April 2019

A Hierarchical, Fuzzy Inference Approach to Data Filtration and Feature Prioritization in the Connected Manufacturing Enterprise

Phillip Matthew LaCasse
University of Wisconsin-Milwaukee

A HIERARCHICAL, FUZZY INFERENCE APPROACH TO DATA FILTRATION AND FEATURE PRIORITIZATION IN THE CONNECTED MANUFACTURING ENTERPRISE

by

Phillip M. LaCasse

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Engineering

at

The University of Wisconsin – Milwaukee

May 2019

ABSTRACT

A HIERARCHICAL, FUZZY INFERENCE APPROACH TO DATA FILTRATION AND FEATURE PRIORITIZATION IN THE CONNECTED MANUFACTURING ENTERPRISE

by

Phillip M. LaCasse

The University of Wisconsin – Milwaukee, 2019
Under the Supervision of Professor Wilkistar Otieno

The current big data landscape is one such that the technology and capability to capture and storage of data has preceded and outpaced the corresponding capability to analyze and interpret it. This has led naturally to the development of elegant and powerful algorithms for data mining, machine learning, and artificial intelligence to harness the potential of the big data environment. A competing reality, however, is that limitations exist in how and to what extent human beings can process complex information. The convergence of these realities is a tension between the technical sophistication or elegance of a solution and its transparency or interpretability by the human data scientist or decision maker. This dissertation, contextualized in the connected manufacturing enterprise, presents an original Fuzzy Approach to Feature Reduction and Prioritization (FAFRAP) approach that is designed to assist the data scientist in filtering and prioritizing data for inclusion in supervised machine learning models. A set of sequential filters reduces the initial set of independent variables, and a fuzzy inference system outputs a crisp numeric value associated with each feature to rank order and prioritize for inclusion in model training. Additionally, the fuzzy inference system outputs a descriptive label to assist in the interpretation of the feature's usefulness with respect to the problem of interest. Model testing is performed using three publicly available datasets from an online machine learning data

repository and later applied to a case study in electronic assembly manufacture. Consistency of model results is experimentally verified using Fisher's Exact Test, and results of filtered models are compared to results obtained by the unfiltered sets of features using a proposed novel metric of performance-size ratio (PSR).

Soli Deo Gloria

To my Creator, wife, children, and family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
AOI	Automated Optical Inspection
AP	Affinity Propagation
BGA	Ball Grid Array
BN	Bayesian Network
BNN	Back-propagation Neural Network
CE	Connected Enterprise
CNN	Condensed Nearest Neighbor
CPS	Cyber-Physical System
DBL-DL	Deep Belief Learning-Based Deep Learning
DBN	Deep Belief Network
DCS	Distributed Control Systems
DNN	Deep Neural Network
DROP	Decremental Reduction by Ordered Pair
DST	Dempster-Shafer Theory
ENN	Edited Nearest Neighbor
ET	Equipment Tracking
FA	Final Assembly
FAFRAP	Fuzzy Approach to Feature Reduction and Prioritization
FDC	Fault Detection and Classification
FDR	False Discovery Rate
FFT	Fast Fourier Transform
FIS	Fuzzy Inference System
GE	General Electric
ICT	Iterative Case Filtering
ICT	In-circuit Testing
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things

IoT	Internet of Things
IT	Information Technology
KNN	K-Nearest Neighbors
KPI	Key Performance Indicator
M2M	Machine-to-Machine
MCC	Mobile Cloud Computing
MD	Mahalanobis Distance
MEC	Mobile Edge Computing
MLP	Multi-Layer Perceptron
mRMR	Minimum Redundancy Maximum Relevance
MSC	Mean Shift Clustering
NBC	Naïve Bayes Classifier
NP	Nondeterministic Polynomial Time
OT	Operational Technology
P&P	Pick and Place
PAC	Programmable Action Controller
PCA	Principal Component Analysis
PCB	Printed Circuit Board
PLC	Programmable Logic Controller
R2F	Run-to-Failure
RA	Rockwell Automation, Inc.
RBF	Radial Bias Function
RBM	Restricted Boltzmann Machine
SCADA	Supervisory Control and Data Acquisition
SMT	Surface Mount Technology
SOM	Self-Organizing Map
SPI	Solder Paste Inspection (machine)
STFT	Short Term Fourier Transform
SVM	Support Vector Machines
SVR	Support Vector Regression

TSFRESH	Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests
UML	Unified Modeling Language
UWM	University of Wisconsin – Milwaukee
WI	Wisconsin
VP	Vice President

ACKNOWLEDGEMENTS

Soli Deo gloria. Latin for “Glory to God alone”, this phrase comprises one of five central tenets, or *solae,* of the Protestant Reformation. For the Christian, there is no higher privilege than to ascribe glory, honor, and praise to God. This has been my objective throughout this doctoral program, and I hope that my internal motivations and external actions were pleasing to that end.

Related, there is arguably no greater danger to the Christian life than the sin of pride. Thankfully, there are so many people without whom I would not have attained this degree, that pride is not a sane response, and it is my great pleasure to acknowledge them. My fear is that my words will not sufficiently convey my gratitude or the degree to which I value the relationships that the past three years have afforded me.

First and foremost, thanks to my wife, Becky, for her love, support, and bearing of the disproportionate share of household and family duties for the past year. I love you and am forever thankful that God has allowed us to do life together. To my children, I love you and am sorry for the late work nights and missed bedtime prayers and songs. You are a constant source of joy to your mother and me, and I am excited for what the future holds for each of you.

It is also my great pleasure to acknowledge and thank my advisor, Dr. Wilkistar Otieno, for her mentorship and behind-the-scenes assistance throughout this process. I cannot thank you enough for your perseverance in facilitating the relationship with our industry partner, Rockwell Automation, and I count it a privilege to be professionally associated with you. You exhibit genuine interest and care for your students, and I will endeavor to emulate that in my own relationships with future students.

To the ladies and gentlemen who served as my dissertation committee, Dr. Naira Campbell-Kyureghyan, Dr. Francisco Maturana, Dr. Matthew Petering, and Dr. Bo Zhang: Thank you for sharing your schedules and for the continuous feedback and constructive criticism that helped shape this work. A special recognition is owed to Dr. Maturana, who was my primary working partner at Rockwell Automation and provided the conduit for access to the data and software tools necessary for this work. Francisco, I have tremendous admiration for your work ethic, knowledge, and pursuit of excellence. I am glad to have known you, and I hope that we can continue to work together in the future.

To the support staff in the College of Engineering and Applied Sciences, namely Betty, Douglas, and Wendy: Thank you for your collegiality and help in anything that I needed. You see students come and go, but, in every interaction, you treated me with your full attention and as if my concern was the most important thing you had on your plate at that time. Your efforts do not go unnoticed.

Thanks to the Data Analytics & Insight (DA&I) group, Rockwell Automation, for their support and for welcoming me as a temporary member of their team for the past year. To Sangeeta Edwin, Jay Schiele, and Troy Mahr: You have a first-class organization and it was a pleasure to be a peripheral member of it. To my pseudo-colleagues in Twinsburg, OH, Mikica Cvijetinovich and Gregory Vance: It was a pleasure working together and learning all about surface mount technology and printed circuit board manufacture. I hope that we can continue our association well into the future. There are more problems to solve.

Finally, thanks to the Racine Bible Church fellowship of brothers and sisters in Christ for your friendship, prayers, counsel, and support. You witnessed me when morale was high and when it was low. There are too many people to thank by name, but please be assured that Becky

and I love you and count it one of the great joys of our life to have shared corporate worship and Christian fellowship with you for these past years.

CHAPTER 1: INTRODUCTION

1.1 The Connected Enterprise

Rockwell Automation, a global manufacturing and consultation corporation headquartered in Milwaukee, WI, employs a term, The Connected Enterprise (CE), to describe its strategy of corporate shared vision for the future of industrial automation [1]. CE strategies address the problem of disconnect by linking people, equipment, and processes for real-time learning of enterprise status in order to enable informed, adaptive, and proactive decisions [2].

The term "big data" may be loosely defined as information such that the size, structure, or variety strain the capability of traditional software or database software tools to capture, store, manage, and analyze it [3], [4]. Not only does big data pose a challenge to software systems and tools, but the volume of data also challenges the ability of human operators, analysts, and leaders to grasp, consume, and understand the critical pieces. Research in fields as diverse as psychology [5], economics [6], and literature [7] have identified limitations in human ability to process, visualize, and synthesize meaning from data. George A. Miller observes that, in a study for which subjects were asked to quickly count the number of dots as they were flashed on a screen, the subjects' performance on fewer than seven dots so starkly contrasted with performance on more than seven dots that it was given a special name. With fewer than seven dots, subjects "subitized" whereas with above seven dots, subjects "estimated"[5], [8]. Herbert A. Simon observes that information requires attention, and as information increases it necessarily requires the consumer of that information to engage in prioritization, ranking, or filtering in order to digest it. He also observed that human beings are essentially "serial" creatures, capable of

focused attention on only one thing at a time [6]. Finally, the iconic Mortimer J. Adler, in the second edition of *How to Read a Book*, laments the evolution of a society with vast quantities of knowledge but little depth of understanding [7]. It is interesting to note that these gentlemen wrote their pieces in 1956, 1971, and 1970, respectively. The exploding volume of data captured by modern systems exacerbates this challenge to a degree likely not conceived of at that time [9].

From the definition of big data and the strategies for CE, it is clear that big data can be seen both as an enabler of the CE as well as an impediment. It is an impediment in that, by definition, it requires innovation and effort to truly harness; it is an enabler in that the successful capture, consumption, and application of big data are precisely the necessary prerequisites to harness the power of the CE to its full potential. The balance between big data as a challenge versus an enabler is illustrated by the "less is more" or "more is less" paradigm. In one sense, the goal is to extrapolate trends from a sample of data, with the smaller the sample the better. Too much data can run the risk of overfitting a model. On the other hand, reducing the data down to a small sample is only a good idea if there is a clear sense of what to be looking for. In the quest for latent factors of interest, this strategy may not be advisable. Paradoxically, it may require vast quantities of data in order to train models to identify the narrow sliver of highly valuable information [10]. This is particularly true for highly unbalanced datasets such as defects in a mature manufacturing process. If a small number of key features can be identified to capture the critical information for the decision of interest, then monitoring those key features in the steady state can address the large volume of data [11].

The purpose of this document is to describe and explain a proposed hierarchical data filtration approach for specific application to the connected manufacturing enterprise but general application to any big data context. Chapter 2 contains literature review. Chapter 3 describes

research objectives and proposed methodology to attain those objectives. Chapter 4 provides background on the concepts underlying the framework. Chapter 5 provides a detailed description of the proposed framework. Chapter 6 contains application to three diverse datasets. Chapter 7 contains an applied case study in the CE, and Chapter 8 provides concluding remarks and directions for future research. This framework contributes to the big data problem by providing a mechanism to identify value-added features for the problem of interest and, most importantly, quantify the usefulness of those features. At a high level, the framework reduces the number of features to consider by applying a series of filters that first identify poor features to discard and then highlight quality features to retain.

