers are still developing new architectures and/or optimizing the existing ones without looking over the available data and maximize their value before exploitation. The available data acts as a fuel for training the DL architectures. Accordingly, focusing on improving the data representation is an urgent need for better modeling performance. Data-centric AI (Andrew Ng Launches A Campaign For Data-Centric AI, 2021.; Wu, 2021) is an emerging approach nowadays for improving the quality of data used for training DL models. Researchers and practitioners are recently starting to organize several occasions with the goal of obtaining the best data representation that achieve the highest prediction performance using the same DL architecture. The Data-Centric AI Competition Hackathon is one of these occasions (Data-Centric AI Competition, 2021).

Given the above mentioned limitations and opportunities of exploiting the generative DL modeling power of cGANs and improving data representation using data-centric AI approaches, this motivates us to use the polygon generation technique proposed in (Elhefnawy et al., 2021a) to transform the time-series data into polygon streams (videos). These videos represent all interrelationships between the time-series inputs and their change over time using Hamiltonian cycles. For mapping the polygon streams of the input variables into that of the KPIs (outputs), we propose to use the video-to-video translation (3D-CycleGAN) technique introduced in (Bashkirova et al., 2018). This technique is based on the cycle-consistent generative adversarial networks as an unsupervised method in which the data does not need to be paired, accordingly, it saves the effort of the labeling process done by the process expert. A model trustworthiness metric is used for tuning the 3D-CycleGAN to ensure the consistency of the acquired polygon streams with the

original polygon streams. After obtaining the translated polygon streams representing the predicted KPIs, an image processing procedure is applied for every video frame to recover the numerical values of the KPIs. This proposed method is validated using a concentrator equipment in a pulp & paper mill located in Canada and the results show that it outperformed other common DL time-series predictors. The method accurately predicts three important KPIs in the concentrator: the evaporated water, the concentrator efficiency, and the fouling index. This helps the mill mitigate environmental impacts and economic losses.

The rest of this paper is organized as follows. Section 2 provides a background on polygon generation for time-series data and unsupervised video-to-video translation (3D-CycleGAN), in addition to some related work on DL time-series prediction in industrial systems. Section 3 presents the proposed method with its detailed steps. Section 4 shows the industrial case study: the concentrator equipment used to validate the proposed method and the experimental setup. Section 5 discusses the results and gives insights and future work directions. Finally, Section 6 concludes the paper.

5.3 Background & Related Work

This section discusses the background and related work to time-series prediction in industrial systems using deep learning methods. It also presents the two main methods used in the proposed method to tackle the problem of time-series prediction. The first method is the polygon generation as an efficient data representation technique that converts the numeric time-series observations into polygon streams (videos). The second one is the video-to-video translation method that maps polygon videos of time-series inputs into outputs (KPIs).

5.3.1 Deep Learning for Time-series Prediction in Industrial Systems

The DL has become an opportunity for developing more accurate time-series predictive models in highly dynamic industrial systems compared to classical machine learning algorithms (Gamboa, 2017). Convolutional neural network (CNN) is one of the DL architectures that are used commonly in image, speech and time-series data (Borovykh et al., 2017; Huang et al., 2015; LeCun et al., 1995). The interested readers can find more applications on DL in time series prediction in

the comprehensive review papers (Gamboa, 2017; Han et al., 2021). In what follows, some related work are presented.

A deep CNN combined with an adaptive time-series window (ATSW) is used in (Hoermann et al., 2018) and validated using a time-series data collected from an industrial furnace. Another augmented multi-dimensional CNN is used in (Hoermann et al., 2018) for industrial soft sensing. Recurrent DL architectures such as LSTM (Gers et al., 2000) has been used extensively in the literature for time-series prediction. A convolutional LSTM encoder-decoder architecture is proposed in (Essien & Giannetti, 2020) for smart manufacturing and validated using real data from an industrial plant in United Kingdom. A spatiotemporal attention-based LSTM is used in (Yuan et al., 2021) for developing industrial soft sensor models. Besides, for quality prediction in manufacturing, LSTM is used in (Bai et al., 2021) as a regression tool along with AdaBoost for model's reinforcement. An LSTM architecture is used in (Soualhi et al., 2021) in the pulp and paper industry using a dataset collected from a heat exchanger located in Canada. Another architecture close to that of LSTM is called gated recurrent unit (GRU) which has less number of gates (less parameters) and used in case of smaller datasets (Cho et al., 2014). A bidirectional GRU with weighted features averaging is used in (Jinjiang Wang et al., 2019) for smart manufacturing.

However, most of the architectures used in the literature works in a supervised way with paired inputs and outputs for training. This pairing needs an additional effort by the process expert. In addition, it is hard to capture the true distribution of the complex industrial data in case of highly dynamic nonlinear process. Therefore, more focus is needed for better data representation for the maximal exploitation of the available industrial data.

5.3.2 Polygon Generation for Data Representation

A data representation technique called "Polygon Generation" was proposed in (Elhefnawy et al., 2021a) to map the numerical observations into polygon images. These polygon images are used for training a deep learning model for accurate classification. This technique was validated using a challenging dataset collected from a reboiler system in a pulp and paper mill located in Canada. Due to its effectiveness in representing the numerical data, we are motivated in this work to adopt that polygon generation technique for time-series prediction in highly dynamic industrial

processes. More detailed steps on polygon generation are illustrated via an illustrative example in (Elhefnawy et al., 2021a). In what follows, we summarize these steps through another toy example.

This numerical example comprises numerical data with six input variables and four outputs. Figure 5.1 shows a regular hexagon where each side represents an input variable. The point coordinates $\overrightarrow{X_j^k}$ (in orange) that represent the standardized values of observation k for the variable X_j are calculated using Eq. (1) and (2), where $j = 1,2,\ldots,6$.

$$Z_{kj} = \frac{x_{kj} - \overline{X_j}}{\delta_j} \qquad (1)$$

$$\overrightarrow{X_j^k} = \overrightarrow{X_j} + \left(Z_{kj} * \widehat{X_j}\right) \qquad (2)$$

where, x_{kj} is the actual numeric value of observation k for the variable X_j, $\overline{X_j}$ and δ_j are their mean value and standard deviation, respectively, Z_{kj} is the standardized value of observation k for the variable X_j, $\widehat{X_j}$ represents the unit vector of each polygon side and $\overrightarrow{X_j}$ are the point coordinates (in blue) that represent the zero standardized value of the variable X_j.

Table 5.1 shows the $\overrightarrow{X_j^k}$ values calculated using Eq. (1) and (2). Similarly, this procedure is applied to the numerical outputs. Figure 5.2 shows a square that represents the four outputs, where each side represents the output Y_h, where $h = 1,2,3,4$. Table 5.2 shows the calculations of $\overrightarrow{Y_h^k}$ (in orange) that represent the standardized outputs Z_{kh}, where y_{kh} is the numeric value of observation k for the output Y_h, $\overline{Y_h}$ and δ_h are their mean value and standard deviation, respectively, $\overrightarrow{Y_h}$ are the point coordinates (in blue) that represent the zero standardized value of the output Y_h and $\widehat{Y_h}$ represents the unit vector of each polygon side.

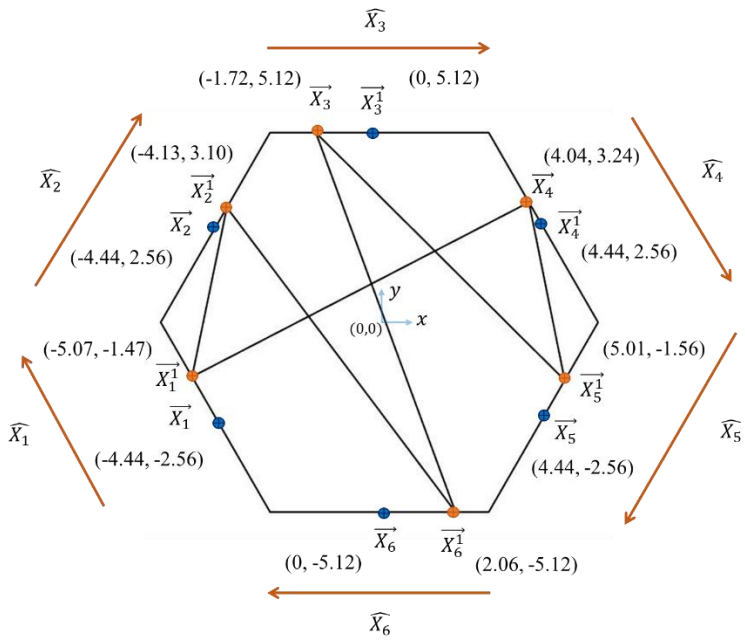


Figure 5.1 - A polygon generated from a numeric observation of six data variables using the method proposed in (Elhefnawy et al., 2021a). All variables are numbered in a clockwise direction.

Table 5.1 - Calculation of point coordinates $\overrightarrow{X_j^1}$ on the sides of the polygon for a numeric observation with six variables shown in Figure 5.1, where $\hat{q}$ and $\hat{l}$ are the unit vectors of x and y directions, respectively.

j	x_{1j}	$\overline{X_j}$	δ_j	Z_{1j}	$\widehat{X_j}$	$\overrightarrow{X_j}$	$\overrightarrow{X_j^1}$
1	69.55	68.54	0.8	1.26	$-0.5\,\hat{q} + 0.87\,\hat{l}$	$-4.44\,\hat{q} - 2.56\,\hat{l}$	$-5.07\,\hat{q} - 1.47\,\hat{l}$
2	125.24	118.12	11.48	0.62	$0.5\,\hat{q} + 0.87\,\hat{l}$	$-4.44\,\hat{q} + 2.56\,\hat{l}$	$-4.13\,\hat{q} + 3.10\,\hat{l}$
3	853.77	860.36	3.83	-1.72	$1\,\hat{q} + 0\,\hat{l}$	$0\,\hat{q} + 5.12\,\hat{l}$	$-1.72\,\hat{q} + 5.12\,\hat{l}$
4	8.95	11.42	3.13	-0.79	$0.5\,\hat{q} - 0.87\,\hat{l}$	$4.44\,\hat{q} + 2.56\,\hat{l}$	$4.04\,\hat{q} + 3.24\,\hat{l}$
5	4035.1	4120.38	74.16	-1.15	$-0.5\,\hat{q} - 0.87\,\hat{l}$	$4.44\,\hat{q} - 2.56\,\hat{l}$	$5.01\,\hat{q} - 1.56\,\hat{l}$
6	59.94	45.33	7.09	-2.06	$-1\,\hat{q} + 0\,\hat{l}$	$0\,\hat{q} - 5.12\,\hat{l}$	$2.06\,\hat{q} - 5.12\,\hat{l}$

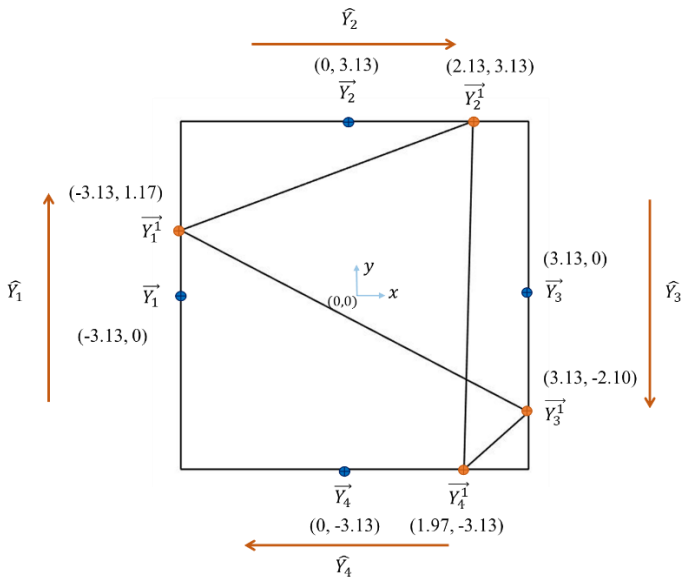


Figure 5.2 - A polygon generated from a numeric observation of the four outputs. All outputs are numbered in clockwise direction.

Table 5.2 - Calculation of point coordinates $\overrightarrow{Y_h^1}$ on polygon sides for a numeric observation with four outputs shown in Figure 5.2, where $\hat{q}$ and $\hat{l}$ are the unit vectors of the x and y directions, respectively.

h	y_{1h}	$\overline{Y_h}$	δ_h	Z_{1h}	$\widehat{Y_h}$	$\overrightarrow{Y_h}$	$\overrightarrow{Y_h^1}$
1	3.65	3.46	0.16	1.17	$0\,\hat{q} + 1\,\hat{l}$	$-3.13\,\hat{q} + 0\,\hat{l}$	$-3.13\,\hat{q} + 1.17\,\hat{l}$
2	102.29	23.88	36.81	2.13	$1\,\hat{q} + 0\,\hat{l}$	$0\,\hat{q} + 3.13\,\hat{l}$	$2.13\,\hat{q} + 3.13\,\hat{l}$
3	63.99	15.17	23.25	2.10	$0\,\hat{q} - 1\,\hat{l}$	$3.13\,\hat{q} + 0\,\hat{l}$	$3.13\,\hat{q} - 2.10\,\hat{l}$
4	0.19	8.47	4.2	-1.97	$-1\,\hat{q} + 0\,\hat{l}$	$0\,\hat{q} - 3.13\,\hat{l}$	$1.97\,\hat{q} - 3.13\,\hat{l}$

Figures Figure 5.1 and Figure 5.2 show one possible connection between the points on polygon sides representing the observation values for input variables and outputs, respectively.

This polygon generation technique represents all interrelationships between variables and outputs through Hamiltonian cycles (Elhefnawy et al., 2021a). Accordingly, each observation is represented as multiple images with all possible connections between points on polygon sides. The algorithm proposed in (Hurley & Oldford, 2010; Wegman, 1990) is used for this multiple images' generation step. More details are found in (Elhefnawy et al., 2021a).

5.3.3 Unsupervised Video-to-Video Translation

There are two main approaches in data-driven modeling; discriminative modeling and generative modeling (Ng & Jordan, 2002). Given input variables X and outputs Y, the discriminative modeling predicts the probability distribution of outputs Y given the variables X, denoted mathematically as $P(Y|X)$ whether in classification problems (categorical Y) or regression problems (continuous Y). The generative modeling on the other hand predicts the data distribution of the inputs X given the outputs Y ($P(X|Y)$) (Jebara, 2012).

One of the state-of-art techniques for generative modeling is the generative adversarial networks (GANs). They are first introduced in (Goodfellow et al., 2014). The GAN architecture has two main components (networks); generator and discriminator. The generator works on synthesizing some fake examples, acting as a forger who tries to mimic the real examples (images, text, videos, etc.) (See Figure 5.3). The discriminator works on assessing whether these synthesized examples are fake or not. It works as an inspector that tries not to be fooled by the forger (the generator). The generator synthesizes fake examples using only random noise and the feedback of the discriminator works on improving its quality over time. The two networks keep competing with each other and training in an adversarial way until the generator becomes a master forger that synthesize examples that are very close to the real ones. Consequently, the discriminator cannot detect if these synthesized examples are fake or real. At this stage, the training process is terminated and the generator model can be saved for later use in testing phase.

There are two different types of GANs; unconditional and conditional (Mirza & Osindero, 2014). Figure 5.3 shows the difference between both types. The generator of the conditional one has a random noise in addition to a control signal Y that can be a class label, image, video, or text that acts as a condition for the generator to synthesize observations for a certain class or map them from one domain into another.