1.2 Trends and Terminology in Manufacturing and Industry

In exploring recent advancements in manufacturing and industry, interested practitioners and researchers might find themselves deciphering a series of seemingly related, sometimes interchangeable, but distinct terms that mean different things to different parties in different contexts. In some cases, specific terminology might be used in one part of the world whereas another term is employed elsewhere. Seven terms, in particular, are of interest: The Connected Enterprise, Industry 4.0 (Germany), Industrial Internet of Things (IIoT), Smart Manufacturing, China Manufacturing 2025 (China), Manufacturing Innovation 3.0 (S. Korea), Usine du Futur (France). Overall, these strategies tend to focus on fostering industry agility to increase productivity, increase safety, reduce risk, reduce time to market, lower cost of resource utilization, optimize asset management and drive toward a customer-centric organizational culture.

“Industry 4.0” , a term most commonly used in German-speaking contexts, follows logically from mechanization (“Industry 1.0”), electrical energy (“Industry 2.0”), and digitization

("Industry 3.0"), and has been referred to as a fourth industrial revolution [12]. It therefore refers to a range of topics exhibiting common fundamental concepts, which may include [12]:

- Smart Manufacturing
- Cyber-physical systems
- Self-organization
- New systems for distribution, procurement, and development of products and services
- Adaptability
- Corporate social responsibility

The IIoT, a term initially coined by General Electric (GE) in 2012, refers to a network of industry devices connected by communications technologies for the purposes of monitoring, collection, exchange, analysis, and delivery of insights to drive smarter, faster business decisions [13]. The IIoT is typically used as a generic term, whereas Industry 4.0 is a specific, conceptually different paradigm for manufacturing [14]. The vision for Industry 4.0 is one of global networks that connect machinery, factories, and warehousing facilities and cyber-physical systems to connect, control, and share information to enhance decision-making [14]. IIoT is a subset of Internet of Things (IoT), a broad term for connected devices. IoT be can subdivided into Consumer IoT and IIoT. Examples of Consumer IoT would be household appliances interconnected via smart phone app for diagnostic monitoring or control.

IIoT has potential for greater impact than Consumer IoT because it carries the potential to bring entirely new infrastructures to the most critical and impactful societal systems [15]. Examples of consortia of companies targeting the IIoT include the Industrial Internet Consortium (IIC) [16], Industry 4.0, and the OpenFog Consortium [17].

Smart Manufacturing is a general term for the use of sensors and wireless technologies to capture data in all stages of production or product lifecycle. Examples include vehicle engines collecting and transmitting diagnostic information or optical scanners detecting defects in printed circuits [18].

The terms discussed are interrelated but not synonymous. Smart manufacturing is a component of Industry 4.0 and the Connected Enterprise. Similar technological enablers such as sensor technology and wireless networks enable the IIoT. Industry 4.0 is a specific, applied paradigm of the IIoT, and the Connected Enterprise is an enterprise-wide extension of Industry 4.0.

The final term to discuss is the CE, upon which this dissertation is grounded. CE is a term coined by Rockwell Automation [1] to describe a strategy of corporate shared vision for the future of industrial automation. As Lyman Tschanz, Rockwell Automation VP of Connected Enterprise External Affairs Operations in an interview in May 2018, "The CE strategy was an organic development of the industrial controls evolution." Following the discovery of electricity, relays in combination with other electro mechanical provided the means for industrial process control. This gave way to programmable logic controllers (PLCs) and Programmable Automation Controllers (PACs), which enabled reliable digitized control of manufacturing processes and fault diagnostics mostly within a single machine. Subsequently, network architectures (hardware and software) enabled machine to machine (M2M) interaction though communication of several PLCs without direct human intervention. Modern industries have distributed control systems (DCS) and Supervisory Control and Data Acquisition (SCADA) architectures that enable further integration of information and operation technologies (IT & OT). This integration enables the interaction between manufacturing process data, people, and

the business enterprise to optimize the Key Performance Indicators of at organization at all levels of the organization: factory level, enterprise level and global supply chain level. Figure 1 is a pictorial representation of the CE strategy that enables the seamless convergence of the information and operation functions of an enterprise.

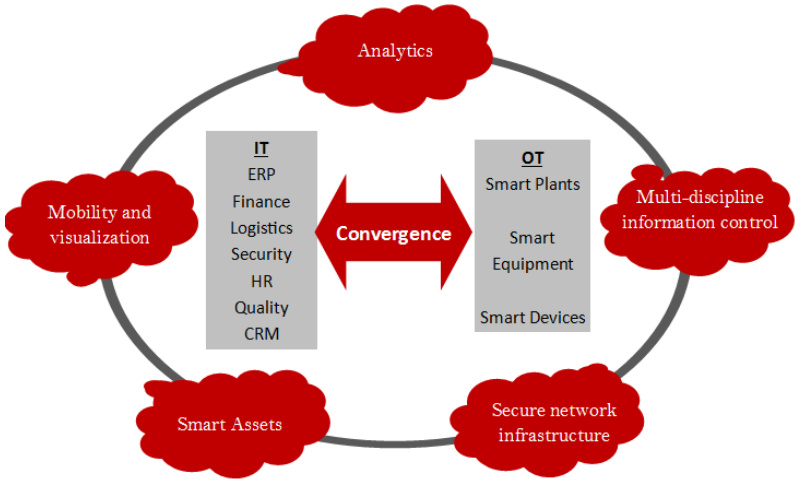


Figure 1: The Connected Enterprise Strategy - Enabling IT-OT Convergence

The Connected Enterprise therefore, is one of: [2]:

- enterprise-wide visibility and collaboration
- interconnected people, equipment, and processes
- real-time learning of enterprise status
- organizational agility by means of increased information to make informed, adaptive, proactive decisions

A true Connected Enterprise requires three components: network infrastructure, working data capital, and security. Network infrastructure links manufacturing and operations with the larger enterprise; working data capital refers to information that is gathered across the spectrum of operations and is distributed to allow employees to do their jobs better; and security refers to

the culture of threat management policy and practice that is necessary for the networked environment [19].

This research concentrates on a specific key enabler to the Connected Enterprise in manufacturing: big data. Especially relevant enabling technologies for information capture, storage, analysis, and sharing in the Connected Enterprise, specifically edge analytics and cloud computing, will be the focus. Edge analytics emphasizes automated data acquisition and analysis at the end device level, primarily to sustain and preserve process privacy, reduce latency and ensure a reliable and robust system that can withstand connection outages. Edge-level analytics looks at device-level data comprising sensor, PLC, and drive to perform near real-time analysis and data synthesis. Edge-level analytics would happen in a multi-tier framework on premise, where different levels of data are aggregated and mixed to enable immediate analysis that can be used to close the loop with the machine or process under control. A benefit of this edge computing is a more robust system that can continue to perform calculations while withstanding connection outages. Cloud computing on the other hand, moves software applications to an off-premise or off-site data center and frees the IT footprint to reduce maintenance and energy costs. Additional benefits of cloud computing include reduction of overall IT costs associated with managing and maintaining IT systems, scalability, business continuity, and collaborative efficiency [20]. Other enablers of information management, which will not be the focus of this dissertation, include mobility, which allows users to access manufacturing data on a variety of smart devices and not be tied down to physical location, and virtualization and digital twinning, which untether hardware from its operating system and reduce dependency on physical servers [19], [21].

CHAPTER 2: LITERATURE REVIEW

This chapter contains content extracted from, “A Survey of Feature Set Reduction Approaches for Predictive Analytics in the Connected Manufacturing Enterprise”, a systematic review that was submitted for publication in the Special Issue *New Industry 4.0 Advances in Industrial IoT and Visual Computing for Manufacturing Processes* of *Applied Sciences*. The manuscript was submitted in 9 January 2019, accepted for publication on 22 February 2019, and published on 27 February 2019 [22].

2.1 Purpose

The purpose of this literature review is to survey and discuss representative examples along the spectrum of existing research into a specific key enabler to the Connected Enterprise in manufacturing: big data. A good working definition of a “big data” environment is one such that the size, structure, or variety of information strains the capability of traditional software or database tools to capture, store, manage, and analyze it [3], [4]. Big data is clearly both an enabler to the CE and an obstacle. It is an obstacle in that, by definition, it requires innovation to truly harness; it is an enabler in that it is precisely the availability of vast untapped data that undergirds the enormous potential of the CE. Bollier (2010) explores this duality in big data for The Aspen Institute in [10].

Three related factors provide the motivation for this research. First, the rapid advent of technology to capture and store manufacturing data without the parallel development of corresponding analytical capabilities has resulted in the circumstance by which vast quantities of data are collected but not effectively analyzed or interpreted [18]. Second, the dual nature of big data as both an enabler and an obstacle to the CE perpetuates the state of affairs by which

manufacturers do not have a clear picture as to what manufacturing data, of the vast volumes collected, is truly valuable versus what can be discarded. This lack of clarity is due to disjoint or "siloed" data analytics capabilities within the organization [23] and by "where-to-start" paralysis brought on by the sheer volume of data and underdeveloped capability to visualize it [24]. Finally, it has been well established that limitations exist in how and to what extent the human analyst can process complex information [5], [6], [8]. The astounding development of the capabilities of automated analytical and artificial intelligence tools prompts interest in steps that can be taken to make them more palatable and digestible to human analysts and decision makers.

The objective of this literature survey is to determine whether, among the extensive body of knowledge on big data in the manufacturing environment, there is room for research into mechanisms or frameworks for feature filtration and prioritization when building applied machine learning models for predictive analytics. The interest is not so much in technology or architecture, which has been explored elsewhere ([25]–[28]), but rather in the human factor and how enablers to individual competencies can address the enterprise-level motivations for this research. It is true that certain machine learning algorithms can accommodate high dimensionality in input data, at the risk of potential issues such as overfitting. However, simply incorporating every potential feature into the model because it can algorithmically handle the calculations can shortchange the organization out of potentially useful information about the data at its disposal. The optimal subset problem is NP-Hard, which makes it impractical to iterate through all possible subsets of features to find the best subset for model training. For this reason, there is practical benefit in identifying how analysts and data scientists in manufacturing organizations decide how to select features for model inclusion and if that process is algorithmic and generalizable or if it is ad hoc, tailored to the specific problem of interest.

The remainder of the chapter breaks down as follows. Section 2.2 provides the methodology employed in identifying which articles to review and how to group them. Section 2.3 contains the body of reviewed literature, with each article receiving short commentary and consolidated observations and commentary following each sub-section. Section 2.4 provides overall observations from the body of reviewed literature, and Section 2.5 draws conclusions.