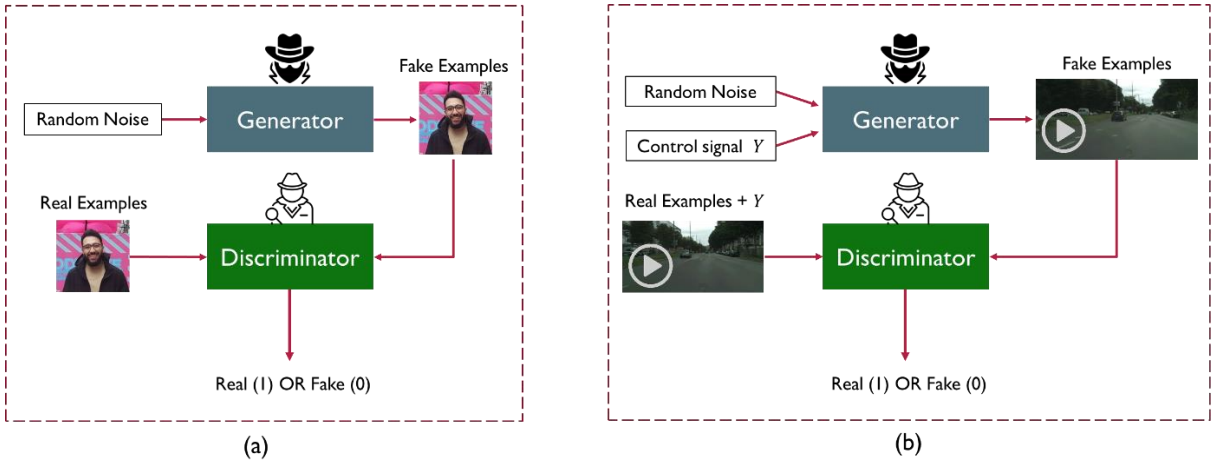


Figure 5.3 - A simplified schematic of (a) unconditional GAN and (b) conditional GAN.

Note: Both unconditional and conditional GANs can be applied to several data types such as images, video, text, etc.

The conditional GANs (cGANs) are used for image-to-image translation, where an image from a certain domain is mapped into another image in different domain (Park et al., 2019). Image translation can be done using cGANs in a supervised or unsupervised way. As a supervised image translation, the pix2pix architecture is proposed in (Isola et al., 2017), where the PatchGAN is used to discriminate each local batch of the image instead of the whole image. Another technique that tries to synthesize multiple outputs using the same input is proposed in (J.-Y. Zhu, Zhang, et al., 2017). Other techniques were proposed in (X. Liu et al., 2019; Tang et al., 2020; T.-C. Wang et al., 2018; Zheng et al., 2020) to improve the quality of these supervised image translation approaches. As an unsupervised technique, the CycleGAN is proposed in (J.-Y. Zhu, Park, et al., 2017) by adding a cycle consistency loss to enforce an image to be translated from one domain to another domain and translated back into the original domain. Unsupervised video generation techniques are discussed in (Srivastava et al., 2015; Vondrick et al., 2016), however none of them considered generating video conditioned on another video. This was tackled in (Bashkirova et al., 2018) where the CycleGAN is adapted to 3D-CycleGAN using 3D convolutional layer. The 3D-CycleGAN is depicted in Figure 5.4 (adopted from (Bashkirova et al., 2018)).

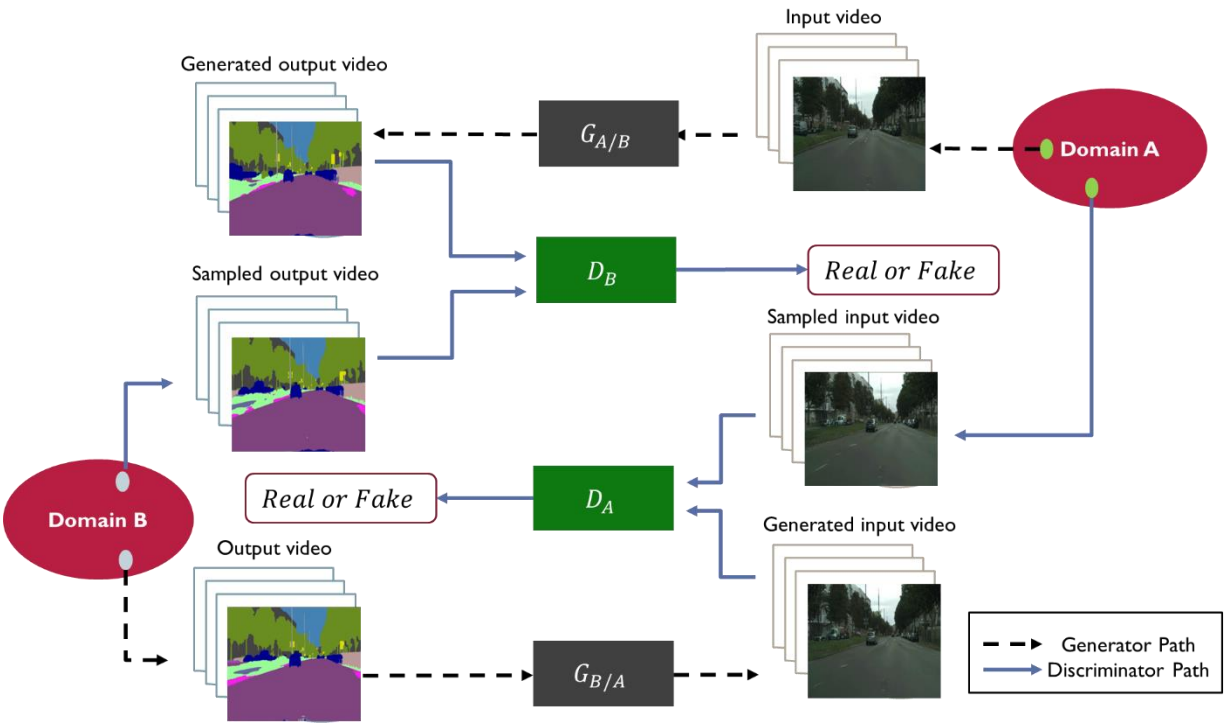


Figure 5.4 - A schematic diagram of the 3D-CycleGAN to translate camera videos into segmented videos (Adopted from (Bashkirova et al., 2018))

The main idea of the cycle consistency can be formulated mathematically in Eq. 3. The objective of the model is to minimize the adversarial loss (L_{GAN}) of the two generators and discriminators shown in Figure 5.4 and the cycle consistency (L_{cyc}) loss for observations X in domain A and observations Y in domain B as defined in Eq. (4), (5), (6) & (7). See (Bashkirova et al., 2018) for more details.

$$G_{A/B}\left(G_{B/A}(x)\right) \approx x \qquad (3)$$

$$L_{GAN}\left(D_B, G_{A/B}, X, Y\right) = \mathrm{E}_{y\sim p_B} \log\left(D_B(y)\right) + \mathrm{E}_{x\sim p_A} \log\left(1 - D_B\left(G_{A/B}(x)\right)\right) \qquad (4)$$

$$L_{GAN}\left(D_A, G_{B/A}, Y, X\right) = \mathrm{E}_{x\sim p_A} \log\left(D_A(y)\right) + \mathrm{E}_{y\sim p_B} \log\left(1 - D_A\left(G_{B/A}(x)\right)\right) \qquad (5)$$

$$L_{cyc}(G_{A/B}, G_{B/A}) = \mathrm{E}_{x\sim p_A}\left(\left\|G_{A/B}\left(G_{B/A}(x)\right) - x\right\|_1\right) + \mathrm{E}_{y\sim p_B}\left(\left\|G_{B/A}\left(G_{A/B}(y)\right) - y\right\|_1\right) \qquad (6)$$

$$\mathcal{L}\left(G_{A/B}, G_{B/A}, D_A, D_B\right) = L_{GAN}\left(D_B, G_{A/B}, X, Y\right) + L_{GAN}\left(D_A, G_{B/A}, Y, X\right) + L_{cyc}(G_{A/B}, G_{B/A}) \qquad (7)$$

The generator $G_{A/B}$ aims to translating videos from domain A to domain B, while the generator $G_{B/A}$ translates the videos from domain B to domain A. The discriminators D_B and D_A can figure out if the translated video fake or real compared to the videos sampled from domain B and domain A, respectively.

Given the successful application of 3D-CycleGAN in unsupervised video-to-video translation, this paper proposes combining the 3D-CycleGAN and polygon generation to solve the problem of time-series prediction. The details of the proposed method are presented in the next section.

5.4 Proposed Method for Time-Series Prediction

The proposed method is comprised of two phases; training and testing, as shown in Figure 5.5. The training phase results in a trained generator using the video-to-video translation technique. The generator maps the polygon streams (videos) representing the time-series inputs into other polygon streams representing the outputs (KPIs). In the testing phase, the trained generator translates the input polygon streams (that have not seen before) into other streams representing the predicted outputs. The two phases are illustrated in details in the following subsections.

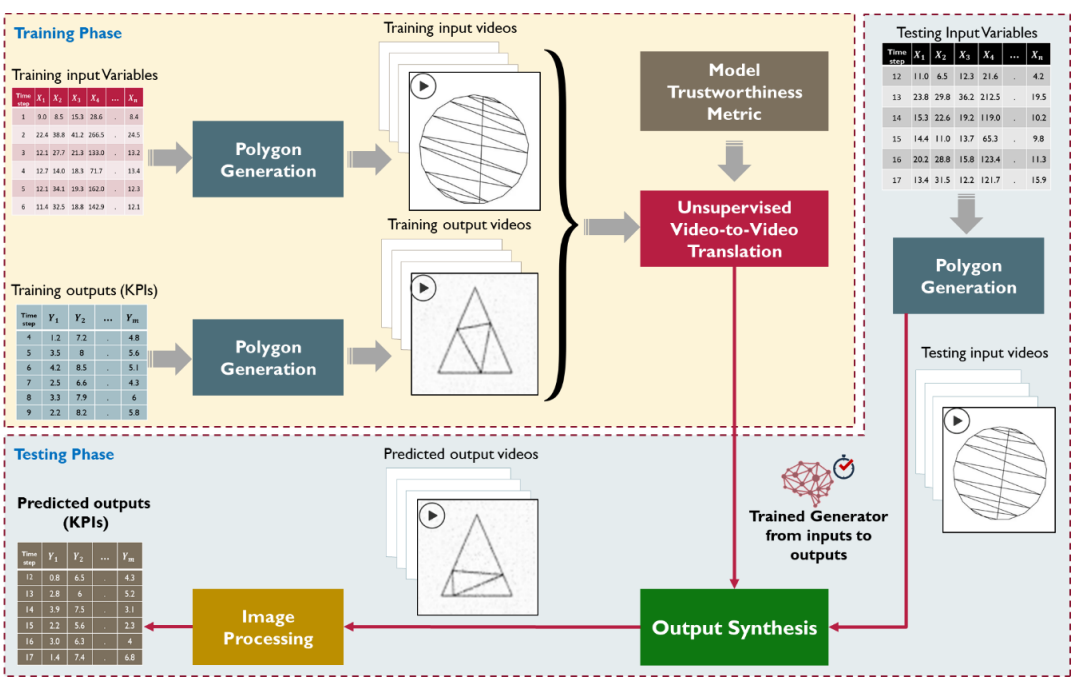


Figure 5.5 - A schematic diagram of the proposed method

5.4.1 Training Phase: Unsupervised Video-to-Video Translation

Our proposed method is targeting time-series numeric data with several inputs and outputs. It works in an unsupervised way, where the input variables do not need to be paired with the corresponding outputs. The purpose of this phase is trying to approximate the true distribution of each of the time-series inputs and outputs. As shown in the schematic diagram of the proposed method (Figure 5.5), the training time-series data is composed of n numerical input variables $(X_1, X_2, \dots, X_n)$ and m numerical outputs $(Y_1, Y_2, \dots, Y_m)$. The first step is applying the polygon generation technique for each of the inputs and outputs separately. This results in streams of polygon images (polygon videos) that represent each of the inputs and outputs, as illustrated in section 2.2. These streams represent all interrelationships between each of the inputs and outputs in addition to reflecting their changes over time.

Figure 5.6 shows how a polygon changes over time for a data of three outputs (KPIs). For the sake of illustration, the KPIs shown in the figure change monotonically, however, this method can deal with any type of data with changing distribution. As shown in the figure, the movement of the point along the polygon side indicates whether its value increases or decreases over time. In our proposed method, we deal with these polygon videos to capture the true distribution of the input variables and outputs and how to match between them using video-to-video translation technique that is illustrated in what follows.

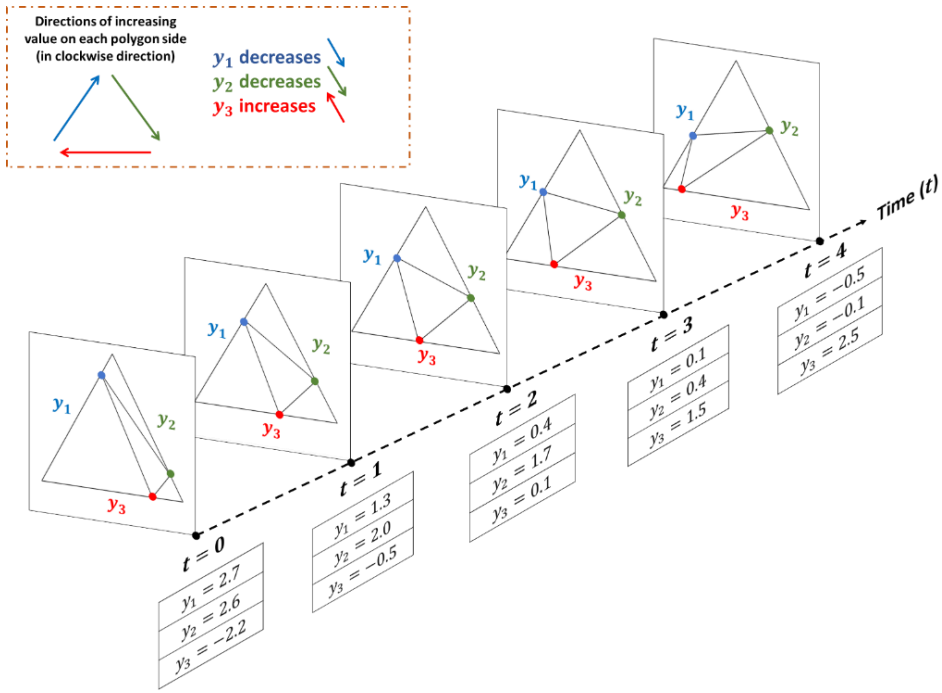


Figure 5.6 - The change of polygon images over time in the form of a polygon stream through an example of three outputs

The unsupervised video-to-video translation technique is fed with both the polygon streams of inputs and outputs. The generator is trained to map the input distribution into the output one. In order to optimize the performance of the video-to-video translation, a model trustworthiness metric is used for ensuring the quality of the synthesized videos compared to the original ones. There are several common metrics for measuring the quality of the video frames such as mean-squared error, peak signal-to-noise ratio, and universal image quality index (Z. Wang & Bovik, 2002). In this work, the universal image quality index proposed in (Z. Wang & Bovik, 2002) is adopted due to its effectiveness. It adequately compiles the similarity between two videos in terms of different aspects; the correlation, the average of the pixel values and contrast. The quality index is mathematically defined as the multiplication of three terms as defined in Eq. (8):

$$Q = \frac{4\sigma_{xy}\overline{x}\overline{y}}{\left(\sigma_x{}^2 + \sigma_y{}^2\right)\left((\overline{x})^2 + (\overline{y})^2\right)} = \frac{\sigma_{xy}}{\sigma_x\sigma_y} \cdot \frac{2\overline{x}\overline{y}}{(\overline{x})^2 + (\overline{y})^2} \cdot \frac{2\sigma_x\sigma_y}{\sigma_x{}^2 + \sigma_y{}^2} \qquad (8)$$

where,

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} x_i \,,$$

$$\overline{\boldsymbol{y}} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

$$\sigma_{\boldsymbol{x}}{}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{\boldsymbol{x}})^2 \;, \sigma_{\boldsymbol{y}}{}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \overline{\boldsymbol{y}})^2 \,, \sigma_{\boldsymbol{xy}} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{\boldsymbol{x}})(y_i - \overline{\boldsymbol{y}})$$

where $\boldsymbol{x} = \{x_i, i = 1,2, \dots, N\}$ represent the polygon stream of the outputs with N frames using polygon generation technique, while $\boldsymbol{y} = \{y_i, i = 1,2, \dots, N\}$ represent the synthesized polygon stream using our proposed method. The first term of Q ($\frac{\sigma_{xy}}{\sigma_x \sigma_y}$) in Eq. (8) represents the correlation between the frames of the two videos (ranges from -1 to 1), the second term ($\frac{2\overline{\boldsymbol{x}}\overline{\boldsymbol{y}}}{(\overline{\boldsymbol{x}})^2 + (\overline{\boldsymbol{y}})^2}$) represents how close the mean pixel values of the video frames are (ranges from 0 to 1) and the last term ($\frac{2\sigma_x \sigma_y}{\sigma_x{}^2 + \sigma_y{}^2}$) represents how close the video contrasts are (ranges from 0 to 1).