2.2 Methodology

2.2.1 Data Collection

Articles cited in this review can be broadly categorized into two groups. The first group consists of featured articles that receive analysis and discussion as pertaining to the motivations and / or objective of this research. The second group consists of background or supporting work that provides context to the introduction, justification to the motivations, or theoretical foundations to techniques or algorithms referenced in the first group.

The process of identifying articles for the first group began with broad queries into databases of scholarly literature using a series of topically relevant keywords. The following keywords were used, typically in pairs but sometimes in groups of three or more: [“big data”], [“smart manufacturing”], [manufacturing], [“machine learning”], [deep learning], [“deep learning”], [“fault detection”], [“fault prediction”], [“fault diagnosis”], [“data reduction”], [“feature selection”], [“feature reduction”], [“instance selection”], and [“instance reduction”]. Quotation marks indicate that the phrase was searched in its entirety. Thus, the keyword [“deep learning”] would not return the phrase “deep neural network learning” but the keyword [deep learning] would.

Academic or scholarly databases searched include ScienceDirect, IEEE, Taylor & Francis, SpringerLink, Google Scholar, and the University of Wisconsin – Milwaukee (UWM) library system. Time parameters were set for 2008 through 2018.

The keywords employed in database searches were selected to initially catch a wide scope of articles and then converge towards articles focusing more directly on the motivations and objective of the review.

A second means of identifying articles was to survey citations in articles identified in the database searches. For example, if a database search identified a survey paper on the use of machine learning for smart manufacturing, it would be possible that the articles cited therein might pose some relevance. The intent is not to duplicate work but rather to complement it. Returning to the previous example, a list of articles analyzed from an algorithmic perspective on which machine learning technique was employed could be relevant to this review by seeing how those same papers approach the human dimension of the project.

To identify citations in the second category of articles, the process was ad hoc and tailored to the specific algorithm or technique that warranted additional background. This category made no restrictions to time window because many techniques employed today have their theoretical foundations in decades past. For example, much of the initial, exploratory research into human limitations in processing information took place decades ago.

2.2.2 Data Analysis

The first layer of analysis consisted of broadly categorizing or organizing the reviewed articles. In keeping with the general search methodology, this resulted in three general groups,

selected for their intuitive sense in logically flowing from a broad, high-level search and then converging on the motivations and objective for the review.

- The first category explores big-data models for general industrial applications, specifically those featuring machine learning or deep learning.
- The second category focuses specifically on big data analyses and frameworks as applied to scenarios specific to smart manufacturing. Two sub-topics emerged in the search results: fault detection and fault prediction.
- The third category addresses data reduction tools and techniques.

The three categories listed above came about partly by design and partly post hoc. From the beginning, the question of interest was data reduction, specifically feature filtration and prioritization. Upon conducting a high-level analysis of articles captured by queries described in Section 2.2.1, it became clear that it would be appropriate to organize by papers explicitly focused on data reduction and those not. Clearly, a paper that is explicitly on the topic of data reduction will cover the subject. However, this review is also interested in how articles approach the topic of data reduction as a step contained within some problem of interest, when the paper is not explicitly about data reduction. This would have resulted in two categories. It subsequently became clear upon examination that, of the papers not explicitly focused on data reduction, they could be subdivided into those focused on a specific manufacturing application and those focused on general applications independent of a specific problem type. This yielded the three categories that ultimately form the organization of Section 2.3.

The second layer of analysis consisted of identifying which articles merit discussion and how to organize that discussion.

The predominant theme for analyzing articles in the first category, general industrial applications, was the degree to which the article focused on enterprise capabilities that enable organizational competencies versus approaches or methodologies that relate to human competencies. The first two motivations for this research are predominantly organizational competencies that are developed by a combination of high-level, enterprise capabilities and low-level, individual competencies. Of interest to this review was whether the reviewed articles gave treatment to the research motivations and, if so, whether that treatment focused on the organizational or the individual competencies.

The focus for analyzing articles in the second category, specific manufacturing applications, was the extent to which data reduction was explicitly performed and, if so, the extent to which that reduction step received treatment in terms of analysis or generalizability. The working hypothesis was that most research would be focused on a specific application or problem of interest, with the input data treated in secondary fashion, being a means to some end and not as potentially an end unto itself. The reasoning behind the working hypothesis is that practitioners and researchers alike have priorities of work; solving the problem of interest is typically Priority #1. Time-constrained efforts to complete the task at hand can sometimes cause both researchers and practitioners alike to miss valuable nuggets of insight that could provide useful in subsequent future work.

The focus for the third category was the context for the data reduction and the type of data reduction performed. If the context was outside of the manufacturing realm, the question was if it would be possible to extend the technique to manufacturing contexts. If already contextualized within manufacturing, the question was how generalizable it might be to other contexts or if the technique was unique to the specific scenario or case study.

Within each section, individual articles receive commentary in isolation. Each section concludes with observations and discussion on themes contained in more than one paper therein. Finally, Section 2.4 provides consolidated observations and discussion for the entire set of reviewed literature, providing a lead to the problem definition and research objectives in Chapter 3.

2.3 Literature Survey

2.3.1 Big Data Approaches for General Industrial Applications

Existing research into big data utilization for general industrial applications may be broadly generalized to contain valuable work and insight into the state of technology, current challenges, and methodologies or high-level frameworks for big-data analytical projects. The following section contains examples that, while not intended to be exhaustive, are representative of the body of literature on the subject. These examples are reviewed with specific interest in how they treat the research motivations from the human versus the architectural or technological dimension.

Wuest et al. (2016) present an overview of machine learning in manufacturing, focusing specifically on advantages, challenges, and applications [29]. Of particular interest is a summary of several recent studies ([30]–[33]) on the key challenges currently faced by the larger global manufacturing industry, with agreement on the following key challenges:

- Adoption of advanced manufacturing technologies
- Growing importance of manufacturing of high value-added products
- Utilizing advanced knowledge, information management, and AI systems
- Sustainable manufacturing (processes) and products
- Agile and flexible enterprise capabilities and supply chains

- Innovation in products, services, and processes
- Close collaboration between industry and research to adopt new technologies
- New manufacturing paradigms.

It is interesting to observe, in addition to what is listed, what is not listed. Specifically, these recent studies did not identify data utility as a key challenge. In other words, there is recognition that voluminous manufacturing data is collected. However, there is not universal agreement that this is a problem that needs to be addressed on the front end [29]; rather, employment of various machine learning techniques is proposed as a means to deal with data [34], with methods towards this end dating as far back as the 1970s [35].

Additionally, a summary of various machine learning techniques is offered, organized by their suitability for various manufacturing applications. Already discussed is the dimensionality problem, or the ability to handle data sets with a large number of features. However, employment of machine learning algorithms to deal with the problem of high dimensionality can lead the analyst directly into one of the main challenges associated with machine learning that the paper identifies, which is that interpretation of results can be difficult. Especially when the model is intended to support real-time monitoring of parameters with respect to proximity to some threshold, the practical usefulness of the model is diminished when large numbers of irrelevant or redundant features are input into the model simply because the machine learning algorithm can accommodate them.

Alpaydin (2014) provides a comprehensive overview of machine learning, with specific techniques that apply to each of the needs described above [36]. It is pointed out, however, that existing applications of machine learning tend to narrowly focus on the problem at hand or on a specific process [37] and not holistically on the manufacturing enterprise or on generalizing the

results to other processes. This observation is noteworthy, as it relates tangentially to the purpose of this literature review. One reason for the willingness to select machine learning algorithms that can handle high dimensionality may be a 'prisoner-of-the-moment' mentality. Analysts and data scientists perform real-world analyses to solve real-world problems, usually on a deadline imposed beyond their control. That deadline may be imposed by supervisors or it may be a function of outside constraints. Circumstances may not afford the luxury to step back, after completing the initial project, and thoroughly comb through the data to draw secondary conclusions about the nature of the input data. Rather, it is on to the next problem.

Wang et al. (2018) unpack the benefits and applications of deep learning for smart manufacturing, identifying benefits that include new visibility into operations for decision-makers and the availability of real-time performance measures and costs [38]. The authors provide, in addition to this practical information, a useful discussion on deep learning as a big-data analytical tool. In particular, they compare deep learning with traditional machine learning and offer three key distinctions between the two. Those distinctions are summarized in Table 1.

Table 1: Distinction between Traditional Machine Learning and Deep Learning

	Feature Learning	**Model construction**	**Model training**
Traditional Machine Learning	Features are identified, engineered, and extracted manually through domain expert knowledge.	Models typically have shallow structures (few hidden layers) and are data-driven using selected features.	Modules are trained step by step.
Deep Learning	Features are learned by transforming the data into abstract representations.	Models are end-to-end, high hierarchies with nonlinear combinations of numerous hidden layers.	Model parameters are trained jointly.

Note the distinction in feature learning. Deep learning models do not explicitly engineer and extract features. Rather, they are learned abstractly. This is both an advantage and a tradeoff.

The blessing is that model performance is typically superior. The tradeoff is in the transparency, traceability, and front-end verifiability of results.

The authors make an interesting observation, in that deep learning has shown itself to be most effective when it is applied to limited types of data and well-defined tasks [38]. This is notable in that conventional wisdom sometimes holds more data is better. Reducing the large data set to the most relevant subset of predictors may actually improve performance. This speaks directly to the motivation for this review and demonstrates the importance of the question. Not only does the capability to reduce a feature set to only the most relevant features enable an organization to build and increase institutional knowledge about the data at its disposal, but it also may lead to superior model performance.

Closely related, Tao et al. (2018) provide a comprehensive look at data-driven smart manufacturing, providing a historical perspective on the evolution of manufacturing data, a development perspective on the lifecycle of big manufacturing data, and a framework envisioning the future of data in manufacturing [39].

The broad framework outlined by Tao et al. (2018) contains four modules:

1. Manufacturing module. The manufacturing module contains conventional manufacturing activities, with the inputs being raw materials and the outputs being the finished product. During this stage, data is collected from a variety of sources that may include manual collection (human operators), production equipment, information systems, or industrial networks.
2. Data driver module. This module takes place in cloud-based data centers. Inputs are the data from the manufacturing module, and outputs from the analysis are explicit information and actionable recommendations for the diverse array of actions such as product design, production planning, and manufacturing execution that take place in the manufacturing module.

3. Real-time monitoring module. The real-time monitoring module performs precisely what its name implies, with the larger purpose being the assurance of process and product integrity. The data driver module is what enables this module to analyze the real-time running status of manufacturing facilities and allows leaders to make adjustments to the manufacturing process as needed in response to machine idleness or product quality defects.
4. Problem processing module. This final module exists to harness data and advanced analytical techniques to identify and predict emerging faults, diagnose root causes, highlight possible solutions, compute key performance indicators or metrics, perform what-if analysis, and other related functions that transform an operation from reactive to proactive or predictive. This module is grounded in data from both the real-time monitoring module and the data drive module.