Accordingly, in the proposed methodology, the video-to-video translation model is tuned based on this quality index metric along with its validation performance to increase the model trustworthiness. Figure 5.7 illustrates the process of training the unsupervised video-to-video translation using the input and output polygon streams.

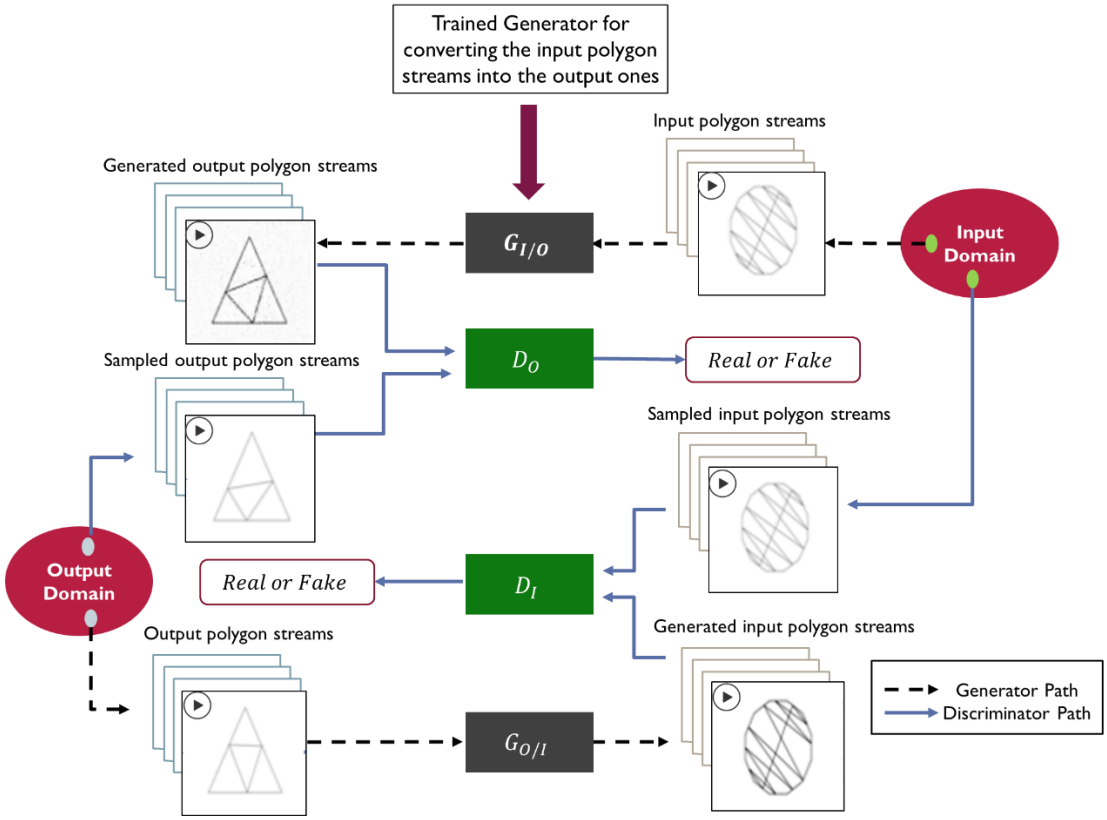


Figure 5.7 - Training of the 3D-CycleGAN model to obtain the generator $G_{I/O}$ that converts polygon streams representing time-series input variables into another set of polygon streams representing the time-series outputs

The 3D-CycleGAN architecture (Bashkirova et al., 2018) is used for this translation task, where $G_{I/O}$ maps the input polygon streams into the output ones, $G_{O/I}$ maps the output polygon streams into the input ones. The discriminators D_I and D_O differentiate between the real and fake input and output streams, respectively. The $G_{I/O}$ is the outcome of the 3D-CycleGAN training phase that is used later in the testing phase.

5.4.2 Testing Phase: Mapping Videos into Time-series Outputs

In the testing phase, the polygon generation technique is applied for the streaming testing data to generate a set of polygon streams representing the testing input variables. The trained generator $G_{I/O}$ is used for translating these polygon streams into another set of polygon streams representing the predicted outputs. In order to map the translated streams into predicted outputs, an

image processing procedure is applied to every frame in the polygon streams as depicted in Figure 5.8 that shows a square representing data with four outputs.

As shown in the figure, first, the corners of the polygon are obtained using the cMinMax algorithm proposed in (Chamzas et al., 2020). Based on these corners, the points on each side are determined, where they represent all possible output values. The pixel values in each frame of the polygon stream are binarized, then the point with the highest values of the surrounding pixels represents the standardized numerical value of its corresponding time-series output. Finally, the midpoints of each polygon side (representing the mean values of the outputs) and the points that represent the standardized output values are used to map the polygon streams back to numerical values.

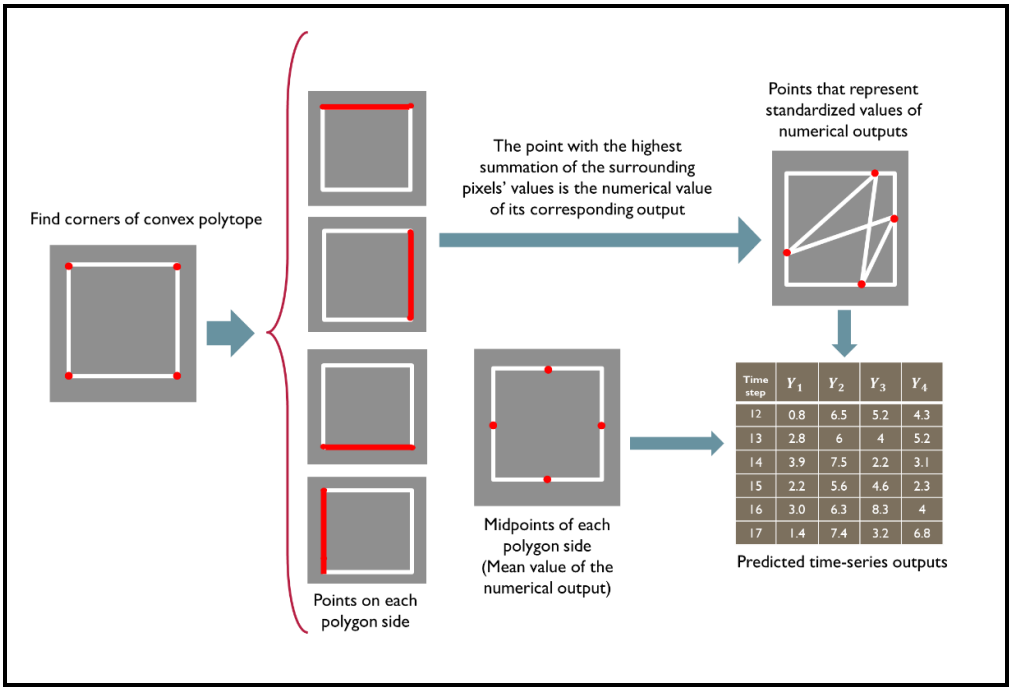


Figure 5.8 - A schematic diagram summarizes the image processing steps for mapping every frame in polygon streams into time-series outputs

5.5 Case Study: Concentrator in Heat Recovery Network (HRN)

The proposed method is validated based on a challenging dataset with complex data distribution collected from the concentrator; a major equipment in heat recovery network (HRN)

of a pulp and paper mill located in Canada. Details on this equipment, its operation, explanation of the KPIs and the dataset description are discussed in this section.

5.5.1 System Operation and KPIs

In the Kraft pulping process, weak black liquor (BL) is a by-product of wood chips cooking and pulp washing steps (Bajpai, 2018; Biermann, 1996). This weak BL is concentrated in multi effect evaporator and concentrators to increase its solid concentration before to feed the recovery boiler. The objectives are to recover the BL inorganics and to burn the organic components. The generated steam in the recovery boiler is used for power generation and for process heating. In order to improve the recovery boiler operation and efficiency, the black liquor solid concentration should be maximized. Typically, multiple-effect evaporation system is used to increase the dissolved solid concentration of the weak BL from 15-18% to about 55% and then concentrators are used to concentrate the BL to about 65-70% before entering the recovery boiler.

Figure 5.9 shows a simplified schematic of the concentrator equipment with the monitored KPIs. The main components of the equipment are a heat exchanger and a flash chamber where vapor is formed and separated from liquid phase (Soualhi et al., 2021). The fresh steam is then used to heat the black liquor in the heat exchanger. More details about the operation of the concentrator are found in (Bajpai, 2018).

As shown in Figure 5.9, the first KPI is the evaporated water flow, the second is the concentrator efficiency where it is calculated as the evaporated water divided by the fresh steam consumed. The third KPI is the fouling index which is an important indicator of the decrease in the overall heat transfer from steam to black liquor. Since the heat exchange rate depends on the temperature difference between the steam and the black liquor, the fouling index is defined using Eq. (9) (Ardsomang et al., 2013).

$$Fouling\ index = \frac{temperature\ of\ steam - temperature\ of\ heavy\ black\ liquor}{evaporated\ water} \tag{9}$$

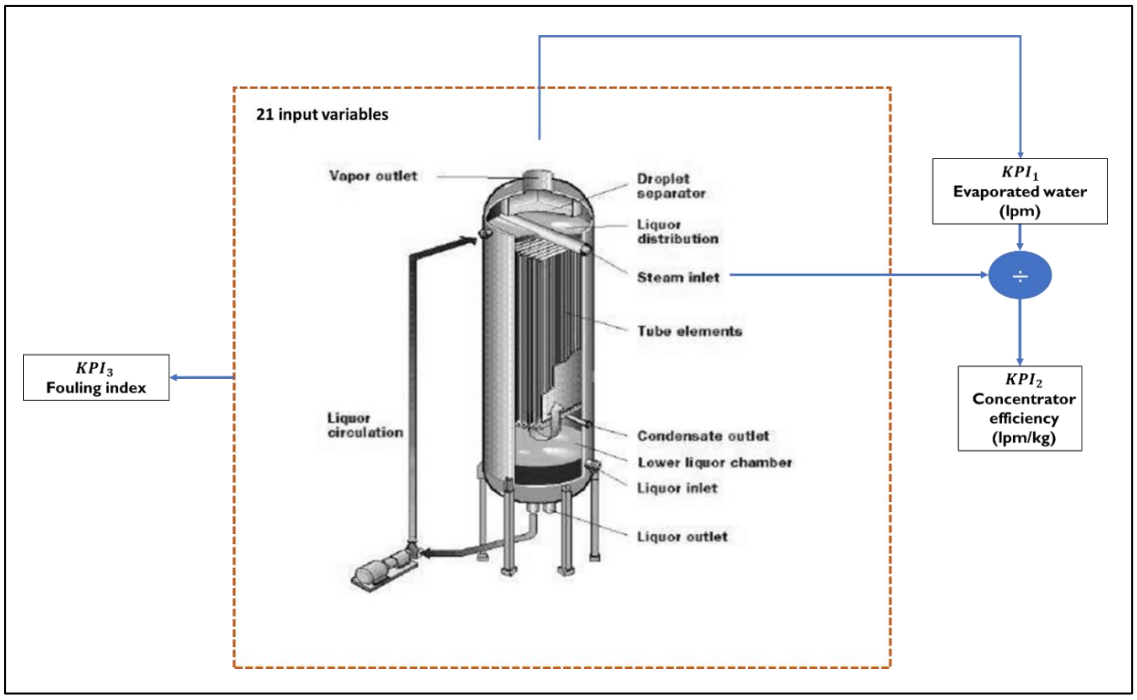


Figure 5.9 - A schematic diagram of the concentrator equipment in the HRN

5.5.2 Experimental Setup

The concentrator dataset composed of 37440 observations collected from the mill historian of 390 days with sampling time of 15 mins. It comprises a total of 42 cycles and includes a total of 21 manipulated and measured variables; selected by the process expert to represent the highly dynamic behavior of the concentrator operation. Examples of these variables are shown in Table 5.3. Data cleaning and preparation through removal of outliers and non-representative data were done by the process expert using the software EXPLORE (Amazouz, 2015).

Table 5.3 - Examples of manipulated, measured variables and KPIs for the concentrator equipment

Manipulated and Measured variables	KPIs
▪ Liquor flow to concentrator (lpm) ▪ Temperature of liquor from concentrator (°C) ▪ Temperature of vapor from concentrator (°C) ▪ Pressure of fresh steam to concentrator (kPa) ▪ Fresh steam flow to concentrator (kg/h) ▪ Temperature differential steam/liquor concentrator (°C)	▪ KPI_1: Evaporated water (lpm) ▪ KPI_2: Concentrator efficiency (evaporated water (lpm) / steam (kg)) ▪ KPI_3: Fouling index

In this work, we used the 3D-CycleGAN proposed in (Bashkirova et al., 2018) with two generators and two discriminators. The generator architecture (Johnson et al., 2016) is illustrated in Figure 5.10, where it is composed of two 3D convolutional blocks followed by nine residual blocks and two 3D deconvolutional blocks for upsampling. Each convolutional block is composed of a 3D convolutional layer (Ji et al., 2012), batch normalization layer and rectified linear-unit (ReLU) as an activation layer. Each deconvolutional block is composed of 3D deconvolutional layer, batch normalization layer and ReLU layer. The residual block is composed of five layers ordered as follows: 3D convolutional, batch normalization, ReLU, 3D convolutional and batch normalization. The output of each residual block is added to that of the previous block as input to the next residual block as shown in Figure 5.10. Since deep neural networks often suffer from the vanishing gradient and performance degradation, the residual block is used to mitigate this effect (K. He et al., 2016).