It should be noted that this framework is general, not tailored to any specific manufacturing type. It is arguable that this is extendable to any industry or industry group that engages in data-driven decision-making and finds itself in the position to capture big data or metadata that it previously did not have at its disposal. The four modules do not operate in series but rather concurrently, with the utility from the problem processing module likely varying the most drastically when applied from one industry to another. The more resources that an organization can dedicate to the activities of anticipating and solving future problems, the better the results that the organization will reap from this framework.

An observation is that Tao et al. (2018) also identify a gap and promising future research direction that aligns indirectly with the focus of this literature review. Specifically, edge computing is identified as a promising direction for future research in incorporating into the proposed framework [39]. Edge computing is precisely one of the promising options in the problem of whittling down the volumes of production data into the core pieces that are truly meaningful and align with the key performance indicators (KPIs) of interest. Edge computing

allows data to be analyzed at the “edge” of a network before being sent to a data center or cloud [40]. A related term, fog computing, was introduced by Cisco systems in 2014 and extends the cloud to be closer to devices that produce and act on IIoT data [41]. The distinction between the two concepts, as well as other emerging paradigms such as Mobile Edge Computing (MEC) and Mobile Cloud Computing (MCC) are not fully mature and are subject to overlap [42]. The commonality is that they represent means for an organization to operationalize the individual competencies that are the focus of this review.

A slightly different application but keeping with the theme of harnessing the new technologies available in the big data and connected manufacturing environment, Andreadis (2015) proposes a framework for social media incorporation into the manufacturing process. In this framework, social network solicitations prompt feedback from subject matter experts in order to inform the product design process media in the form of live-streaming subject-matter expertise. Subsequently, social media in the form of live-streaming opinions and practical experiences from external experts informs process planning such as the selection of cutting conditions (a process in the case study) [43]. The point being to harness emerging technologies and using them to facilitate the exchange of ideas, opinions, and experiences between a virtually limitless pool of experts or consumers with respect to a specific product, thereby bringing in customer insight into the manufacturing process. For additional treatment of the use of technology to provide useful tools for the productive management of enterprise information, see [44]–[47].

A final framework for general industrial application of big data is presented by Flath and Stein (2017), specifically in the form of a data science “toolbox” for manufacturing prediction tasks. The objective is to bridge the gap between machine learning research and practical needs.

Feature engineering is identified as an important step that must take place prior to deriving useful patterns from the input data, and a case study employs Kullback-Leibler divergence to reduce 968 numeric features to 150 and 2140 categorical features to 27. The framework offered is a five-phase procedure [48].

- The first phase, data collection, is self-explanatory and may result in both structured and unstructured data; pre-processing or clean-up of data may be necessary.
- The second phase, exploratory data analysis, is necessary to identify properties of the dataset, eliminate duplicates, and further explore both the features and the processes involved. It may require employment of unsupervised machine learning techniques to identify the low-information features of the dataset.
- The third phase is to identify an appropriate metric for the specific problem at hand. It is certainly possible that this phase may be done out of sync, although doing so would prevent dataset characteristics such as data imbalance from being taken into consideration.
- The fourth phase is to select the learning algorithm, which will depend on the specific task (i.e. regression versus classification) and the metric(s) selected in the previous phase.
- The final phase is model execution, which may involve any number of steps to include model training, incorporation of new features, cross-validation, and interpretation of results.

The preceding literature, summarized in Table 2 below, shows high-level analysis of trends and challenges. It also provides examples of methodologies and frameworks for applied big data analytics in manufacturing.

A first observation is that there is not uniform agreement with regard to the question of dimensionality. On one extreme, the question is treated as a non-issue, to be handled by the machine learning algorithm selected. Other articles addressed the question at a high-level as

important but always within the context of the larger problem-solving approach and not to the level of detail that would be useful to the data scientist.

A second observation is that the approaches for predictive analytics in this section are geared less towards the detailed steps that an analyst might perform and more towards the infrastructure, architecture, and general data landscape that an organization should possess in order to have the capability to perform applied predictive analytics projects. This is not entirely unexpected, as the articles in this section are selected specifically for their high-level, broad outlook. The expectation is that articles in Section 2.3.2 and 2.3.3 will provide greater detail on the subject because articles reviewed in those sections focus more precisely on contexts that align better to activities at the level of the analyst or data scientist.

A third observation is gap identified by more than one researcher, which is the lack of holistic generalization of results beyond the specific, local problem under examination. This is related to manufacturers' limited knowledge of the relative utility or value contained among the different elements of the vast volumes of data that they collect in a somewhat mutually-reinforcing way. A lack of knowledge regarding the data landscape makes it difficult to generalize a dataset's utility from one application to the next. On the same token, not taking incremental steps to analyze projects after the fact for relevance and generalizability to other contexts perpetuates the deficiency in institutional knowledge.

Table 2: Summary - Big Data for General Industrial Applications

Author(s)	Focus
Wuest et al. [29]	Key challenges for global manufacturing industry
Alpaydin [36]	Machine learning overview
Wang et al. [38]	Deep learning for smart manufacturing
Tao et al. [39]	Data-driven smart manufacturing
Andreadis [43]	Social media and the manufacturing process
Flath and Stein [48]	Data science "toolbox" for industrial analytics

2.3.2 Big Data Frameworks for Specific Manufacturing Applications

This section moves from the higher level of general industrial or manufacturing applications to approaches geared towards specific smart manufacturing applications. The following literature instances fall into one of two sub-categories: fault detection and fault prediction. Fault detection and fault prediction are important areas of interest, and it is not surprising that predictive analytics projects gravitate to those topics. Predictive analytics in any the context will naturally gravitate to the dominant interests or challenges facing decision makers in that context, and, for manufacturers, KPIs associated with cost, quality, and time are negatively influenced by faults in machinery or output. Most manufacturing processes involve some form of creation or assembly at a given stage followed by some manner of inspection or validation before moving on to the next stage. Components are assembled into some final product, which itself undergoes functional testing prior to distribution to the customer. Machine downtime for unscheduled maintenance will negatively impact cycle time and, by extension, cost. Undetected malfunctions or nonconformities in machinery can lead to defective products escaping from one stage of manufacture to the next. There is an ever-present need to reduce defective products, which creates a natural partnership between smart manufacturing and predictive analytics. It is therefore unsurprising that much of the literature in predictive analytics in the manufacturing context will be applied to case studies in either fault detection or fault prediction.

It will be observed that different publications employ different frameworks, techniques, models, or methodologies to address specific manufacturing applications, often addressing specialized sub-problems or challenges. The focus in the ensuing section is how, from the human data scientist perspective, these analyses approach the challenge posed by big data. Is the big

data challenge one of an excessive number of diverse features that may contain hidden predictive potential? Is the challenge one of data volume, with exceedingly large numbers of records produced? Neither? Both? Additionally, this review will analyze the ensuing articles with an eye towards knowledge management, or the extent to which there is opportunity to generalize beyond the specific problem of interest.

2.3.2.1. Fault Detection

In [49], a MapReduce framework is proposed and applied to the fault diagnosis problem in cloud-based manufacturing under the circumstance of a heavily unbalanced dataset. An unbalanced dataset is one in which a large number of examples but another class is represented by comparatively far fewer [50], [51]. In terms of features for use in model training, each record of input data contains 27 independent variables and one fault type. There is no explicit discussion of reducing the 27 input variables to a smaller subset or what steps might be taken to do so for a scenario with higher dimensionality.

A hybridized CloudView framework is proposed in [52] for analyzing large quantities of machine maintenance data in a cloud computing environment. The hybridized framework contrasts with a global or offline approach [53] and a local or online approach [54], providing the advantage of being able to analyze sensor data in real-time while also predicting faults in machines using global information on previous faults from a large number of machines [52]. Feature selection is discussed at a high level, but the illustrative case study employs only three data inputs. The purpose of the case study is simply to illustrate the case-based reasoning applied and not apparently to address a specific situation.

In [55], Tamilselvan and Wang employ deep belief networks (DBN) for health state classification of manufacturing machines, with IIoT sensor data employed for model inputs. Specifically, data from seven different signals out of a possible 21 were selected for model training. Selection of which signals to include for model training was made based on literature and not on a specific methodological approach.

Deep belief networks are compared favorably to support vector machines (SVM), back-propagation neural networks (BNN), Mahalanobis distance (MD), and self-organizing maps (SOM) [55]. The deep belief network structure consists of a data layer, a network layer, and some number of hidden layers in between. This particular framework structures its hidden layers as a stacked network of restricted Boltzmann machines (RBMs) [56], with the hidden layer of the n^{th} RBM as the data layer of the $(n+1)^{th}$ RBM.

A similar machine learning methodology is employed by Jia et al. (2016) for fault characterization of a rotating machinery in an environment characterized by massive data using deep neural networks (DNNs) [57]. A DNN is similar to the DBN, except that the layers are not constrained to be RBM. For an extensive overview of deep learning in neural networks, see [58]. In a case study in fault diagnosis of rolling element bearings, a total of 2400 features are extracted from 200 signals using fast Fourier transform (FFT); no explicit reduction step is performed or discussed. Rather, the full dataset is input into the DNN.

The DNN model achieves impressive results when compared with a BNN, with correct classification rates over 99% compared to 65-80% for the BNN [57]. This indicates that the specific algorithm employed can have a non-trivial impact on the results, depending on the problem under study.

A framework for fault signal identification is proposed by Banerjee et al. (2010) in [59] using short term Fourier transforms (STFT) to separate the signal and SVM to classify it, and Banerjee and Das (2012) extend the approach in [60]. An explicit discussion on data preparation or feature filtration is absent due to the manageable feature set used for model training. However, the approach to extract features from signal data can lead to an excessive number of potential features, making such a step value-added.

Note also that this framework is a hybrid of several techniques, taking sensor data into the SVM after it has already been processed by signal processing and the time-based model. This is in contrast to frameworks relying exclusively on SVM [61], [62] or exclusively on time series analysis [63].

Probabilistic frameworks for fault diagnosis grounded in Bayesian networks (BN) and the more generalized Dempster-Shafer theory (DST) are examined in [64] and [65], respectively. For background and additional information on DST, see [66]. The challenge explored by Xiong et al. (2016) in [65] is that of conflicting evidence, with the observation that, in practice, sensors are often disturbed by various factors. This can result in a conflict in the obtained evidence, specifically in a discrepancy between the observed results and the results obtained by fusion through Dempster's combination rule. This challenge reveals the need to reprocess the evidence using some framework or methodology prior to fusing it. Xiong et al. (2016) propose to do so with an information fusion fault diagnosis method based on the static discounting factor, and a combination of K-nearest neighbors (KNN) and dimensionless indicators [65].