The discriminator in the 3D-CycleGAN is the PatchGAN architecture introduced in (Demir & Unal, 2018). PatchGAN divides each video into 70 * 70 * *h* patches, where *h* is the video depth. This architecture predicts the classification probability in a form of a 3D matrix where every value refers to the probability for the corresponding patch in the video frame. For the sake of simplification, we use a single image as a video frame in Figure 5.11 to illustrate the operation of PatchGAN. The 3D matrix with all entries of ones refers to a real video, while the one with zeros refers to a fake one.

The TensorFlow (Mart\’\in Abadi et al., 2016) with Python 3.7 was used to implement, train and test the proposed method (PG + 3D-CycleGAN) and other baseline algorithms on a high performance computing (HPC) infrastructure in Natural Resources Canada with following specifications: Intel® Xeon® Gold 6140 CPU @2.3 GHz, 1 TB of RAM + 4 GPUs (NVIDIA Tesla V100).

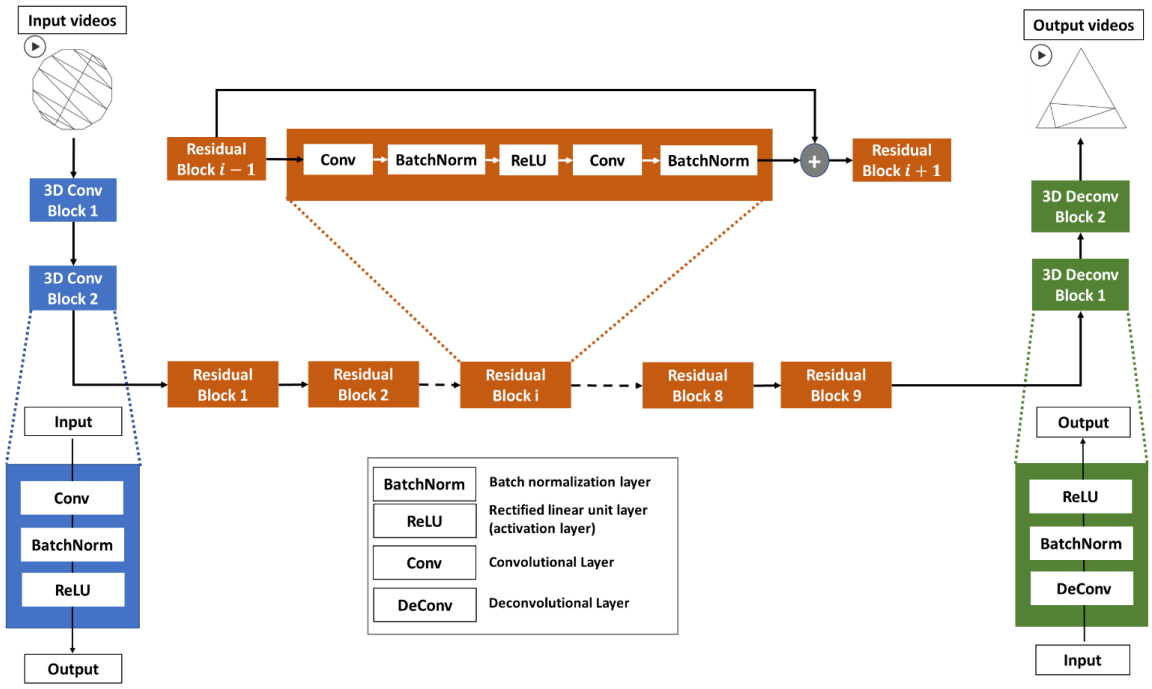


Figure 5.10 - The generator architecture in the 3D-CycleGAN

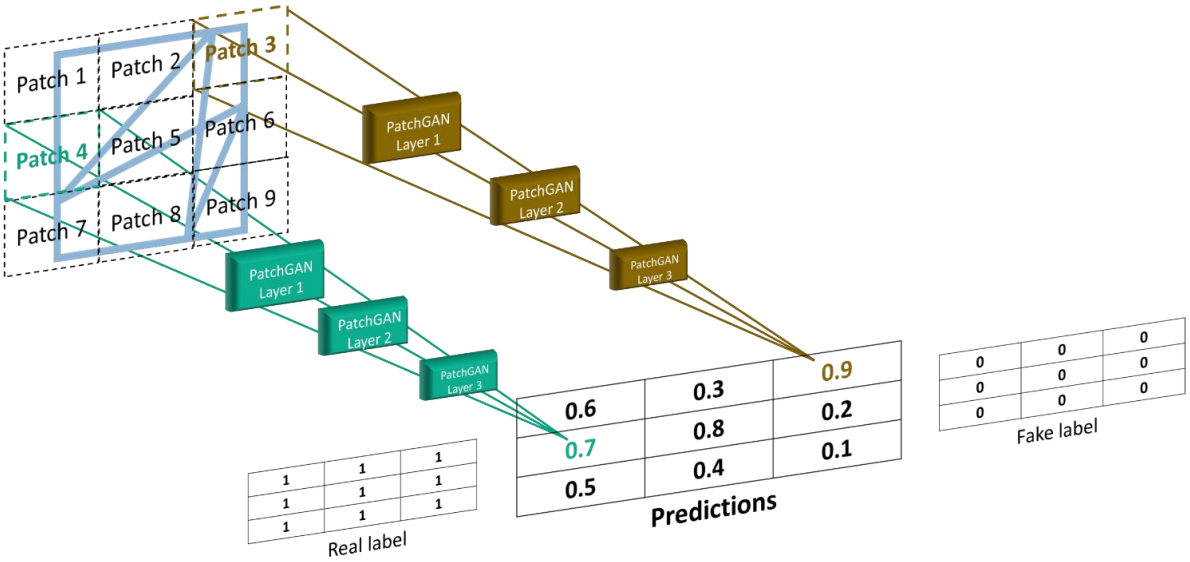


Figure 5.11 - PatchGAN: The discriminator of the 3D-CycleGAN

5.6 Results, Discussion & Future Work

The performance of the proposed method is compared with other baseline time-series predictors; recurrent neural network (RNN), long-short term memory (LSTM), one-dimensional convolutional neural network (1D-CNN). These time-series predictors have been used extensively in the literature and in practice (Dong et al., 2017; Lanzetti et al., 2019; Soualhi et al., 2021; Zagrebina et al., 2019). The hyperparameters for each baseline predictor are optimized using grid search for the sake of the best performance. The tuning of our proposed method was based on the model trustworthiness metric (universal image quality index) that mentioned in Section 3. The goal is to maximize the value of this index to ensure that the synthesized videos are structurally close to the desired videos that represent the time-series outputs.

It is worth mentioning that, during the process of training the 3D-CycleGAN using polygon streams, there was a problem of generating the same output video for multiple input videos. This phenomenon is common in GAN training, and it is called “mode-collapse” (Durall et al., 2020). The real distribution of the time-series outputs in most industrial processes is a multi-modal distribution due to the highly nonlinear dynamic nature of these systems. The generator sometimes can fool the discriminator through synthesizing fake videos with only one mode, while the discriminator cannot figure out if it is fake or not. There are some hacks to overcome this problem such as the normalization of input videos: the grey-scale videos can be normalized to have the values in interval [-1,1] or [0,1] instead of [0,255]. Another hack is to decrease the learning rate of the optimizer used in generator and discriminator. After we followed these two hacks in this case study, the 3D-CycleGAN was able to synthesize multi-modal output videos.

Table 5.4 shows the range of the hyperparameters for grid search in each time-series predictor. A random seed is fixed for the reproducibility of the results. Both R-squared values (R^2) and root mean square error ($RMSE$) are used as metrics to compare the performance of all predictors (Bustillo, Reis, et al., 2020; Kasuya, 2019).

Table 5.4 - Range of hyperparameters of each time-series predictor using the concentrator equipment

Algorithm	**Hyperparameters**
Proposed Method (PG + 3DCycleGAN)	# filters in conv layer = [4,64], filter size = (2,3) # epochs = [30,150]
RNN & LSTM	# units = [15,40], activation function = {sigmoid, ReLU, tanh}, recurrent activation = {sigmoid, ReLU, tanh}, dropout = [0,1]
1D-CNN	# filters = [4,32], batch size=[4,32], # epochs = [10,50], activation function = {sigmoid, ReLU}, kernel size = [2.8]

Moreover, a penalty function is used as a validation criterion for each time-series predictor, taking into consideration the underestimation and overestimation of the three KPIs. The penalty function $L^j(t_k)$ of KPI_j , $j = 1,2,3$ at instance t_k is defined in Eq. (10).

$$L^j(t_k) = \begin{cases} \alpha_j u(j)\left(KPI_j(t_k) - \widehat{KPI_j}(t_k)\right), & \widehat{KPI_j}(t_k)u(j) < KPI_j(t_k)u(j) \\ 0, & \widehat{KPI_j}(t_k) = KPI_j(t_k) \\ \beta_j u(j)\left(\widehat{KPI_j}(t_k) - KPI_j(t_k)\right), & \widehat{KPI_j}(t_k)u(j) > KPI_j(t_k)u(j) \end{cases} \quad (10)$$

where, α_j and β_j are the underestimation and overestimation parameters for each KPI_j, respectively and $\widehat{KPI_j}(t_k)$ and $KPI_j(t_k)$ are the predicted and true values of KPI_j at instance t_k, respectively. The term $u(j)$ has a value of 1 or -1 depending on the predicted KPI_j. $u(1) = u(2) = 1$ (the overestimation of KPI_1 and KPI_2 is penalized more than the underestimation), while $u(3) = -1$ (the underestimation of KPI_3 is penalized more than the overestimation). These parameters were assigned according to the energy efficiency importance of each KPI as confirmed by the process expert. Accrodingly, in this work, the values of α_j and β_j are assigned the values shown in Table 5.5.

Table 5.5 - Underestimation (α_j) and overestimation (β_j) parameters for each KPI_j in the penalty function as defined by the process expert

	$j = 1$	$j = 2$	$j = 3$
α_j	0.1	0.1	0.1
β_j	0.15	0.3	0.2

The average penalty score for each KPI is calculated as shown in Eq. (11).

$$L_{AP}^{j} = \frac{1}{N}\sum_{k=1}^{N} L^{j}(t_k), \qquad (11)$$

where N is the total number of time steps. The total average penalty score for each time-series regression model is calculated as shown in Eq. (12).

$$L_{model} = \frac{1}{3}\sum_{j=1}^{3} L_{AP}^{j}, \qquad (12)$$

5.6.1 Results

As previously mentioned in Section 4, three KPIs are used for this case study; the evaporated water flow, the concentrator efficiency, and the fouling index. Based on the polygon generation technique and the number of input variables in concentrator data (21 variables), each observation has 10 different Hamiltonian cycle connections (10 polygon streams). These streams represent all interrelationships between the input variables and their changes over time, while there is only one Hamiltonian cycle connection (one polygon stream) for the 3 outputs. It is worth mentioning that the choice of the number of frames per video is limited by the memory of a single GPU unit. Therefore, the number of frames is set to 30 per video using the HPC infrastructure mentioned previously.

The R-squared and $RMSE$ values of all predictors are listed in Table 5.6 and the total average penalty incurred from the erroneous prediction of each predictor is listed in Table 5.7 for the concentrator case study. As shown in Table 5.6 and Table 5.7, the proposed method (PG + 3D-

CycleGAN) has achieved the highest R-squared value and lowest $RMSE$ on each of the KPIs and the lowest total average penalty score. The numbers in bold indicate the best results obtained. It can be observed from the results that there is a significant improvement of the prediction of the concentrator efficiency (KPI_2).

Figure 5.12 visualizes the performance of the time-series prediction of the proposed method in comparison with the true values and every baseline prediction model. It shows the predicted values of the concentrator efficiency over time.

These results are validated by the process expert and shown to be useful for the mill operator. This helped better monitor such highly dynamic operation and mitigating the economic losses and environmental impacts resulting from the past inaccurate prediction over time. Besides, this helped the operator prescribe the proper actions in real-time.

Table 5.6 - R-squared and root mean square error values of each algorithm in the concentrator dataset

Algorithm	KPI_1		KPI_2		KPI_3	
	R^2	$RMSE$	R^2	$RMSE$	R^2	$RMSE$
Proposed Method (PG+3D CycleGAN)	**0.75**	**14.63**	**0.8**	**0.014**	**0.95**	**0.029**
RNN	0.64	24.6	0.63	0.033	0.79	0.062
LSTM	0.69	18.44	0.63	0.026	0.93	0.038
1D-CNN	0.67	19.58	0.62	0.029	0.92	0.037

Table 5.7 - Total average penalty scores for each algorithm in the concentrator dataset

Algorithm	PG+3D	RNN	LSTM	1D-CNN
Total average penalty score	**0.0261**	0.0315	0.0294	0.0309

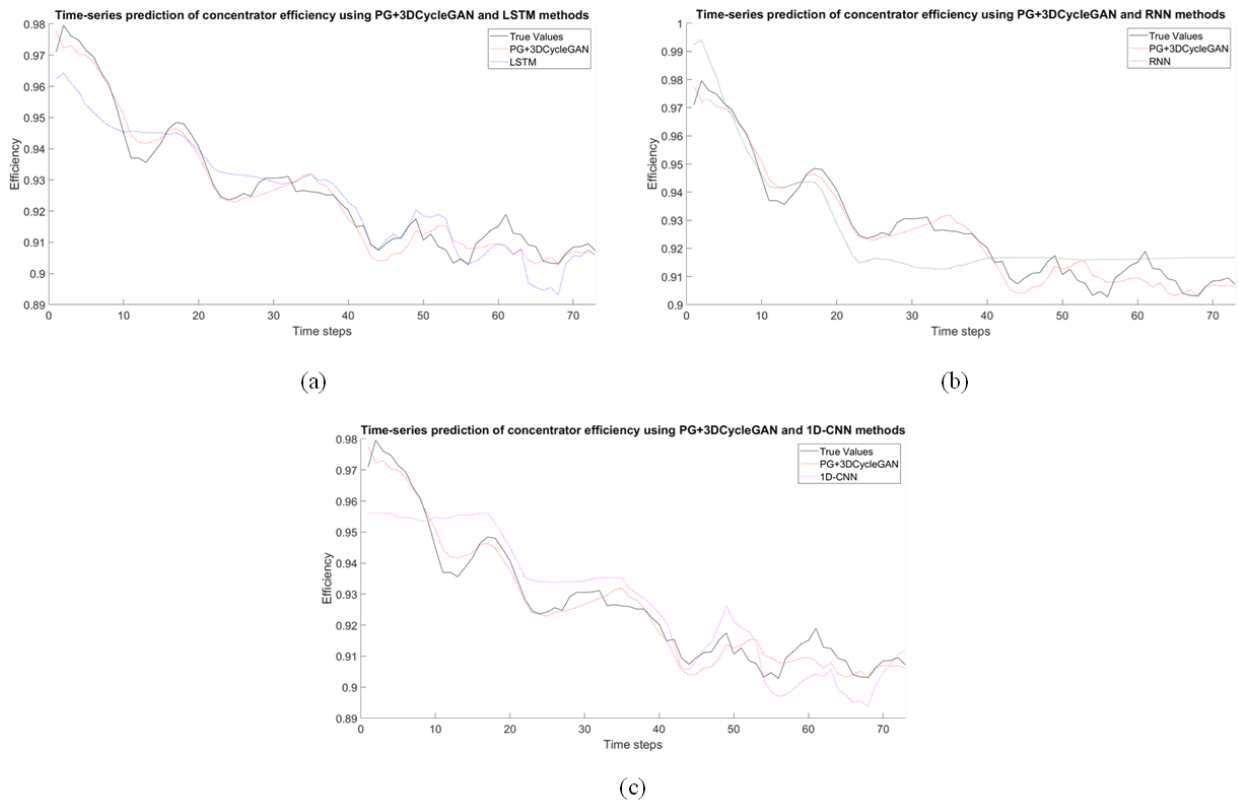


Figure 5.12 - Prediction of KPI_2 (Concentrator efficiency) using the proposed method (PG + 3DCycleGAN) and other prediction models (a) LSTM (b) RNN (c) 1D-CNN.