Just as in Jia et al. (2016), Xiong et al. (2016)'s method is applied to fault diagnosis among rotating machinery in a large-scale petrochemical enterprise.

Khakifirooz et al. (2017) employ Bayesian inference to mine semiconductor manufacturing data for the purposes of detecting underperforming tool-chamber at a given production time. The authors use Cohen's kappa coefficient to eliminate the influence of extraneous variables [67].

The tool-chamber problem examined in [67] is relevant to this review in that it employs a large number of binary input variables in its model, one for each tool and each step, equal to 1 if the tool-chamber feature was used in a step and equal to 0 if not. The feature filtration approach employed is a two-fold application of Cohen's kappa coefficient, once for pairwise comparison of the features against each other and once for features against the target. Features exhibiting high agreement with each other are wrapped with peers into a group; feature exhibiting low agreement with the target are removed from the model, with 0.20 as the threshold for removal.

This method is appropriate when features and the target are both binary; a limitation is the method is not suitable for data in other forms. This requires the target to be transformed from a continuous yield percentage to a categorical classification. A second possible limitation is that each variable is tested independently of the others, with no consideration for interaction. It is logically possible that a feature could have a poor Cohen's kappa coefficient but could interact with other features to produce an overall better model. An advantage of the approach, though not specifically discussed in the article, is that Cohen's kappa coefficient scores for each feature may be preserved from one analysis to the next and analyzed to see if they harbor latent relationships that might point to root causes of inadequate tool-chamber and not simply forecast it.

The final framework for fault detection that this literature review will explore is a cyber-physical system (CPS) architecture proposed by Lee (2017) for fault detection and classification (FDC) in manufacturing processes for vehicle high intensity discharge (HID) headlight and cable

modules [68]. For additional background and exploration of CPS, see [69]–[72]. Although much of the article is devoted to material outside the scope of this review, such as network and database architecture, the manufacturing process explored is notable because it involves multiple sub-processes, some of which are performed in-house and some of which are outsourced to external parties. Furthermore, although there is a small set of main defects that may be observed (shorted cable, cable damage, insufficient soldering, and bad marking), those faults are not directly traceable to a single sub-process. Rather, any number of different sub-processes may result in any fault type. The impact, when performing fault detection and classification, is that the cause-effect relationships and the backwards tracing of faults to diagnoses must take place beforehand.

The input data for the case study consists of eight signals, three from torque sensors and five from proximity sensors, and three learning models are explored: support vector regression (SVR), radial bias function (RBF), and deep belief learning-based deep learning (DBL-DL). In the SVR and RBM models, no additional step in data filtration or feature extraction is performed; in the DBL-DL model, features are extracted in the form of two hidden layers. Unsurprisingly, the DBL-DL model outperforms the other two, with a classification error rate of 7% as compared to 8% for SVR and 9% for RBM [68].

2.3.2.2. Fault Prediction

In [73], Wan et al. (2017) present a manufacturing big data approach to the active preventive maintenance problem, which includes a proposed system architecture, analysis of data collection methods, and cloud-level data processing. The paper mainly focuses on data processing in the cloud, with pseudocode provided for a real-time processing algorithm. Two types of active maintenance are proposed as necessary: a real-time component to facilitate

immediate responses to alarms and an offline component to analyze historic data to predict failures of equipment, workshops or factories.

Of interest to this review is to note that the aforementioned approach is in the context of an organization's ability to perform active preventive maintenance and not in the context of how a data scientist goes about performing his or her analysis. For example, 'data collection' in the context that Wan et al. (2017) describe refers to the required service-oriented architecture to integrate data from diverse sources. To the data scientist, 'data collection' is the employment of that architecture in identifying and obtaining specific data elements for model inclusion.

Munirathinam & Ramadoss (2014) apply big data predictive analytics to proactive semiconductor production equipment maintenance. Beginning with a review of maintenance strategies, the researchers present advantages and disadvantages for each of four different maintenance strategies: run to failure (R2F), preventive, predictive, and condition-based. Following this background, an approach for predictive maintenance is presented as follows [74]:

- Collect raw fault detection and classification (FDC), equipment tracking (ET), and metrology data
- Perform data reduction using a combination of Principal Component Analysis (PCA) and subject matter expertise. This step, in the semiconductor case study, reduces the set of possible parameters from over 1000 to precisely 16
- Train model
- Display output to dashboard with a Maintenance / No Maintenance status

Two immediate observations are apparent when considering the data reduction step employed in this model. First, the use of PCA is effective but it carries with it the loss of interpretability after the fact. This limits the options associated with the dashboards created for visualization of model results. If there were an alternative to PCA that retains interpretability, it may be possible to identify specific thresholds in the input data that are triggers for required maintenance and then track proximity to those thresholds in a dashboard. A second observation is that PCA requires linearity among the parameters because it relies on Pearson correlation coefficients. It also assumes that a feature's contribution to variance in the data relates directly to its predictive power [75]. It is not clear that this is always an appropriate assumption.

Ji and Wang (2017) present a big-data analytics-based fault prediction approach for shop floor scheduling. This application of the big data problem focuses less on the availability of machining resources and more on the problem of potential errors after scheduling [76]. Specifically, it is observed that task scheduling using traditional techniques considers currently available equipment, with time and cost saving as the main objectives. Missing from consideration is the condition prediction of the machines and their states. In other words, scheduling is made absent of any information on the expected condition of the machines during the production process. In the proposed framework, tasks are represented by a set of data attributes, which are then compared to fault patterns mined through big data analytics. This information is then used to assign a risk category to tasks based on generated probabilities. The model provides the opportunity for prediction of potential machine-related faults such as machine error, machine fault, or maintenance states based on scheduling patterns. This knowledge can lead to better machine utilization.

It should be noted that this particular framework, while creative, was not tested on actual data but rather on hypothetical datasets due to data proprietorship policy [76], hence providing clear opportunities for future research.

Artificial Neural Networks (ANN) are applied to recognize lubrication defects in a cold forging process, [77] predict ductile cast iron quality [78], optimize micro-milling parameters [79], predict flow behavior of aluminum alloys during hot compression [80], and predict dimensional error in precision machining [81]. Finally, a process approach is taken to improve reliability of high speed mould machining [82].

It was seen in the preceding models featuring ANN that data reduction plays a role of minimal importance because the neural network accomplishes feature creation and selection in the hidden layers. In [77], a total of 20 features are selected for model input with no explicit data reduction step. Nor was any reduction step performed in [78], where the dataset was relatively small, consisting of only 700 instances of 14 independent variables in the training set. In [79] and [80], only three features are input into the ANN. In [81], an extension of a simulation and process planning approach in [82], [83], the number of input variables is five.

An observation across the set of articles reviewed in Section 3.2 is that a specific data reduction step is rarely utilized, either because the feature set was small to begin with or because the machine learning technique could accommodate. The exceptions used either statistical measures (Cohen's kappa) or PCA to reduce the feature set. The article using the former technique did not report how many features the case study began with and how many were ultimately used for model training. It is, therefore, not clear the extent to which the technique is useful. In the case of PCA with subject matter expertise, a feature set of 1000 reduced to 16. Additional discussion and possible extension will be included in Section 4.

A second observation is that, as in Section 3.1, variation exists in the frame of reference for which different articles approach the topic of predictive analytics. Some articles focus on the organizational capability to perform predictive analytics. These incorporate robust discussion on technology-centric elements such as architecture for data capture, storage, and extraction or at which levels different analyses may be performed (cloud, edge, real-time, offline, etc.). These typically featured commercially available technologies such as Hadoop or MapReduce and address some of the prerequisites for building organizational competencies in this area. Other articles, on the other hand, employed the term 'framework' to refer to a problem-solving approach or methodology, a sequence of actions to be performed by the analyst or data scientist. These articles more directly align with the objective of this literature review, but it is important to distinguish between the two perspectives as each are important. Indeed, the organizational capability for data capture, storage, and migration must necessarily precede any in-house capability to analyze smart manufacturing data or use it to train a machine learning model.

Table 3 provides a summary of the foregoing studies that approach big data analytics applied to specific manufacturing use cases. The table summarizes whether the paper focuses on organizational capabilities, methodological approaches for the analyst, case studies, or some combination. For case studies, the machine learning algorithm used is listed.

Table 3: Big data approaches for specific industrial applications

Authors(s)	Focus	Explicit Data Reduction Step
Kumar et al. [49]	Enterprise-level architecture; methodology to address class imbalance	No
Bahga and Madiseti [52]	Enterprise-level architecture	No
Tamilselvan and Wang [55]	Case study: Machine health states – DBN	No
Jia et al. [57]	Case study: Fault characterization – DNN	No

Banerjee et al. [59]	Case study: Fault signal identification – SVM	Discussed, not implemented
Xiong et al. [65]	Methodology: Information fusion to reconcile conflicting evidence in fault detection	No
Khakifirooz et al. [67]	Case study: Yield enhancement – Bayesian inference	Yes
Lee [68]	Enterprise-level architecture; Case study: Fault detection and classification – SVR, RBF, DBL-DL	No
Wan et al. [73]	Enterprise-level architecture; Methodology: Real-time and offline components; Case study: Fault prediction – Neural Network	No
Munirathinam & Ramadoss [74]	Enterprise-level architecture	Yes
Ji and Wang [76]	Enterprise-level architecture; Simulated proof of concept case study: Fault prediction for shop floor scheduling	No
Rolfe et al. [77]	Case study: Lubrication defects in cold forging process – NN	No
Perzyk and Kochanski [78]	Ductile cast iron quality - NN	No
Kilickap et al. [79]	Micro-milling parameter optimization - NN	No
Changqing et al. [80]	Alloy flow behavior - NN	No
Arnaiz-Gonzalez et al. [81]	Dimensional error in precision machining – NN	No

2.3.3 Data Reduction

The third and final category of literature that this review will examine focuses on techniques or approaches specifically for data reduction, which includes feature reduction/selection and instance reduction/selection. There exists a substantial body of influential data preprocessing algorithms for missing values imputation, noise filtering, dimensionality reduction, instance reduction, and treatment of data for imbalanced processing [84]. Specific algorithms for feature selection include *Las Vegas Filter/Wrapper* [85], *Mutual Information Feature Selection* [86], *Relief* [87], and *Minimum Redundancy Maximum Relevance (mRMR)* [88]. Specific algorithms for instance reduction include condensed nearest neighbor (CNN) [89],

edited nearest neighbor (ENN) [90], decremental reduction by ordered projections (DROP) [91], and iterative case filtering (ICF) [92].