5.6.2 Discussion and Future Work

To sum up, the historical data is collected from the industrial plant through multiple sensors, accordingly, this time-series data represents the fuel of our proposed method. The proposed method was able to solve the problem of distribution matching of input variables and the outputs in this challenging industrial dataset. This is attributed to the following facts. Both of the time-series input variables and KPIs are converted into polygon videos to train the unsupervised video-to-video translation architecture (3D- CycleGAN) in the training phase. It is worth mentioning that the testing phase is neither computationally expensive nor time consuming. A fast and accurate testing phase is a desired characteristic from an industrial perspective. This testing phase in the proposed method includes preparing polygon videos compiling the data stream of input variables, then testing the trained generator to map these videos to other set of videos that represent the predicted

KPIs, finally the numerical KPI values are recovered easily using the image procedure mentioned previously.

The polygon generation technique used in the proposed method was able to express all interrelationships between the input variables and the KPIs based on the Hamiltonian cycles. This efficiently represents the numerical data and leverage its quality as one of data-centric AI goals. Moreover, to ensure the consistency of the structure of the translated polygon streams, a model trustworthiness metric is used to tune the architecture of the 3D-CycleGAN to maximize the universal image quality index for each video frame.

Besides the above strengths, we make use of the breakthrough of the DL and its impressive performance in the computer vision problems especially in generative modeling. DL can capture the high dynamic behavior of the equipment with minimal intervention of the process expert. It is worth mentioning that the proposed method is an end-to-end learning process that does not need the effort of manual feature engineering that is done by the process expert in a tedious manner. In addition, it saves the expert's effort for the labeling process as the method works in an unsupervised way. Besides, the availability of advanced IT infrastructure in modern industries makes the proposed method feasible especially in the training phase, in which training a deep architecture using massive amount of data is needed.

From the practical point of view, all these merits can guarantee the operationally deployable implementation of the proposed method in industrial settings. Other challenging industrial datasets will be collected from a number of non-linear and dynamic processes in the future for further testing of our proposed method. The resolution of the polygon videos will be further investigated as it may have a significant effect on the method performance.

The proposed method opens the door for industrial data fusion in terms of merging numerical data, images, and videos. This can help efficiently exploit the available heterogeneous data to maximize the global value of isolated data silos provide the operator with valuable knowledge. One of our future research directions is to develop a platform that can integrate and process different types of data in terms of structure and format. The platform will consist of an ensemble of DL models each used to process specific data type. Moreover, the visualization of the KPIs changes and the input variables in the form of representational videos can play a key role in interpretation

of DL models. The final goal is to provide the end users with accurate and transparent knowledgebase with explainable rules.

5.7 Conclusion

This paper proposes a novel time-series prediction method that is based on two main building blocks; polygon generation and unsupervised video-to-video translation. The time-series numerical observations are converted into a set of polygon streams (videos) using polygon generation. The unsupervised video-to-video translation is used to map the videos representing the input variables into others representing the outputs as a distribution matching problem. The proposed method takes advantage of the unprecedented performance of generative deep learning (DL) modeling to capture the dynamic and complex data distribution, which is hard to determine in highly non-linear process industries. The method is tested successfully using a challenging industrial dataset collected from a concentrator equipment in a thermomechanical pulp mill located in Canada. The results show that the proposed method outperformed other comparable time-series DL predictors in terms of KPIs' prediction accuracy. As the proposed method has the advantage of working in an unsupervised way, it saves the effort of data labeling process done by the process expert. The trustworthiness of the video translation model is maximized using an index to maintain a consistent structure of the translated polygon streams. Moreover, the interpretability of DL models is one of our current research directions. Besides, using polygon generation as a data representation technique opens the door for fusion of heterogeneous data types in various industrial processes.

5.8 References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others. (2016). Tensorflow: A system for large-scale machine learning. *12th $\{$USENIX$\}$ Symposium on Operating Systems Design and Implementation ($\{$OSDI$\}$ 16)*, 265–283.

Alpaydin, E. (2010). *Introduction to machine learning, 2nd edn. Adaptive computation and machine learning*. The MIT Press (February 2010).

Amazouz, M. (2015). *Improving Process Operation Using the Power of Advanced Data Analysis*. Retrieved from

https://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/canmetenergy/files/pubs/EXPLORE-brochure_ EN.pdf

Andersson, E., & Thollander, P. (2019). Key performance indicators for energy management in the Swedish pulp and paper industry. *Energy Strategy Reviews*, *24*(December 2018), 229–235. https://doi.org/10.1016/j.esr.2019.03.004

Andrew Ng Launches A Campaign For Data-Centric AI. (n.d.). Retrieved from https://www.forbes.com/sites/gilpress/2021/06/16/andrew-ng-launches-a-campaign-for-data-centric-ai/?sh=5dea92f374f5

Ardsomang, T., Hines, J. W., & Upadhyaya, B. R. (2013). Heat exchanger fouling and estimation of remaining useful life. *Annual Conference of the PHM Society*, *5*(1).

Bai, Y., Xie, J., Wang, D., Zhang, W., & Li, C. (2021). A manufacturing quality prediction model based on AdaBoost-LSTM with rough knowledge. *Computers and Industrial Engineering*, *155*(January), 107227. https://doi.org/10.1016/j.cie.2021.107227

Bajpai, P. (2018). Brief Description of the Pulp and Papermaking Process. In *Biotechnology for pulp and paper processing* (pp. 9–26). Springer.

Bashkirova, D., Usman, B., & Saenko, K. (2018). *Unsupervised Video-to-Video Translation*. (Nips). Retrieved from http://arxiv.org/abs/1806.03698

Biermann, C. J. (1996). *Handbook of pulping and papermaking*. Elsevier.

Borovykh, A., Bohte, S., & Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10614 LNCS*, 729–730.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley \& Sons.

Brown, R. G., & Meyer, R. F. (1961). The fundamental theorem of exponential smoothing. *Operations Research*, *9*(5), 673–685.

Bustillo, A., Reis, R., Machado, A. R., & Pimenov, D. Y. (2020). Improving the accuracy of

machine-learning models with data from machine test repetitions. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01661-3

Chamzas, D., Chamzas, C., & Moustakas, K. (2020). cMinMax: A fast algorithm to find the corners of an N-dimensional convex polytope. *ArXiv:2011.14035v2*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Data-Centric AI Competition. (n.d.). Retrieved from https://https-deeplearning-ai.github.io/data-centric-comp/

Demir, U., & Unal, G. (2018). Patch-based image inpainting with generative adversarial networks. *ArXiv:1803.07422v1*.

Dong, D., Li, X.-Y., & Sun, F.-Q. (2017). Life prediction of jet engines based on LSTM-recurrent neural networks. *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 1–6.

Durall, R., Chatzimichailidis, A., Labus, P., & Keuper, J. (2020). Combating Mode Collapse in GAN training: An Empirical Analysis using Hessian Eigenvalues. *ArXiv Preprint ArXiv:2012.09673*.

Elhefnawy, M., Ragab, A., & Ouali, M.-S. (2021). Fault classification in the process industry using polygon generation and deep learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-021-01742-x

Environment challenges | Climate Action. (n.d.). Retrieved from https://ec.europa.eu/clima/policies/adaptation/how/challenges_en#tab-0-1

Essien, A., & Giannetti, C. (2020). A Deep Learning Model for Smart Manufacturing Using Convolutional LSTM Neural Network Autoencoders. *IEEE Transactions on Industrial Informatics*, *16*(9), 6069–6078. https://doi.org/10.1109/TII.2020.2967556

Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83–85.

Gamboa, J. C. B. (2017). *Deep Learning for Time-Series Analysis*. Retrieved from http://arxiv.org/abs/1701.01887

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Bengio, Y. (2014). Generative Adversarial Networks. *Commun. ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Han, Z., Zhao, J., Leung, H., Ma, K. F., & Wang, W. (2021). A Review of Deep Learning Models for Time Series Prediction. *IEEE Sensors Journal*, *21*(6), 7833–7848. https://doi.org/10.1109/JSEN.2019.2923982

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hoermann, S., Bach, M., & Dietmayer, K. (2018). Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2056–2063.

Huang, J. T., Li, J., & Gong, Y. (2015). An analysis of convolutional neural networks for speech recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2015-August*, 4989–4993. https://doi.org/10.1109/ICASSP.2015.7178920

Hurley, C. B., & Oldford, R. W. (2010). Pairwise display of high-dimensional information via eulerian tours and hamiltonian decompositions. *Journal of Computational and Graphical Statistics*, *19*(4), 861–886.

Ian Goodfellow Yoshua Bengio, A. C. (2017). The Deep Learning Book. *MIT Press*, *521*(7553), 785. https://doi.org/10.1016/B978-0-12-391420-0.09987-X

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional

adversarial networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua*, 5967–5976. https://doi.org/10.1109/CVPR.2017.632

Jebara, T. (2012). *Machine learning: discriminative and generative* (Vol. 755). Springer Science & Business Media.

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221–231.

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9906 LNCS*, 694–711. https://doi.org/10.1007/978-3-319-46475-6_43

Kasuya, E. (2019). On the use of r and r squared in correlation and regression. *Ecological Research*, *34*(1), 235–236. https://doi.org/10.1111/1440-1703.1011

Kedem, B., & Fokianos, K. (2005). *Regression models for time series analysis* (Vol. 488). John Wiley \& Sons.

Lanzetti, N., Lian, Y. Z., Cortinovis, A., Dominguez, L., Mercangöz, M., & Jones, C. (2019). Recurrent neural network based MPC for process industries. *2019 18th European Control Conference (ECC)*, 1005–1010.

Lapedes, A., & Farber, R. (1987). *Nonlinear signal processing using neural networks: Prediction and system modelling*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

LeCun, Y., Bengio, Y., & others. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(10), 1995.

Liu, M. Y., Huang, X., Yu, J., Wang, T. C., & Mallya, A. (2021). Generative Adversarial Networks for Image and Video Synthesis: Algorithms and Applications. *Proceedings of the IEEE*, *109*(5), 839–862. https://doi.org/10.1109/JPROC.2021.3049196

Liu, X., Yin, G., Shao, J., Wang, X., & Li, H. (2019). Learning to predict layout-to-image

conditional convolutions for semantic image synthesis. *ArXiv Preprint ArXiv:1910.06809*.

Lv, F., Wen, C., Bao, Z., & Liu, M. (2016). Fault diagnosis based on deep learning. *Proceedings of the American Control Conference*, *2016-July*(2), 6851–6856. https://doi.org/10.1109/ACC.2016.7526751

Mirza, M., & Osindero, S. (2014). *Conditional Generative Adversarial Nets*. 1–7. Retrieved from http://arxiv.org/abs/1411.1784

National Inventory Report. (2019). *GREENHOUSE GAS SOURCES AND SINKS IN CANADA CANADA'S SUBMISSION TO THE UNITED NATIONS FRAMEWORK CONVENTION ON CLIMATE CHANGE Executive Summary*.

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems*, 841–848.

Pan, R. (2010). H olt--W inters Exponential Smoothing. *Wiley Encyclopedia of Operations Research and Management Science*.

Park, T., Liu, M.-Y., Wang, T.-C., & Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2337–2346.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., … Waldman-Brown, A. (2019). Tackling climate change with machine learning. *ArXiv Preprint ArXiv:1906.05433*.

Schat, E., van de Schoot, R., Kouw, W. M., Veen, D., & Mendrik, A. M. (2020). The data representativeness criterion: Predicting the performance of supervised classification based on data set similarity. *PLoS ONE*, *15*(8 August), 1–16. https://doi.org/10.1371/journal.pone.0237009

Soualhi, M., El Koujok, M., Nguyen, K. T. P., Medjaher, K., Ragab, A., Ghezzaz, H., … Ouali, M.-S. (2021). Adaptive prognostics in a controlled energy conversion process based on long- and short-term predictors. *Applied Energy*, *283*, 116049.

Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015). Unsupervised learning of video representations using lstms. *International Conference on Machine Learning*, 843–852.

Tang, H., Qi, X., Xu, D., Torr, P. H. S., & Sebe, N. (2020). Edge guided GANs with semantic preserving for semantic image synthesis. *ArXiv Preprint ArXiv:2003.13898*.

Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Generating videos with scene dynamics. *Advances in Neural Information Processing Systems*, *29*, 613–621.

Wang, J., Yan, J., Li, C., Gao, R. X., & Zhao, R. (2019). Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction. *Computers in Industry*, *111*, 1–14. https://doi.org/10.1016/j.compind.2019.06.001

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8798–8807.

Wang, Z., & Bovik, A. C. (2002). A universal image quality index. *IEEE Signal Processing Letters*, *9*(3), 81–84.

Wegman, E. J. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, *85*(411), 664–675.

Wu, A. (n.d.). *A Chat with Andrew on MLOps: From Model-Centric to Data-Centric AI. 2021*.

Yuan, X., Li, L., Shardt, Y. A. W., Wang, Y., & Yang, C. (2021). Deep Learning with Spatiotemporal Attention-Based LSTM for Industrial Soft Sensor Model Development. *IEEE Transactions on Industrial Electronics*, *68*(5), 4404–4414. https://doi.org/10.1109/TIE.2020.2984443

Zagrebina, S. A., Mokhov, V. G., & Tsimbol, V. I. (2019). Electrical energy consumption prediction is based on the recurrent neural network. *Procedia Computer Science*, *150*, 340–346.

Zheng, H., Liao, H., Chen, L., Xiong, W., Chen, T., & Luo, J. (2020). Example-guided image synthesis across arbitrary scenes using masked spatial-channel attention and self-supervision. *ArXiv Preprint ArXiv:2004.10024*.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2223–2232.

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., & Shechtman, E. (2017). Multimodal Image-to-Image Translation by Enforcing Bi-Cycle Consistency. *Advances in Neural Information Processing Systems*, 465–476.

CHAPTER 6 GENERAL DISCUSSION, CONCLUSION & FUTURE WORK

This chapter discusses the contributions presented in Chapters 3, 4, and 5 towards accomplishing the specific objectives stated in the introduction of this thesis. The first specific objective is to implement a novel data representation for maximizing the global value of the numerical data commonly available in different industrial applications. This objective was achieved in Chapter 3 by developing the "Polygon Generation" approach that can transform the numerical data into images of polygons where the interrelationships between the data variables are properly represented through the Hamiltonian cycles. These polygon images are fed into a DL architecture (e.g. CNN) to train a robust and accurate classification model for challenging (noisy, unbalanced, etc.) datasets.