The interest in the ensuing articles reviewed in this section is in their suitability for application to the CE. To this end, the domain in which the articles implement any applied case studies is also examined. It will be observed that the reviewed articles contain tasks that fall within Step 3 of the data source selection methodology outlined in [93] and broadly fall into one of three categories: sampling reduction, feature reduction, or instance reduction. Sampling reduction applies to contexts such as optical inspection or reengineering, where there is a need to obtain information for an entire component or surface. If that information may be obtained using fewer samples, then benefits in cost or efficiency follow. Instance reduction applies to contexts in which large numbers of data points are collected for a relatively smaller set of attributes or features. Feature selection is the process of reducing the number of attributes or columns to be input into a machine learning model for training.

Habib ur Rehman, et al. (2016) propose an enterprise-level data reduction framework for value creation in sustainable enterprises, which, while not contextualized to manufacturing, is easily extendable to this domain. The framework considers a traditional five-layer architecture for big data systems and adds three data reduction layers [94].

The first layer for local data reduction is intended for use in mobile devices to collect, preprocess, analyze, and store knowledge patterns. This physical layer can easily be conceptually translated to the CE. The second layer, for collaborative data reduction, is situated prior to the cloud level, with edge computing servers executing analytics to identify knowledge patterns. Note that “edge computing” may be referred to as “fog computing” in some cases [95]. This step will exist in varying degrees in the CE depending on the maturity of the process or organization.

In the context of user Internet of Things (IoT) mobile data, as initially presented in the paper, there exists a body of data that must automatically be discarded in accordance with external constrains such as privacy laws. This brings a practical purpose to this initial filtration layer. In smart manufacturing, the physical layer represents IIoT machine or production data, all of which might theoretically harbor some purpose. It may not be prudent to automatically discard chunks of data until it has been definitively determined that there is little risk in doing so. Finally, a layer for remote data reduction is added to aggregate the knowledge patterns from edge servers that are then distributed to cloud data centers for data applications to access and further analyze [94].

It should be noted that this framework is at the institutional level and not at the level of the data scientist. The data reduction layers are presented as automated processes applied to the raw source data and not dependent on a specific project or problem of interest.

At the data scientist level, a second point-based data reduction scenario is presented in [96], in which Ma and Cripps (2011) develop a data reduction algorithm for 3D surface points for use in reverse engineering. In reverse engineering, data is captured from an existing surface on the order of millions of scanned points. There are challenges associated with volume of data, and there are challenges in the form of increased error associated with removing data. The data reduction algorithm is based on Hausdorff distance and works by first collecting a set of 3D point data from a surface using an optical device such as a laser scanner, iterating through the set of points, and determining if a point can be removed without causing the local estimation of surface characteristics to fall out of tolerance. This is done by comparing shape pre- and post-removal. The procedure is tested on an idealized aircraft wing but is extendable to any manner of reverse engineering that employs 3D measurement data. It is possible that this could also be extended to inspection-type applications, but the challenge is that the end-state number of

required data points will be dependent on the nature of the surface. Additionally, it is not certain that Hausdorff distance would be the appropriate metric for other contexts such as automated optical inspection.

Considering data reduction with respect to the set of features to be used for model training, Jeong et al. (2016) propose a feature selection approach based on simulated annealing and apply it to a case study for detecting denial of service attacks [97]. This approach is similar to [98], which uses the same data set but a different local search algorithm.

The model starts with a randomly-generated set of features to include, trains a model on that set of features, and tests it by way of some pre-designated machine learning technique. The case study is a classification problem, and so examples used include SVM, multi-layer perceptron (MLP), and naïve Bayes classifier (NBC). After obtaining a solution and measure of performance using some cost function, neighborhood solutions are obtained and tested. Superior solutions are retained, and inferior solutions are either discarded or retained based on a probability calculation. This ability to retain an inferior solution allows the simulated annealing algorithm to "jump" out of a local extrema [99]. The intrusion detection case study employed 41 factors, which reduced to 14, 16, and 19 factors when using MLP, SVM, and NBC respectively. A limitation to this approach is that it requires model training at every iteration of the simulated annealing. This may limit the options for which machine learning technique to select; preference should be given to algorithms that quickly converge. Again, for only 41 factors, this is less of an issue. If there are hundreds or thousands, then this approach may be impractical.

Lalehpour, Berry, and Barari (2017) propose an approach for data reduction for coordinate measurement of planar surfaces for the purposes of reducing the number of samples required to adequately validate that a part has been build according to design specifications

[100]. The larger context for this approach is manufacturing, but the applicability is narrowly scoped to an inspection station along an assembly line. Thus, this approach could be used in programming firmware for an optical inspection machine so that it can diagnose defective components as efficiently as possible. However, it would not be useful in performing root cause analysis to find the source of the defects or predict future occurrences.

Ul Haq, Wang, and Djurdjanovic (2016) develop a set of ten features that may be constructed from streaming signal data from semiconductor fabrication equipment. Technological developments allow the collection of inline data at ever increasing sampling rates. This has resulted in two effects, the first being an increase in the amount of data required to store, and the second being the ability to discern features that were previously not discernible [101]. Specifically, high sampling rates allow information to be gleaned from transient periods between steady signal states. This enables the extraction of features from the signal that could not be calculated with lower sampling rates.

The approach can be extended to any signal-style continuous data source from which samples are taken, although the implication is that the lower the sampling rate, the less likely that these new features will provide value. These constructed features are applied to case studies of tool and chamber matching and defect level prediction. A reasonable extension might be to apply the approach to machine diagnostic information for active preventive maintenance.

From a feature selection or dimensionality perspective, which is of most interest to this review, the ten features are calculated every time the signal transitions from one steady state to another. For relatively static signals, this will result in a manageable feature set; for more

dynamic signals or for large time windows, the number of calculated features may become prohibitively large. This could be alleviated by adding an additional layer of features that employ various means to aggregate the values of the ten calculated features over the entire span of time.

Continuing on the topic of feature selection, Christ, Kempa-Liehr, and Feindt (2016) propose an algorithm for time series feature extraction, TSFRESH, that not only generates features but also employs a feature importance filter to screen out irrelevant features [102]. This framework, illustrated in Figure 2, begins by extracting up to 794 predefined features from time series data. Subsequently, the vector representing each individual feature is independently tested for significance against the target. This produces a vector of p-values with the same cardinality as the number of features. Finally, the vector of p-values is evaluated to decide which features to keep. The method for evaluating the vector of p-values is to control the false discovery rate (FDR) using the Benjamini-Hochberg procedure [103]. A case study using data from the UCI Machine Learning Repository [104] reduced an initial set of 4764 features to 623 [105].

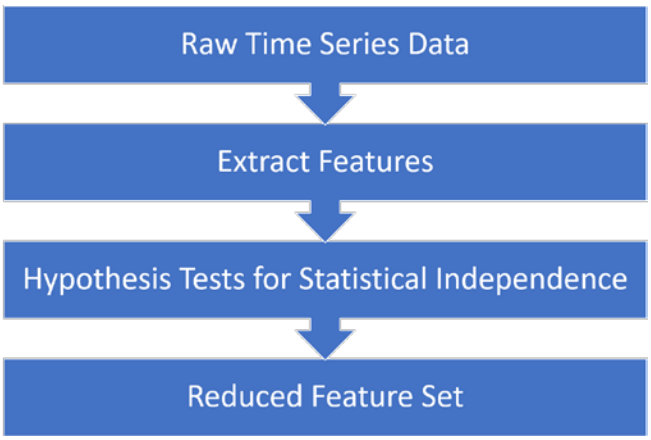


Figure 2: High-level TSFRESH process, adapted from [102]

For instance reduction, Wang et al. (2016) employ a framework based on two clustering algorithms, affinity propagation (AP) and KNN, to extract “exemplars”, or representations of some number of actual data points [106]. A clustering algorithm is employed to cluster the data instances into similar groups; an exemplar is then defined to represent the group. The context is in network security, specifically anomaly detection. The idea is that records for http traffic and network data under normal circumstances can be grouped or aggregated into representations of those conditions, which can produce cost savings in data storage. The technique is potentially extendable to other areas of manufacturing, although for mature processes there may not be the desire to perform aggregation of records because the “easy” relationships have already been discovered. Rather, a large number of records may be necessary to identify hidden structures or correlations in sub-groups that might otherwise, in smaller sample sizes, be considered outliers [107], [108].

A second instance reduction, by Nikolaidis, Goulermas, and Wu (2010), draws a distinction between instances close to class boundaries and instances farther away from class boundaries [109]. The reasoning is that instances farther away from class boundaries are less critical to the classification process and are therefore more “expendable” from an instance reduction standpoint. The four-step framework first uses a filtering component such as ENN to smooth class boundaries and then classifies instances as “border” and “non-border”. Following a pruning step for the border instances, the non-border instances are clustered using mean shift clustering (MSC).

Table 4: Frameworks for data reduction

Author(s)	Focus
Habib ur Rehman, et al. [94]	High level / Institutional framework
Jeong et al. [97]	Feature selection meta-heuristic (simulated annealing)

Lalehpour, Berry, and Barari [100]	Sample reduction
Ma and Cripps [96]	Shape preservation with data reduction for 3D surface points
Ul Haq, Wang, and Djurdjanovic [101]	Feature extraction from streaming signal data
Christ, Kempa-Liehr, and Feindt [102]	Feature extraction and selection from time series data
Wang et al. [106]	Clustering algorithms to extract representative data instances
Nikolaidis, Goulermas, and Wu [109]	Instance reduction based on distance from class boundaries

As previously indicated, the reviewed articles from the Section 2.3.3, summarized in Table 4, cover reductions in the number of samples required to obtain a satisfactory result, techniques to reduce the number of instances or records, and techniques to reduce the number of features or attributes.

Of greatest interest to this review is the second category, feature selection, and two approaches seen in this section merit further discussion in relation to each other. The first approach, the TSFRESH approach, generates a list of up to 794 features from a single time series and, using statistical independence as the test, reduces the feature set by eliminating the features that do not exhibit a significant statistical dependence with the response. Using this approach, a model with N time series inputs would have 794N features extracted by TSFRESH. Even if TSFRESH then filters out 50% of the features, there still could remain many hundreds of features in the model. This could be an excessive number of features that strains the capacity of the analyst to truly grasp what is going on or pinpoint the critical relationship(s) of interest. Extending the approach to include subsequent filter(s) could be a step in remedying this challenge.

The second approach of interest is the use of optimization heuristics to obtain a near-optimal subset of features for the problem at hand. It might be a reasonable extension to

TSFRESH to incorporate a second filter that seeks to better optimize the feature set with respect to the objective function, possibly using a heuristic such as simulated annealing. This would also add the dimension of feature interaction, which is currently not present in the TSFRESH statistical independence filter.

A final observation from the third category of reviewed literature is that the set of literature on reducing or filtering the features that might go into a machine learning model is reasonably robust but is relatively less so concerning the prioritization of the remaining features. This implies a gap in terms of approaches to quantitatively or qualitatively stack features against each other. An alternative explanation is that such approaches exist but were simply not employed in the reviewed literature. This seems unlikely, as, the benefit of such capability would be to see how a particular feature of interest fares in its utility from one problem to the next. In smart manufacturing, the same features of data are continually collected and used repeatedly in different analyses. It may be of interest to know which of those features tend to be valuable in harboring predictive power and which ones tend not to.

2.4 Observations and Discussion

This chapter reviewed existing research into frameworks or approaches for big-data analytics as applied to three levels of projects, with increasing degrees of precision or detail. The first level reviewed was a high-level look at frameworks for general industrial applications. The second level focused specifically and local, lower-level smart manufacturing applications of fault detection and fault prediction. Finally, the third and most specialized level looked at approaches specifically oriented towards data reduction.

In each section, articles were discussed individually as they pertain to the motivation for this research and their applicability to the CE. At the end of each section, discussion followed to summarize any observations across the set of articles within the section and relate them to each other. The final level of discussion is to look at the full picture and identify any observations, trends, or commonalities that span the three levels.

The first observation is that there is a dichotomy in how the same verbiage can be applied to different contexts. Terms like ‘framework’, ‘big data’, and ‘predictive analytics’ in some cases are contextualized as architecture required to build organizational competencies and in other cases as approaches or methodologies to build individual competencies.

In the context of organizational competencies for big data analytics, much consideration is made as to the architecture for where the data exists, how it moves from one location to the next, and at which level or echelon the analysis takes place. In general, there is some layer or module at which the initial data is generated or collected. At that point, there are different options for what to do with the collected data. One option is to migrate the data to a cloud-based data center for analysis in a consolidated location. Another option is to perform data analysis at local nodes. Whether to analyze at the edge or in the cloud will typically be a function of resources and of the time window available to perform corrective action. Actions that require real-time processing for quick action might not be performed at the cloud level because, by the time data is captured, cleaned, pre-processed, and run through a model, the window to correct an identified fault may have already passed. On the other hand, if there is sufficient time between the data collection point and the decision point, such as a manufacturing process in which there might be a gap of hours or days between assembly line procedures and testing, then analysis at the cloud level might be suitable.

It should be noted that, from an organizational competency perspective, the infrastructure is a prerequisite to the development of individual competencies in the form of data scientist best practices. However, it is those data scientists' best practices that become contributing factors to other organizational competencies such as knowledge management and decisions on long term data retention. There is an iterative and cyclic relationship such that organizational competencies produce individual competencies which then build and reinforce other organizational competencies.

A second observation across the three sections of reviewed literature is that there was a conspicuous absence of any discussion of the generalization of results beyond the specific problem of interest. This is true on both the 'front' end and the 'back' end of the articles reviewed. In other words, upon conclusion of the experiment or analysis, there was no discussion in any reviewed article of knowledge management or steps to generalize results from an input data perspective. There was certainly discussion about future research opportunities in generalizing an overall approach or algorithm, but in no cases did that discussion manifest itself the form of practical reflection on a feature set's utility for the problem of interest and prospects for utility in other scenarios. Similarly, during model formulation, there was no discussion of institutional knowledge that might play a role in feature selection. Only one reviewed paper referenced a data screening decision that was made based on prior work. The context in that situation was 21 possible signals to use as model inputs, of which seven were selected based on reviewed literature.

This observation is not intended as a negative criticism of any past work. It is quite natural to expect that this might be the case because finite resources drive priorities, and in a fast-paced world there is often little time to breathe between the completion of one project and the

start of another. Given this reality, there appears to be value in anything that can facilitate the creation and preservation of institutional knowledge in this domain.

A third observation is that feature selection approaches in most cases were performed using a single technique at a single point in the model building process. Feature filtration using Kullback-Leibler divergence reduced features sets of 1460 and 1460 to 198 and 175, respectively. Feature filtration using Cohen's kappa was stated as a step in one case study, but no results were provided as to how many features were filtered out. A combination of PCA and subject matter expertise reduced a feature set of 1000 to 16, although it was not clearly identified how many of those features were reduced from PCA and how many from subject matter expertise. In the case of TSFRESH, statistical hypothesis tests for independence filter out features that are statistically independent, reducing 4764 features to 623.

A natural next step for any of these techniques is to explore the possibility to layer one technique after another depending on how many features remain after a given filter. In the case of 1000 features reduced to 16, it is possible to successively iterate through all 65536 subsets of features to arrive at an optimal subset with minimal effort. In the case of 4764 features reduced to 623, however, this is computationally impractical. It is unlikely that the optimal subset of the 623 remaining features would be all 623 of those features; a layered approach to continue to weed out features would be a value-added step to analyses with large numbers of features remaining. This is especially true if there is the desire for the model and its results to be understandable and digestible on the human side of the enterprise. Furthermore, what is understandable and digestible to the data scientist may be neither understandable nor digestible to the decision maker. Communication and visualization are critical components to the human

element, particularly for decision makers who may not have background in the technical aspects of data science.

2.5 Conclusions and Proposed Contribution

The papers reviewed are not intended to represent an exhaustive list of all existing research on the subject. However, it is believed that the reviewed examples do provide a representative sample of the sort of research currently performed in this discipline.

A conclusion that may be drawn from the first general observation in Section 2.4 is that there is value in having a standard set of terminology when speaking about the big data environment in order to distinguish when one is referring to organizational capabilities or individual competencies. In the reviewed literature, terms like ‘framework’ or ‘data collection’ carried wide variance in their meaning depending on the context. It is likely that standard terms will be settled on over time, either bottom-up from common use or top-down from professional organizations in industry, academia, or government. At this point, it may suffice simply to be aware of the different contexts in which the topic may be broached. Attention to detail is always a good rule of thumb in any endeavor, and that may be a good temporary solution for now.

A second conclusion, following from the second general observation in Section 2.4, is that a generalized approach to provide clarity as to what input data is valuable and what input data is not valuable, perhaps with both a quantitative and qualitative dimension, can shape analysis decisions in the big-data environment. Those decisions might be localized to the problem of interest, as in deciding which features to include in the model. Those decisions might also extend to larger, resource-oriented decisions, such as start-up priorities for transitioning from a legacy manufacturing facility to a CE. From a knowledge management standpoint, there

is value in building institutional knowledge regarding features that perform poorly as well as features that perform well. Knowing which features tend to habitually appear in good solutions and which features habitually appear in bad solutions, if such knowledge exists, would be tremendously helpful in long term data capture and storage decisions.

Finally, the third general observation in Section 2.4 lends itself to the conclusion that there is room for additional research into practical means for feature filtration and prioritization. On the surface, there appears to be no reason why the single-layer filtration techniques employed in the reviewed articles cannot be extended into a series of hierarchical filters. One possible limitation would be that several of the techniques employ similarly-themed filters that may produce only limited improvement when performed in sequence. For example, filtering once by Cohen's kappa and then by testing for statistical independence might not produce substantial improvement. However, following the initial filtration by way of Cohen's kappa with an optimization heuristic such as simulated annealing or genetic algorithm to find an optimal or near-optimal subset of features might be a promising avenue to explore.

It is also worth exploring, from a knowledge management standpoint, feature reduction and selection techniques that preserve as much interpretability as possible. It has already been discussed that PCA is a common approach, but the reduction in dimensions from M to K, where $K < M$, will necessarily take away the physical meaning from those K features. Techniques in feature reduction that preserve the nature of the original features are a value-added contribution to this question.

The preceding observations and their corresponding conclusions provide the conceptual domain in which the research objectives in Chapter 3 will reside. The literature review has identified the opportunity for research into feature reduction and selection techniques that not

only retain interpretability to the greatest extent possible but also include a qualitative component that can be used to facilitate knowledge management beyond the local problem of interest. Such an approach would allow any machine learning project to serve a dual purpose, with the first purpose being to answer the immediate problem and the second purpose being to add to the organization's institutional knowledge regarding the data at its disposal.

CHAPTER 3: RESEARCH OBJECTIVES

Given the research gap identified in the preceding literature review, the following objectives are proposed for this dissertation.

Objective #1: Conceptualize and develop a high-level, hierarchical feature selection method

Discussion: The first step in a generalizable model that identifies which features are important and which are not is to create the conceptual architecture for precisely modeling how one feature will be scored or rated against another. This research selects two previously-reviewed works, [102] and [97], each employing a different discipline and theoretical underpinning, and integrates them in a way previously not seen in the body of literature. The contribution is twofold: the innovative application of existing techniques and the superior result that ensues.

Objective #2: Conceptualize and develop a Fuzzy Inference System (FIS) to quantify the relative value of features against each other and provide qualitative depth of knowledge regarding the individual features in the dataset

Discussion: The FIS developed in this research represents original content, from the selection of fuzzy input membership functions, rules, fuzzy output membership function, and linguistic labels applied to the output. The contribution is in the application of fuzzy systems to this specific scenario and in providing a crisp, quantified output to each feature that is not dependent on sequence as in techniques such as forward propagation or backward propagation.

Objective #3: Execute the hierarchical feature selection and labeling architecture on a diverse array of machine learning datasets.

Objective #4: Execute the hierarchical feature selection and labeling architecture on an applied case study in manufacturing using real-world, current smart manufacturing data.

Discussion: Objective 3 and Objective 4 are related but represent escalating levels of model validation and contribution. Objective 3 provides assurances as to model validity and sensitivity analysis, and Objective 4 provides assurances as to its practicality. This research, at its heart, is applied and not theoretical in nature.

Objective #5. Lay the groundwork for operationalization of the proposed framework in the smart, connected manufacturing enterprise.

Discussion: This objective continues the contribution of Objective 4 by taking the next step from the theoretical to the practical realm. The scope of this objective is limited to activities that precede those high-level programming functions typically performed by data scientists and software developers. Creation of functional models and migrating those models to platforms commonly employed by commercial manufacturers are within the scope of this objective, as is the creation of Unified Modeling Language (UML) diagrams to communicate relevant information to software development teams.

CHAPTER 4: BACKGROUND CONCEPTS

The proposed framework integrates four diverse concepts: machine learning, statistical measures of association, applied metaheuristics, and fuzzy inference systems (FIS).

4.1 Machine Learning

Machine learning, as a discipline, intersects several other academic disciplines, most notably computer science, engineering, statistics, and mathematics, with applications in fields as diverse as manufacturing, biology, finance, medicine, and chemistry [110], [111].

It is not the intent of this section to provide a comprehensive overview or tutorial on machine learning, but rather simply to note that the proposed data filtration framework is inextricably linked to the execution of some particular machine learning algorithm in order to solve some particular problem of interest, initially in the arena of smart manufacturing but not restricted to that discipline.