The second objective in this doctoral research is to implement a multi-output regression method using DL and the developed polygon generation. In Chapter 4, this objective was achieved through adapting the polygon generation technique to deal with the collected numerical data with continuous outputs and ill-defined distributions. A generative DL technique is the key to this adaptation. Image-to-image translation technique using cGAN is used to translate the polygon images representing the data variables into those representing the outputs. This translation solves the complex distribution matching problems found in most industrial data without making unrealistic assumptions.

The third and last objective is to go further and apply the developed polygon generation and recurrent DL approach for time-series prediction. Expressing the highly dynamic behavior of the industrial systems accurately is the goal of this objective. In Chapter 5, this objective was achieved through using the polygon generation technique and a video-to-video translation technique. The translation technique is used for mapping polygon streams of input variables into their corresponding time-series outputs. The DL model's trustworthiness is ensured using a metric that measures the quality of the synthesized polygon streams.

The labeling process in industrial processes is a highly distinct problem due to the intricate nature of these high-dimensional data. This process needs an experienced operator with sufficient knowledge since it is more complex than the labeling of the typical problems found in domains

such as computer vision, therefore, the crowd-sourcing approach is inappropriate for most tasks (Gärtler et al., 2021). Moreover, the confidentiality of the industrial data in some cases can hinder the involvement of third parties (e.g. researchers and experts in the targeted domain) in the labeling process. It is worth mentioning that the developed approach using the polygon generation and the video-to-video translation saves the effort and the time spent in the labeling phase, as the translation technique is based on cycle-consistent GAN that works in an unsupervised way.

The above-mentioned contributions are validated using different challenging industrial datasets collected from plants located in Canada. The first contribution in Chapter 3 was validated using a dataset collected from a reboiler system in a thermomechanical pulp mill located in Canada. The second one in Chapter 4 was validated using another dataset collected from a black liquor recovery boiler equipment in a heat recovery network in a Kraft pulp & paper mill located in Canada. The third one in Chapter 5 was validated using another challenging dataset collected from black liquor concentrator in a Canadian pulp mill. By comparing the performance of each proposed method with other baseline ML and DL methods, it was observed that the proposed methods outperformed the other baseline predictors.

Given the contributions in this thesis work, our main objective of improving the performance of industrial systems is achieved through exploiting the modeling power of DL and improving the data representation, thus maximizing its global value. The original contribution of this thesis is the introduction of polygon generation and generative DL as two main blocks that capture the true distribution of the industrial data. This will help accurately predict the system performance which is an urgent and prioritized need for many industrial systems.

Polygon generation achieves one of the data-centric AI approaches that focus on the quality of data and how to improve it through an efficient representation. All interrelationships between the data variables and outputs are represented in the form of Hamiltonian cycles within the polygons. It is worth mentioning that according to the best of our knowledge, it is the first time to use the generative DL techniques (image-to-image and video-to-video translation techniques) in this industrial context using the available numerical data. Moreover, all the contributions in this thesis are end-to-end learning processes that do not need the effort of manual feature engineering that is done by the process expert in a tedious manner.

Despite the above-mentioned strengths of this thesis work, the resolution of the polygon images or videos is critical in the prediction methods. That is attributed to the fact that the numerical values of the outputs are extracted from the synthesized polygons using an image processing procedure. The accuracy of this procedure depends on the resolution of these polygons. Increasing the resolution may significantly increase the accuracy of the whole prediction approach, but will result in a more computationally expensive task.

In order to train different DL architecture for translation tasks, a massive amount of data and an infrastructure with high memory and powerful GPUs are needed. However, this training phase is done only once in the training phase in the developed methods. On the other hand, the online or testing phase is neither computationally expensive nor time-consuming. Accordingly, this can guarantee the operationally deployable implementation of the proposed approach in industrial settings.

The adaptability of the proposed methods in the thesis allows their application in different industrial problems. This is owed to the flexibility of the DL algorithms used in the prediction methods and the generic nature of the polygon generation as a data representation method. This ensures the versatility of the proposed approach and its potential success in systems other than the process industry.

The contribution in this thesis paves the way to efficient industrial data fusion in terms of merging numerical data, images, and videos. We claim that the polygon generation representation can facilitate the fusion of different types of data at different levels (raw, information, and knowledge levels). This can help efficiently exploit the available heterogeneous data to maximize the global value of isolated data silos to provide the users with valuable knowledge. The developed methods in this thesis are generic and can be applied to various fields such as the forestry industry where growth models are highly desired to predict different attributes such as the total biomass volume, the biomass quality, etc.

One of our future research directions is to deal with the interpretability of the DL models and develop a platform that can integrate and process different types of data in terms of type, format, structure, and sampling frequencies. The platform will include an ensemble of DL models; each used to process specific data types.

It is worth mentioning that the polygon generation technique was inspired by the concept of parallel coordinate plots (PCP) proposed in (Inselberg & Dimsdale, 1990). Despite using PCP in several applications for the visualization of high-dimensional data, it still has some limitations that need to be addressed. The order of variables in PCP significantly changes the pattern that could be extracted from the targeted datasets. Therefore, it still needs the heavy involvement of the process expert to manually extract these patterns using different orders of variables. On the other hand, the polygon generation method saves the effort of the process expert through a systematic way of representing all possible interrelationships between the data variables through Hamiltonian cycles. We claim that the proposed methods in this thesis and their advantage of visualizing the KPI changes and the input variables in the form of representational videos (streams) can play a key role in the interpretation of the trained DL models. Accordingly, this can offer some insights into the reasoning of the DL architectures. The final goal is to provide the end-users with an accurate and transparent knowledge base with explainable rules.

This doctoral research opens the door to deal with information security problems and can play a key role in the data encryption domain. The raw data representation in the form of polygons can tackle the problem of data privacy and governance and this can facilitate data sharing without affecting its confidentiality. Moreover, dealing with generative DL models instead of the raw data can motivate the data sharing process, thus facilitating the implementation of data fusion platforms that would be open for different stakeholders (public, governments, private companies, organizations, etc.).

REFERENCES

Abadi, Mart\'\in, Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., & others. (2016). Tensorflow: A system for large-scale machine learning. *12th $\{$USENIX$\}$ Symposium on Operating Systems Design and Implementation ($\{$OSDI$\}$ 16)*, 265–283.

Abadi, Martin, Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., S. Corrado, G., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Yangqing, J., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/

Afrasiabi, S., Afrasiabi, M., Parang, B., Mohammadi, M., Arefi, M. M., & Rastegar, M. (2019). Wind Turbine Fault Diagnosis with Generative-Temporal Convolutional Neural Network. *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, 1–5.

Aggarwal, K., Kirchmeyer, M., Yadav, P., Keerthi, S. S., & Gallinari, P. (2019). *Benchmarking Regression Methods: A comparison with CGAN*. *May*. http://arxiv.org/abs/1905.12868

Alam, F., Mehmood, R., Katib, I., Albogami, N. N., & Albeshri, A. (2017). Data fusion and IoT for smart ubiquitous environments: A survey. *IEEE Access*, *5*, 9533–9554.

Alpaydin, E. (2010). *Introduction to machine learning, 2nd edn. Adaptive computation and machine learning*. The MIT Press (February 2010).

Amazouz, M. (2015). *Improving Process Operation Using the Power of Advanced Data Analysis*. https://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/canmetenergy/files/pubs/EXPLORE-brochure_ EN.pdf

Anaconda Software Distribution. (2020). In *Anaconda Documentation*. Anaconda Inc. https://docs.anaconda.com/

Andersson, E., & Thollander, P. (2019). Key performance indicators for energy management in the Swedish pulp and paper industry. *Energy Strategy Reviews*, *24*(December 2018), 229–235. https://doi.org/10.1016/j.esr.2019.03.004

Andrew Ng Launches A Campaign For Data-Centric AI. (n.d.).

https://www.forbes.com/sites/gilpress/2021/06/16/andrew-ng-launches-a-campaign-for-data-centric-ai/?sh=5dea92f374f5

Ardsomang, T., Hines, J. W., & Upadhyaya, B. R. (2013). Heat exchanger fouling and estimation of remaining useful life. *Annual Conference of the PHM Society*, *5*(1).

Arendt, J. S., & Lorenzo, D. K. (2010). *Evaluating process safety in the chemical industry: A user's guide to quantitative risk analysis* (Vol. 3). John Wiley \& Sons.

Athar, M., Mohd Shariff, A., Buang, A., Shuaib Shaikh, M., & Ishaq Khan, M. (2019). Review of Process Industry Accidents Analysis towards Safety System Improvement and Sustainable Process Design. *Chemical Engineering and Technology*, *42*(3), 524–538. https://doi.org/10.1002/ceat.201800215

Ayubi Rad, M. A., & Yazdanpanah, M. J. (2015). Designing supervised local neural network classifiers based on EM clustering for fault diagnosis of Tennessee Eastman process. *Chemometrics and Intelligent Laboratory Systems*, *146*, 149–157. https://doi.org/10.1016/j.chemolab.2015.05.013

Bache, K., & Lichman, M. (2013). *UCI machine learning repository*.

Bai, Y., Xie, J., Wang, D., Zhang, W., & Li, C. (2021). A manufacturing quality prediction model based on AdaBoost-LSTM with rough knowledge. *Computers and Industrial Engineering*, *155*(January), 107227. https://doi.org/10.1016/j.cie.2021.107227

Bajpai, P. (2018). Brief Description of the Pulp and Papermaking Process. In *Biotechnology for pulp and paper processing* (pp. 9–26). Springer.

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., & others. (2012). NCBI GEO: archive for functional genomics data sets—update. *Nucleic Acids Research*, *41*(D1), D991--D995.

Bashkirova, D., Usman, B., & Saenko, K. (2018). *Unsupervised Video-to-Video Translation*. *Nips*. http://arxiv.org/abs/1806.03698

Bathelt, A., Ricker, N. L., & Jelali, M. (2015). Revision of the tennessee eastman process model. *IFAC-PapersOnLine*, *48*(8), 309–314.

Baturynska, I., & Martinsen, K. (2020). Prediction of geometry deviations in additive

manufactured parts: comparison of linear regression with machine learning algorithms. *Journal of Intelligent Manufacturing*, *32*(1), 179–200. https://doi.org/10.1007/s10845-020-01567-0

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Biermann, C. J. (1996). *Handbook of pulping and papermaking*. Elsevier.

Borovykh, A., Bohte, S., & Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10614 LNCS*, 729–730.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley \& Sons.

Brown, R. G., & Meyer, R. F. (1961). The fundamental theorem of exponential smoothing. *Operations Research*, *9*(5), 673–685.

Bustillo, A., Pimenov, D. Y., Mia, M., & Kapłonek, W. (2020). Machine-learning for automatic prediction of flatness deviation considering the wear of the face mill teeth. *Journal of Intelligent Manufacturing*, *mm*. https://doi.org/10.1007/s10845-020-01645-3

Bustillo, A., Reis, R., Machado, A. R., & Pimenov, D. Y. (2020). Improving the accuracy of machine-learning models with data from machine test repetitions. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01661-3

Camacho, J., Magán-Carrión, R., García-Teodoro, P., & Treinen, J. J. (2016). Networkmetrics: multivariate big data analysis in the context of the internet. *Journal of Chemometrics*, *30*(9), 488–505. https://doi.org/10.1002/cem.2806

Cao, S., Wen, L., Li, X., & Gao, L. (2018). Application of Generative Adversarial Networks for Intelligent Fault Diagnosis. *IEEE International Conference on Automation Science and Engineering*, *2018-Augus*, 711–715. https://doi.org/10.1109/COASE.2018.8560528

Castanedo, F. (2013). A review of data fusion techniques. In *The Scientific World Journal* (Vol.

2013). Hindawi Publishing Corporation. https://doi.org/10.1155/2013/704504

Chamzas, D., Chamzas, C., & Moustakas, K. (2020). cMinMax: A fast algorithm to find the corners of an N-dimensional convex polytope. *ArXiv:2011.14035v2*.

Chen, X., Zhang, B., & Gao, D. (2020). Bearing fault diagnosis base on multi-scale CNN and LSTM model. *Journal of Intelligent Manufacturing*, *December 2019*. https://doi.org/10.1007/s10845-020-01600-2

Chen, Z., Zeng, X., Li, W., & Liao, G. (2016). Machine fault classification using deep belief network. *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, *2016-July*(51475170). https://doi.org/10.1109/I2MTC.2016.7520473

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Choi, H., Lee, H., & Kim, H. (2009). Fast detection and visualization of network attacks on parallel coordinates. *Computers & Security*, *28*(5), 276–288.

Chollet, F., & others. (2015). *Keras*. GitHub.

Chollet, F., & others. (2018). Keras: The python deep learning library. *Astrophysics Source Code Library*.

Cocchi, M. (2019). Introduction: Ways and Means to Deal With Data From Multiple Sources. *Data Handling in Science and Technology*, *31*, 1–26. https://doi.org/10.1016/B978-0-444-63984-4.00001-6

D'Angelo, M. F. S. V, Palhares, R. M., Camargos Filho, M. C. O., Maia, R. D., Mendes, J. B., & Ekel, P. Y. (2016). A new fault classification approach applied to Tennessee Eastman benchmark process. *Applied Soft Computing*, *49*, 676–686. https://doi.org/https://doi.org/10.1016/j.asoc.2016.08.040

Data-Centric AI Competition. (n.d.). https://https-deeplearning-ai.github.io/data-centric-comp/

Demir, U., & Unal, G. (2018). Patch-based image inpainting with generative adversarial networks.

ArXiv:1803.07422v1.

Dong, D., Li, X.-Y., & Sun, F.-Q. (2017). Life prediction of jet engines based on LSTM-recurrent neural networks. *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 1–6.

Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, *17*(3), 245–255.

Dunia, R., Edgar, T. F., & Nixon, M. (2013). Process monitoring using principal components in parallel coordinates. *AIChE Journal*, *59*(2), 445–456.

Durall, R., Chatzimichailidis, A., Labus, P., & Keuper, J. (2020). Combating Mode Collapse in GAN training: An Empirical Analysis using Hessian Eigenvalues. *ArXiv Preprint ArXiv:2012.09673*.

Duvall, P. M., & Riggs, J. B. (2000). On-line optimization of the Tennessee Eastman challenge problem. *Journal of Process Control*, *10*(1), 19–33. https://doi.org/https://doi.org/10.1016/S0959-1524(99)00041-4

Elhefnawy, M., Ragab, A., & Ouali, M.-S. (2021a). Fault classification in the process industry using polygon generation and deep learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-021-01742-x

Elhefnawy, M., Ragab, A., & Ouali, M. S. (2021b). Fault classification in the process industry using polygon generation and deep learning. *Journal of Intelligent Manufacturing*, *0123456789*. https://doi.org/10.1007/s10845-021-01742-x

Elizabeth Bush Nathan Gillett, E. W. J. F., & others. (2019). *Canada's Changing Climate Report*. https://changingclimate.ca/CCCR2019/

Environment challenges | Climate Action. (n.d.). https://ec.europa.eu/clima/policies/adaptation/how/challenges_en#tab-0-1

Eren, L., Ince, T., & Kiranyaz, S. (2019). A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier. *Journal of Signal Processing Systems*, *91*(2), 179–189.

Essien, A., & Giannetti, C. (2020). A Deep Learning Model for Smart Manufacturing Using Convolutional LSTM Neural Network Autoencoders. *IEEE Transactions on Industrial*

Informatics, *16*(9), 6069–6078. https://doi.org/10.1109/TII.2020.2967556

FP, L. (1996). *Loss Prevention in the Process Industries: Hazard Identification, Assessment and Control*.

Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83–85.

Fu, K., Peng, J., Zhang, H., Wang, X., & Jiang, F. (2020). Image super-resolution based on generative adversarial networks: A brief review. *Computers, Materials and Continua*, *64*(3), 1977–1997. https://doi.org/10.32604/cmc.2020.09882

Fuentes, R., Fuster, B., & Lillo-Bañuls, A. (2016). A three-stage DEA model to evaluate learning-teaching technical efficiency: Key performance indicators and contextual variables. *Expert Systems with Applications*, *48*, 89–99. https://doi.org/10.1016/j.eswa.2015.11.022

Gamboa, J. C. B. (2017). *Deep Learning for Time-Series Analysis*.

Gärtler, M., Khaydarov, V., Klöpper, B., & Urbas, L. (2021). The Machine Learning Life Cycle in Chemical Operations – Status and Open Challenges. *Chemie Ingenieur Technik*, *12*, 1–19. https://doi.org/10.1002/cite.202100134

Ge, Z. (2017). Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemometrics and Intelligent Laboratory Systems*, *171*(September), 16–25. https://doi.org/10.1016/j.chemolab.2017.09.021

Gecgel, O., Ekwaro-Osire, S., Dias, J. P., Serwadda, A., Alemayehu, F. M., & Nispel, A. (2019). Gearbox Fault Diagnostics Using Deep Learning with Simulated Data. *Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management*, 1–8.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471.

Gheisari, M., Wang, G., & Bhuiyan, M. Z. A. (2017). A survey on deep learning in big data. *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, *2*, 173–180.

Golshan, M., boozarjomehry, R. B., & Pishvaie, M. R. (2005). A new approach to real time optimization of the Tennessee Eastman challenge problem. *Chemical Engineering Journal*,

112(1), 33–44. https://doi.org/https://doi.org/10.1016/j.cej.2005.06.005

Gomez-Cabrero, D., Abugessaisa, I., Maier, D., Teschendorff, A., Merkenschlager, M., Gisel, A., Ballestar, E., Bongcam-Rudloff, E., Conesa, A., & Tegnér, J. (2014). *Data integration in the era of omics: current and future challenges*. BioMed Central.

Goodfellow, I. (2016). NIPS 2016 tutorial: Generative adversarial networks. *ArXiv*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. *Commun. ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative Adversarial Networks. *Commun. ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Gunerkar, R. S., Jalan, A. K., & Belgamwar, S. U. (2019). Fault diagnosis of rolling element bearing based on artificial neural network. *Journal of Mechanical Science and Technology*, *33*(2), 505–511. https://doi.org/10.1007/s12206-019-0103-x

Han, Z., Zhao, J., Leung, H., Ma, K. F., & Wang, W. (2021). A Review of Deep Learning Models for Time Series Prediction. *IEEE Sensors Journal*, *21*(6), 7833–7848. https://doi.org/10.1109/JSEN.2019.2923982

Hasan, M. J., Sohaib, M., & Kim, J.-M. (2019). 1D CNN-Based Transfer Learning Model for Bearing Fault Diagnosis Under Variable Working Conditions. In S. Omar, W. S. Haji Suhaili, & S. Phon-Amnuaisuk (Eds.), *Computational Intelligence in Information Systems* (pp. 13–23). Springer International Publishing.

Hauser, H., Ledermann, F., & Doleisch, H. (2002). Angular brushing of extended parallel coordinates. *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, 127–130.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

He, Y., Cowell, L., Diehl, A. D., Mobley, H. L., Peters, B., Ruttenberg, A., Scheuermann, R. H., Brinkman, R. R., Courtot, M., Mungall, C., & others. (2009). VO: vaccine ontology. *The 1st International Conference on Biomedical Ontology (ICBO 2009) Nature Precedings*, *2009*.

Hendrickx, D. M., Aerts, H. J. W. L., Caiment, F., Clark, D., Ebbels, T. M. D., Evelo, C. T., Gmuender, H., Hebels, D. G. A. J., Herwig, R., Hescheler, J., & others. (2014). diXa: a data infrastructure for chemical safety assessment. *Bioinformatics*, *31*(9), 1505–1507.

Heo, S., & Lee, J. H. (2018). Fault detection and classification using artificial neural networks. *IFAC-PapersOnLine*, *51*(18), 470–475.

Himeur, Y., Alsalemi, A., Al-Kababji, A., Bensaali, F., & Amira, A. (2020). Data fusion strategies for energy efficiency in buildings: Overview, challenges and novel orientations. *Information Fusion*, *64*(June), 99–120. https://doi.org/10.1016/j.inffus.2020.07.003

Hoermann, S., Bach, M., & Dietmayer, K. (2018). Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2056–2063.

Hsu, C. Y., & Liu, W. C. (2020). Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, *0123456789*. https://doi.org/10.1007/s10845-020-01591-0

Huang, J. T., Li, J., & Gong, Y. (2015). An analysis of convolutional neural networks for speech recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2015-Augus*, 4989–4993. https://doi.org/10.1109/ICASSP.2015.7178920

Hurley, C. B., & Oldford, R. W. (2010). Pairwise display of high-dimensional information via eulerian tours and hamiltonian decompositions. *Journal of Computational and Graphical Statistics*, *19*(4), 861–886.

Ian Goodfellow Yoshua Bengio, A. C. (2017). The Deep Learning Book. *MIT Press*, *521*(7553), 785. https://doi.org/10.1016/B978-0-12-391420-0.09987-X

Inselberg, A. (2009). *Parallel coordinates*. Springer.

Inselberg, A., & Dimsdale, B. (1990). Parallel coordinates: a tool for visualizing multi-dimensional geometry. *Proceedings of the 1st Conference on Visualization'90*, 361–378.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern*

Recognition, CVPR 2017, *2017-Janua*, 5967–5976. https://doi.org/10.1109/CVPR.2017.632

Jain, A., Smarra, F., Behl, M., & Mangharam, R. (2018). Data-driven model predictive control with regression trees-an application to building energy management. *ACM Transactions on Cyber-Physical Systems*, *2*(1), 1–21. https://doi.org/10.1145/3127023

Jain, S., Seth, G., Paruthi, A., Soni, U., & Kumar, G. (2020). Synthetic data augmentation for surface defect detection and classification using deep learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01710-x

Jebara, T. (2012). *Machine learning: discriminative and generative* (Vol. 755). Springer Science & Business Media.

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221–231.

Jing, C., & Hou, J. (2015). SVM and PCA based fault classification approaches for complicated industrial process. *Neurocomputing*, *167*, 636–642. https://doi.org/https://doi.org/10.1016/j.neucom.2015.03.082

Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9906 LNCS*, 694–711. https://doi.org/10.1007/978-3-319-46475-6_43

Jurkovic, Z., Cukor, G., Brezocnik, M., & Brajkovic, T. (2018). A comparison of machine learning methods for cutting parameters prediction in high speed turning process. *Journal of Intelligent Manufacturing*, *29*(8), 1683–1693. https://doi.org/10.1007/s10845-016-1206-1

Karagkouni, D., Paraskevopoulou, M. D., Chatzopoulos, S., Vlachos, I. S., Tastsoglou, S., Kanellos, I., Papadimitriou, D., Kavakiotis, I., Maniou, S., Skoufos, G., & others. (2017). DIANA-TarBase v8: a decade-long collection of experimentally supported miRNA--gene interactions. *Nucleic Acids Research*, *46*(D1), D239--D245.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2019). Analyzing and improving the image quality of StyleGAN. *ArXiv*, 8110–8119.

Kasuya, E. (2019). On the use of r and r squared in correlation and regression. *Ecological Research*, *34*(1), 235–236. https://doi.org/10.1111/1440-1703.1011

Kedem, B., & Fokianos, K. (2005). *Regression models for time series analysis* (Vol. 488). John Wiley \& Sons.

Khaleghi, B., Khamis, A., Karray, F. O., & Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, *14*(1), 28–44. https://doi.org/10.1016/j.inffus.2011.08.001

King, R. D., Feng, C., & Sutherland, A. (1995). Statlog: comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence an International Journal*, *9*(3), 289–333.

Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. *Big Data & Society*, *1*(1), 205395171452848. https://doi.org/10.1177/2053951714528481

Kletz, T. A. (1988). *Learning from accidents in industry*. Butterworth-Heinemann.

Kourniotis, S. P., Kiranoudis, C. T., & Markatos, N. C. (2000). Statistical analysis of domino chemical accidents. *Journal of Hazardous Materials*, *71*(1–3), 239–252.

Krämer, S., & Engell, S. (2018). *Resource efficiency of processing plants: monitoring and improvement*. John Wiley & Sons.

Kusiak, A. (2020). Convolutional and generative adversarial neural networks in manufacturing. *International Journal of Production Research*, *58*(5), 1594–1604. https://doi.org/10.1080/00207543.2019.1662133

Lahat, D., Adali, T., & Jutten, C. (2015). Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. *Proceedings of the IEEE*, *103*(9), 1449–1477. https://doi.org/10.1109/JPROC.2015.2460697

Lanzetti, N., Lian, Y. Z., Cortinovis, A., Dominguez, L., Mercangöz, M., & Jones, C. (2019). Recurrent neural network based MPC for process industries. *2019 18th European Control Conference (ECC)*, 1005–1010.

Lapedes, A., & Farber, R. (1987). *Nonlinear signal processing using neural networks: Prediction and system modelling*.

Larsson, T., Hestetun, K., Hovland, E., & Skogestad, S. (2001). Self-optimizing control of a large-scale plant: The Tennessee Eastman process. *Industrial & Engineering Chemistry Research*, *40*(22), 4889–4901.

Larsson, T., & Skogestad, S. (2000). *Plantwide control-A review and a new design procedure*.

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, *6*(4), 239–242.

Lau, B. P. L., Marakkalage, S. H., Zhou, Y., Hassan, N. U., Yuen, C., Zhang, M., & Tan, U. X. (2019). A survey of data fusion in smart city applications. *Information Fusion*, *52*(January), 357–374. https://doi.org/10.1016/j.inffus.2019.05.004

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

LeCun, Y., Bengio, Y., & others. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(10), 1995.

Lee, D., Siu, V., Cruz, R., & Yetman, C. (2016). Convolutional Neural Net and Bearing Fault Analysis. *Int'l Conf. Data Mining*, 194–200. https://doi.org/https://pdfs.semanticscholar.org/6e45/f39b1e50cfd10deaabd1d786f

Lee, K. B., Cheon, S., & Kim, C. O. (2017). A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, *30*(2), 135–142. https://doi.org/10.1109/TSM.2017.2676245

Li, M. J., & Tao, W. Q. (2017). Review of methodologies and polices for evaluation of energy efficiency in high energy-consuming industry. *Applied Energy*, *187*, 203–215. https://doi.org/10.1016/j.apenergy.2016.11.039

Li, P., Chen, Z., & Zhang, J. (2020). *A Survey on Deep Learning for Multimodal Data Fusion*.

Li, Xiang, Zhang, W., Ding, Q., & Sun, J. Q. (2020). Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. *Journal of Intelligent Manufacturing*, *31*(2), 433–452. https://doi.org/10.1007/s10845-018-1456-1

Li, Xin, Liang, Y., Zhao, M., Wang, C., & Jiang, Y. (2019). Few-shot learning with generative adversarial networks based on WOA13 data. *Computers, Materials and Continua*, *60*(3), 1073–1085.

Li, Z., Wang, Y., & Wang, K. (2020). A data-driven method based on deep belief networks for backlash error prediction in machining centers. *Journal of Intelligent Manufacturing*, *31*(7), 1693–1705. https://doi.org/10.1007/s10845-017-1380-9

Lindberg, C. F., Tan, S., Yan, J., & Starfelt, F. (2015). Key Performance Indicators Improve Industrial Performance. *Energy Procedia*, *75*, 1785–1790. https://doi.org/10.1016/j.egypro.2015.07.474

Lipton, Z. C., Elkan, C., & Naryanaswamy, B. (2014). Optimal thresholding of classifiers to maximize F1 measure. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 225–239.

Liu, M. Y., Huang, X., Yu, J., Wang, T. C., & Mallya, A. (2021). Generative Adversarial Networks for Image and Video Synthesis: Algorithms and Applications. *Proceedings of the IEEE*, *109*(5), 839–862. https://doi.org/10.1109/JPROC.2021.3049196

Liu, X., Yin, G., Shao, J., Wang, X., & Li, H. (2019). Learning to predict layout-to-image conditional convolutions for semantic image synthesis. *ArXiv Preprint ArXiv:1910.06809*.

Long, M., Peng, F., & Zhu, Y. (2019). Identifying natural images and computer generated graphics based on binary similarity measures of PRNU. *Multimedia Tools and Applications*, *78*(1), 489–506.

Lv, F., Wen, C., Bao, Z., & Liu, M. (2016). Fault diagnosis based on deep learning. *Proceedings of the American Control Conference*, *2016-July*(2), 6851–6856. https://doi.org/10.1109/ACC.2016.7526751

Mandreoli, F., & Montangero, M. (2019). Dealing With Data Heterogeneity in a Data Fusion Perspective: Models, Methodologies, and Algorithms. In *Data Handling in Science and Technology* (Vol. 31). https://doi.org/10.1016/B978-0-444-63984-4.00009-0

Mart\’\in-Lopo, M. M., Boal, J., & Sánchez-Miralles, Á. (2020). A literature review of IoT energy platforms aimed at end users. *Computer Networks*, *171*, 107101.

Martens, H. (2015). Quantitative Big Data: where chemometrics can contribute. *Journal of Chemometrics*, *29*(11), 563–581. https://doi.org/10.1002/cem.2740

Mathieu, M., Couprie, C., & LeCun, Y. (2016). Deep multi-scale video prediction beyond mean

square error. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, *2015*, 1–14.

McAvoy, T. J., & Ye, N. (1994). Base control for the Tennessee Eastman problem. *Computers & Chemical Engineering*, *18*(5), 383–413.

McEntyre, J., & Lipman, D. (2001). PubMed: bridging the information gap. *Cmaj*, *164*(9), 1317–1319.

Meng, T., Jing, X., Yan, Z., & Pedrycz, W. (2020). A survey on machine learning for data fusion. *Information Fusion*, *57*(2), 115–129. https://doi.org/10.1016/j.inffus.2019.12.001

Mirza, M., & Osindero, S. (2014). *Conditional Generative Adversarial Nets*. 1–7. http://arxiv.org/abs/1411.1784

Narayanasamy, R., & Padmanabhan, P. (2012). Comparison of regression and artificial neural network model for the prediction of springback during air bending process of interstitial free steel sheet. *Journal of Intelligent Manufacturing*, *23*(3), 357–364. https://doi.org/10.1007/s10845-009-0375-6

National Inventory Report. (2019). *GREENHOUSE GAS SOURCES AND SINKS IN CANADA CANADA'S SUBMISSION TO THE UNITED NATIONS FRAMEWORK CONVENTION ON CLIMATE CHANGE Executive Summary*.

Neurohive. (n.d.). *FaceStyleGAN: GAN Network Generates Style Portraits in Snapchat*. https://neurohive.io/en/news/facestylegan-gan-network-generates-style-portraits-in-snapchat/

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems*, 841–848.

Olmschenk, G., Zhu, Z., & Tang, H. (2019). Generalizing semi-supervised generative adversarial networks to regression using feature contrasting. *Computer Vision and Image Understanding*, *186*(June), 1–12. https://doi.org/10.1016/j.cviu.2019.06.004

Olsina, L., & Martin, M. de los A. (2004). Ontology for software metrics and indicators. *J. Web Eng.*, *2*(4), 262–281.

Om, H., & Kundu, A. (2012). A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, 131–136.

Pan, R. (2010). H olt--W inters Exponential Smoothing. *Wiley Encyclopedia of Operations Research and Management Science*.

Park, T., Liu, M.-Y., Wang, T.-C., & Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2337–2346.

Parmenter, D. (2020). *Key perfomance indicator developing, implementing and using winning KPIs*. John Wiley & Sons.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, D., Liu, Z., Wang, H., Qin, Y., & Jia, L. (2019). A novel deeper one-dimensional CNN with residual learning for fault diagnosis of wheelset bearings in high-speed trains. *IEEE Access*, *7*, 10278–12093. https://doi.org/10.1109/ACCESS.2018.2888842

Petzold, B., Roggendorf, M., Rowshankish, K., & Sporleder, C. (2020). Designing data governance that delivers value. *Mc Kinsey Digital*, *June*. https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/designing-data-governance-that-delivers-value

Qi, X., Yuan, Z., & Han, X. (2015). Diagnosis of misalignment faults by tacholess order tracking analysis and RBF networks. *Neurocomputing*, *169*, 439–448. https://doi.org/10.1016/j.neucom.2014.09.088

Qiu, X., Zhang, L., Ren, Y., Suganthan, P., & Amaratunga, G. (2014). Ensemble deep learning for regression and time series forecasting. *IEEE SSCI 2014 - 2014 IEEE Symposium Series on Computational Intelligence - CIEL 2014: 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning, Proceedings*. https://doi.org/10.1109/CIEL.2014.7015739

Ragab, A., El-Koujok, M., Poulin, B., Amazouz, M., & Yacout, S. (2018). Fault diagnosis in industrial chemical processes using interpretable patterns based on Logical Analysis of Data.

Expert Systems with Applications, *95*, 368–383. https://doi.org/https://doi.org/10.1016/j.eswa.2017.11.045

Ragab, A., Koujok, M. El, Ghezzaz, H., Amazouz, M., Ouali, M.-S., & Yacout, S. (2019). Deep understanding in industrial processes by complementing human expertise with interpretable patterns of machine learning. *Expert Systems with Applications*, *122*, 388–405. https://doi.org/https://doi.org/10.1016/j.eswa.2019.01.011

Ragab, A., Ouali, M. S., Yacout, S., & Osman, H. (2016). Remaining useful life prediction using prognostic methodology based on logical analysis of data and Kaplan–Meier estimation. *Journal of Intelligent Manufacturing*, *27*(5), 943–958. https://doi.org/10.1007/s10845-014-0926-3

Ragab, A., Yacout, S., Ouali, M. S., & Osman, H. (2019). Prognostics of multiple failure modes in rotating machinery using a pattern-based classifier and cumulative incidence functions. *Journal of Intelligent Manufacturing*, *30*(1), 255–274. https://doi.org/10.1007/s10845-016-1244-8

Ricker, N. L. (1996). Decentralized control of the Tennessee Eastman challenge process. *Journal of Process Control*, *6*(4), 205–221.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., & Waldman-Brown, A. (2019). Tackling climate change with machine learning. *ArXiv Preprint ArXiv:1906.05433*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015* (pp. 234–241). Springer International Publishing.

Sakpal, M. (2021). *12 Actions to Improve Your Data Quality*. https://www.gartner.com/smarterwithgartner/how-to-improve-your-data-quality

Sanderson, C., & Gruen, R. (2006). *Analytical models for decision-making*. McGraw-Hill Education (UK).

Santamaria, R., Therón, R., & Quintales, L. (2008). A visual analytics approach for understanding biclustering results from microarray data. *BMC Bioinformatics*, *9*(1), 247.

Schat, E., van de Schoot, R., Kouw, W. M., Veen, D., & Mendrik, A. M. (2020). The data representativeness criterion: Predicting the performance of supervised classification based on data set similarity. *PLoS ONE*, *15*(8 August), 1–16. https://doi.org/10.1371/journal.pone.0237009

Schmidt, C., Li, W., Thiede, S., Kornfeld, B., Kara, S., & Herrmann, C. (2016). Implementing Key Performance Indicators for Energy Efficiency in Manufacturing. *Procedia CIRP*, *57*, 758–763. https://doi.org/10.1016/j.procir.2016.11.131

Shao, S., Wang, P., & Yan, R. (2019). Generative adversarial networks for data augmentation in machine fault diagnosis. *Computers in Industry*, *106*, 85–93. https://doi.org/10.1016/j.compind.2019.01.001

Shao, S. Y., Sun, W. J., Yan, R. Q., Wang, P., & Gao, R. X. (2017). A Deep Learning Approach for Fault Diagnosis of Induction Motors in Manufacturing. *Chinese Journal of Mechanical Engineering (English Edition)*, *30*(6), 1347–1356. https://doi.org/10.1007/s10033-017-0189-y

Shen, Y., Yang, F., Habibullah, M. S., Ahmed, J., Das, A. K., Zhou, Y., & Ho, C. L. (2020). Predicting tool wear size across multi-cutting conditions using advanced machine learning techniques. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01625-7

Siirtola, H., & Räihä, K.-J. (2006). Interacting with parallel coordinates. *Interacting with Computers*, *18*(6), 1278–1309.

Simmonds, A., Sandilands, P., & Van Ekert, L. (2004). An ontology for network security attacks. *Asian Applied Computing Conference*, 317–323.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv Preprint ArXiv:1409.1556*.

Soualhi, M., El Koujok, M., Nguyen, K. T. P., Medjaher, K., Ragab, A., Ghezzaz, H., Amazouz, M., & Ouali, M.-S. (2021). Adaptive prognostics in a controlled energy conversion process based on long-and short-term predictors. *Applied Energy*, *283*, 116049.

Sridharan, M. (2017). *Using governance to achieve data quality objectives*. https://www.bloomberg.com/professional/blog/using-governance-achieve-data-quality-

objectives/

Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015). Unsupervised learning of video representations using lstms. *International Conference on Machine Learning*, 843–852.

Talbot, D., & Boiral, O. (2013). Can we trust corporates GHG inventories? An investigation among Canada's large final emitters. *Energy Policy*, *63*, 1075–1085.

Tang, H., Qi, X., Xu, D., Torr, P. H. S., & Sebe, N. (2020). Edge guided GANs with semantic preserving for semantic image synthesis. *ArXiv Preprint ArXiv:2003.13898*.

Telea, A. C. (2007). Data Visualization: Principles and practice. In *Data Visualization: Principles and Practice*. https://doi.org/10.1201/b10679

Tian, J., Morillo, C., Azarian, M. H., & Pecht, M. (2016). Motor Bearing Fault Detection Using Spectral Kurtosis-Based Feature Extraction Coupled With K-Nearest Neighbor Distance Analysis. *IEEE Transactions on Industrial Electronics*, *63*(3), 1793–1803. https://doi.org/10.1109/TIE.2015.2509913

Tidriri, K., Chatti, N., Verron, S., & Tiplica, T. (2016). Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual Reviews in Control*, *42*, 63–81.

Uguz, S., & Ipek, O. (2021). Prediction of the parameters affecting the performance of compact heat exchangers with an innovative design using machine learning techniques. *Journal of Intelligent Manufacturing*, *2013*. https://doi.org/10.1007/s10845-020-01729-0

Uysal, M. P., & Sogut, M. Z. (2017). An integrated research for architecture-based energy management in sustainable airports. *Energy*, *140*, 1387–1397.

Vakkilainen, E., & others. (2005). *Kraft recovery boilers--Principles and practice*.

Vasara, P. (2001). Scandinavia: Through different eyes: environmental issues in Scandinavia and North America. *Tappi Journal*, *84*(6), 46–49.

Vlachos, I. P. (2014). A hierarchical model of the impact of RFID practices on retail supply chain performance. *Expert Systems with Applications*, *41*(1), 5–15. https://doi.org/10.1016/j.eswa.2013.07.006

Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Generating videos with scene dynamics.

Advances in Neural Information Processing Systems, *29*, 613–621.

Wald, L. (1999). *Definitions and terms of reference in data fusion*.

Wang, Jin, Yang, Y., Wang, T., Sherratt, R. S., & Zhang, J. (2020). Big data service architecture: a survey. *Journal of Internet Technology*, *21*(2), 393–405.

Wang, Jinjiang, Yan, J., Li, C., Gao, R. X., & Zhao, R. (2019). Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction. *Computers in Industry*, *111*, 1–14. https://doi.org/10.1016/j.compind.2019.06.001

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., & Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8798–8807.

Wang, Y., Zhou, J., Zheng, L., & Gogu, C. (2020). An end-to-end fault diagnostics method based on convolutional neural network for rotating machinery with multiple case studies. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01671-1

Wang, Z., & Bovik, A. C. (2002). A universal image quality index. *IEEE Signal Processing Letters*, *9*(3), 81–84.

Wegman, E. J. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, *85*(411), 664–675.

White, F. E. (1991). JDL, data fusion lexicon. *Technical Panel for C*, *3*, 15.

Wilke, C. O. (2019). Fundamentals of Data Visualization. In *Serial Mentor*.

Wong, P. C., & Bergeron, R. D. (1996). Multiresolution multidimensional wavelet brushing. *Proceedings of Seventh Annual IEEE Visualization'96*, 141–148.

Wu, A. (n.d.). *A Chat with Andrew on MLOps: From Model-Centric to Data-Centric AI. 2021*.

Wu, H., & Zhao, J. (2018). Deep convolutional neural network model based chemical process fault diagnosis. *Computers and Chemical Engineering*, *115*, 185–197. https://doi.org/10.1016/j.compchemeng.2018.04.009

Wu, J., Zhang, C., Xue, T., Freeman, W. T., & Tenenbaum, J. B. (2016). Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. *Advances in Neural*

Information Processing Systems, *Nips*, 82–90.

Xia, C., Pan, Z., Polden, J., Li, H., Xu, Y., & Chen, S. (2021). Modelling and prediction of surface roughness in wire arc additive manufacturing using machine learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-020-01725-4

Xu, Y., Pei, J., & Lai, L. (2017). Deep Learning Based Regression and Multiclass Models for Acute Oral Toxicity Prediction with Automatic Chemical Feature Extraction. *Journal of Chemical Information and Modeling*, *57*(11), 2672–2685. https://doi.org/10.1021/acs.jcim.7b00244

Yin, S., Ding, S. X., Haghani, A., Hao, H., & Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control*, *22*(9), 1567–1581. https://doi.org/https://doi.org/10.1016/j.jprocont.2012.06.009

Yourtechdiet. (n.d.). *Top Tools in Generative Adversarial Networks*. https://www.yourtechdiet.com/blogs/generative-adversarial-networks-tools/

Yuan, X., Li, L., Shardt, Y. A. W., Wang, Y., & Yang, C. (2021). Deep Learning with Spatiotemporal Attention-Based LSTM for Industrial Soft Sensor Model Development. *IEEE Transactions on Industrial Electronics*, *68*(5), 4404–4414. https://doi.org/10.1109/TIE.2020.2984443

Zagrebina, S. A., Mokhov, V. G., & Tsimbol, V. I. (2019). Electrical energy consumption prediction is based on the recurrent neural network. *Procedia Computer Science*, *150*, 340–346.

Zakharov, E., Shysheya, A., Burkov, E., & Lempitsky, V. (2019). Few-shot adversarial learning of realistic neural talking head models. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-Octob*, 9458–9467. https://doi.org/10.1109/ICCV.2019.00955

Zhang, D., Li, Q., Yang, G., Li, L., & Sun, X. (2017). Detection of image seam carving by using weber local descriptor and local binary patterns. *Journal of Information Security and Applications*, *36*, 135–144.

Zhang, J., Zhong, S., Wang, T., Chao, H.-C., & Wang, J. (2020). Blockchain-based systems and applications: a survey. *Journal of Internet Technology*, *21*(1), 1–14.

Zhang, Qing, Gao, J., Dong, H., & Mao, Y. (2018). WPD and DE/BBO-RBFNN for solution of rolling bearing fault diagnosis. *Neurocomputing*, *312*, 27–33. https://doi.org/10.1016/j.neucom.2018.05.014

Zhang, Qingchen, Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, *42*, 146–157.

Zhang, S., Zhang, S., Wang, B., & Habetler, T. G. (2019). Machine learning and deep learning algorithms for bearing fault diagnostics-a comprehensive review. *ArXiv Preprint ArXiv:1901.08247*.

Zhang, W., Li, X., Jia, X. D., Ma, H., Luo, Z., & Li, X. (2020). Machinery fault diagnosis with imbalanced data using deep generative adversarial networks. *Measurement: Journal of the International Measurement Confederation*, *152*. https://doi.org/10.1016/j.measurement.2019.107377

Zhang, Z., & Zhao, J. (2017). A deep belief network based fault diagnosis model for complex chemical processes. *Computers and Chemical Engineering*, *107*, 395–407. https://doi.org/10.1016/j.compchemeng.2017.02.041

Zhao, D., Ivanov, M., Wang, Y., & Du, W. (2020). Welding quality evaluation of resistance spot welding based on a hybrid approach. *Journal of Intelligent Manufacturing*, *0123456789*. https://doi.org/10.1007/s10845-020-01627-5

Zhao, M., Liu, X., Yao, X., & He, K. (2020). Better visual image super-resolution with Laplacian pyramid of generative adversarial networks. *CMC-COMPUTERS MATERIALS \& CONTINUA*, *64*(3), 1601–1614.

Zheng, H., Liao, H., Chen, L., Xiong, W., Chen, T., & Luo, J. (2020). Example-guided image synthesis across arbitrary scenes using masked spatial-channel attention and self-supervision. *ArXiv Preprint ArXiv:2004.10024*.

Zhou, P., Ang, B. W., & Zhou, D. Q. (2012). Measuring economy-wide energy efficiency performance: A parametric frontier approach. *Applied Energy*, *90*(1), 196–200. https://doi.org/10.1016/j.apenergy.2011.02.025

Zhou, W., Li, X., Yi, J., & He, H. (2019). A Novel UKF-RBF Method Based on Adaptive Noise Factor for Fault Diagnosis in Pumping Unit. *IEEE Transactions on Industrial Informatics*,

15(3), 1415–1424. https://doi.org/10.1109/TII.2018.2839062

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2223–2232.

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., & Shechtman, E. (2017). Multimodal Image-to-Image Translation by Enforcing Bi-Cycle Consistency. *Advances in Neural Information Processing Systems*, 465–476.

Zhu, L., Johnsson, C., Mejvik, J., Varisco, M., & Schiraldi, M. (2018). Key performance indicators for manufacturing operations management in the process industry. *IEEE International Conference on Industrial Engineering and Engineering Management*, *2017-Decem*, 969–973. https://doi.org/10.1109/IEEM.2017.8290036

Zitnik, M., Nguyen, F., Wang, B., Leskovec, J., Goldenberg, A., & Hoffman, M. M. (2019). Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, *50*(September 2018), 71–91. https://doi.org/10.1016/j.inffus.2018.09.012