Specifically relevant to this research are supervised machine learning algorithms for classification or regression. In supervised learning, a set of data exists with known outcomes provided. This data set, called a training set, is then used by the algorithm to develop generalizations or identify patterns in the independent variables to predict the dependent variable. Those generalizations or patterns are then validated using a test set and the model is judged based on appropriate metrics depending on whether the goal is classification (categorical response variable) or regression (continuous response variable).

4.2 Statistical Measures of Association

Statistical measures of association attempt to quantify the strength of relationship between sets of data. Four cases are relevant to this dissertation, indicated in Table 5.

Table 5: Relevant Cases for Statistical Measures of Association

	Predictor	**Response**
Case 1:	Categorical	Categorical
Case 2:	Categorical	Continuous
Case 3:	Continuous	Categorical
Case 4:	Continuous	Continuous

The first case is encountered when both the predictor and response variables are categorical. In this circumstance, the test that immediately comes to mind is Pearson's Chi-Square Test [112]. This hypothesis test computes a test statistic by computing the squared deviation between observed and expected frequencies and then dividing by the expected frequency. This quantity, summed over all possible groups, follows the Chi-square distribution and may be tested using conventional hypothesis test procedures.

Pearson's Chi-Square Test, however, is based on the assumption that, within every group, the sample frequencies are normally distributed about the expected population value [113]. Because observed frequencies are nonnegative quantities, small values in any group cast doubt on the validity of the normality assumption. For this reason, Fisher's Exact Test [114], [115] may be preferable to Pearson's Chi-Square Test when low frequencies are expected in one or more groups. For additional discussion, [116] surveys the development and usage of exact inferential methods for contingency tables.

In the second and third cases, the Kolmogorov-Smirnov Test [117] is used to test whether the distribution of the continuous variable at each level of the categorical variable is the same. In Case 2, the null hypothesis is that the distribution of the response variable is the same when conditioning on each possible level of the predictor. Case 3 uses similar reasoning except that it transposes the target and feature for the purposes of the test. The continuous predictor serves as the target, with its conditional distribution determined at each level of the categorical response.

Finally, in the case of a continuous predictor and a continuous response, Kendall's rank test [118] is applied. For a more in-depth treatment of the preceding discussion, as well as application to features extracted from time series data, see [102].

4.3 Applied Metaheuristics

The question of how to select the best subset of features to apply to a model is an application of the optimal subset problem, which is NP-hard [119] because the number of possible subsets grows proportionally to the size of the domain. For this reason, it is necessary to explore heuristic-based techniques that attempt to achieve good, if not optimal, solutions.

At the most basic level, a local search can be performed. This can be summarized by the following "hill-climbing" process:

1. Generate a feasible solution and associated cost function
2. Generate neighborhood solution and associated cost function
3. If cost function improves, store current solution
4. Repeat Step 2 and Step 3 until stopping criteria is achieved

Most simply, the initial feasible solution and neighborhood solutions may be computed randomly. Alternatively, neighborhood solutions may be computed by slightly perturbing one or more elements of the feasible solution. For example, given an initial feasible solution that includes some number of features out of a large number of possible features, the neighborhood solution might keep all but one feature, drop one feature, and add one feature not previously included.

Clearly, the process described above is crude and, while leading to iteratively-improving feasible solutions, is susceptible to stopping at mediocre solutions found at local optima. Three

options to achieve superior results are Tabu Search, Simulated Annealing, and Genetic Algorithm.

4.3.1 Tabu Search

Tabu search extends hill-climbing methods by intentionally allowing for the selection of worse solutions when a local optima is obtained, with the incorporation of short-term memory to ensure that previously-visited solutions are not repeated [120]. For detailed background and theory, see [121]–[124]. The following procedure, outlined in [122], describes a simple Tabu search. Key is the establishment of a set of "off limits" moves that prevent the algorithm from selecting a recently-selected solution.

Notation:

- Let f be the cost function and f* be the best-known objective function value
- Let x be the current solution
- Let x* be the best-known solution,
- Let S(x) be the set of all neighborhood solutions about x
- Let T be the set of prohibited moves, known as the tabu list.

Procedure:

1. As in the hill-climbing process, begin with an initial feasible solution, x_0, and associated cost function, $f(x_0)$. Initially define x* = x = x_0 and f* = $f(x_0)$.
2. Generate neighborhood solutions, S(x) and select the best cost function from the set S(x) – T.
3. If the cost function improves, define x* = x; if not, no change to x*
4. Update T

5. Repeat Step 2 through Step 4 until stopping criteria is achieved.

Note that there is nothing in the algorithm about local optima. If the best cost function attainable from the neighborhood solutions in Step 2 is worse than the best known cost function at x = x*, then the algorithm simply generates a new set of neighborhood solutions and continues until the stopping criteria occurs. Stopping criteria might be some fixed number of iterations, some consecutive number of iterations without an improvement, or some predefined threshold value for the objective function to attain.

Additionally, it should be noted that Tabu search provides significant flexibility to the modeler with respect to how T and S(x) are defined. The neighborhood structure and tabu list structure can heavily influence the algorithm's performance. In particular, a neighborhood structure should attempt to move the solution in an intuitively correct direction [120].

4.3.2 Simulated Annealing

A second option for achieving superior results to a traditional hill-climbing heuristic is simulated annealing, a term coined for its parallelism with the annealing process in metallurgy. In metallurgy, the process of annealing is a heat treatment procedure by which the alloy is first heated, then held at a certain temperature called the annealing temperature, and then cooled in a controlled manner [125]. The purpose of annealing is to relieve internal stresses, soften, and transform the grain structure of the metal into a more stable state [125].

In simulated annealing, the heuristic allows an inferior solution to be retained according to some probabilistic determination. This helps reduce the risk of converging at a local optimum. As the number of iterations increases, the probability of accepting an inferior solution decreases. The high-level procedure for simulated annealing is as follows [126].

Notation:

- Let f be the cost function and f* be the best-known objective function value
- Let x be the current solution
- Let x* be the best-known solution,
- Let S(x) be the set of all neighborhood solutions about x
- Let k be the index of the current iteration
- Let T(k) be a temperature parameter
- Let *p* be a random number on the interval (0, 1)

Note that T(k) is a function of the number of iterations that have taken place. In keeping with the analogy towards metallurgical annealing, T begins at a high temperature and gradually decreases over time. A higher T corresponds to a relatively higher probability of accepting an inferior solution.

Procedure:

1. Similar to the Tabu Search algorithm, begin with an initial feasible solution, x_0, and associated cost function, $f(x_0)$. Initially define x* = x = x_0 and f* = $f(x_0)$. Initialize k = 0.
2. Generate neighborhood solutions, x_{k+1}, from $S(x_k)$ and the associated $f(x_{k+1})$. As previously, the manner of generating neighborhood solutions will likely have a significant effect on the quality of results and the speed of convergence.
3. If $f(x_{k+1})$ improves from the previous iteration, then x* = x_{k+1} and f* = $f(x_{k+1})$.
4. If $f(x_{k+1})$ does not improve from the previous iteration:
    a. Compute p = random(0,1)

b. If $p < exp^{\left(\frac{f(x_k)-f(x_{k+1})}{T(k)}\right)}$, then x* = x_{k+1} and f* = $f(x_{k+1})$.

5. Increment k = k + 1 and update T accordingly.
6. Repeat Step 2 through Step 5 until stopping criteria is achieved.

Note that the above procedure assumes a minimization problem, so that an improvement to f from x_k to x_{k+1} implies that $f(x_{k+1}) < f(x_k)$. Thus, in Step 4, $f(x_{k+1}) > f(x_k)$ because $f(x_{k+1})$ does not improve on the previous iteration. This makes the quantity $f(x_k) - f(x_{k+1})$ negative, which is necessary for the parallelism with metallurgical annealing to hold – higher values of T(k) result in higher probabilities that an inferior solution will be selected; lower values of T(k) result in lower probabilities.

Simulated annealing has been the subject of extensive research, both theoretical and applied. For additional reading on the theory and application of simulated annealing, see [127], [128]. For textbooks on the subject, see [129], [130].

4.3.3 Genetic Algorithms

Genetic algorithms, so named because they employ terminology and techniques that are based on principles of natural selection and genetics, are search methods that evaluate a group of candidate solutions and then evolve subsequent solutions based on the observed attributes in the individual member solutions. The desire is to select and pass on attributes from better solutions and discard the poor solutions and the latent attributes that they carry [131].

Terms and Definitions:

- Candidate solutions are referred to as *chromosomes*, and are analogous to the feasible solutions discussed in previous heuristics

- Elements or sub-components of the chromosomes are referred to as *genes*, and the values assigned to genes are called *alleles*.
- *Selection* refers to the process by which candidate solutions with better fitness values are assigned preference, imposing a survival-of-the-fittest mechanism to the algorithm.
- *Recombination* refers to the generation of new candidate solutions by merging select elements of two or more solutions identified as having traits conducive to fitness. Solutions designated by the selection process as good are then designated as *parent solutions* to be combined to produce *offspring* or *child solutions*.
- *Mutation* refers to the process by which the algorithm will randomly modify a child solution by slightly altering one or more of its elements.

Procedure:

1. Begin with an initial population of feasible candidate solutions. The starting population is typically obtained randomly, although it is possible to use a more guided method.
2. Evaluate each candidate solution using some predefined cost function or measure.
3. Select a subset of the population with superior fitness to serve as parent solutions to create the next population of candidate solutions. A number of possible selection strategies exist [132], including but not limited to:
    a. Select the best *n* candidate solutions
    b. Randomly select *n* candidate solutions, where the probability that a specific candidate solution is selected is proportional to its fitness.
    c. Randomly select *n* groups of *k* candidate solutions. In each group, the *k* solutions are compared and the most fit solution is selected.

4. Perform recombination on the selected candidate solutions to generate the next population of candidate solutions. As with selection, there are a number of recombination strategies [133], two of which described below and illustrated in Figure 3.
    a. Select a crossover point. The offspring takes on gene values (alleles) from Parent A (B) for all genes on one side of the crossover point and alleles from Parent B (A) for all genes on the other side of the cross-over point. Two parents produce a total of two offspring.
    b. Uniform crossover [134]. For each gene, randomly select whether to pass the allele from Parent A or Parent B to the first offspring. The allele from the parent not selected is passed to the second offspring.
5. Add variation to the new population through mutation.
6. Replace the current population of candidate solutions with the newly-generated population.
7. Repeat Step 2 through Step 6 until stopping criteria is achieved.

It could be asked why Step 3(b) and Step 3(c) are options, as they might select inferior candidate solutions to be parents. The reason is simply for the purposes of maintaining genetic variety and maximizing diversity in the gene pool. Additionally, different selection techniques may offer benefits in terms of performance, time to convergence, or computational complexity [132].

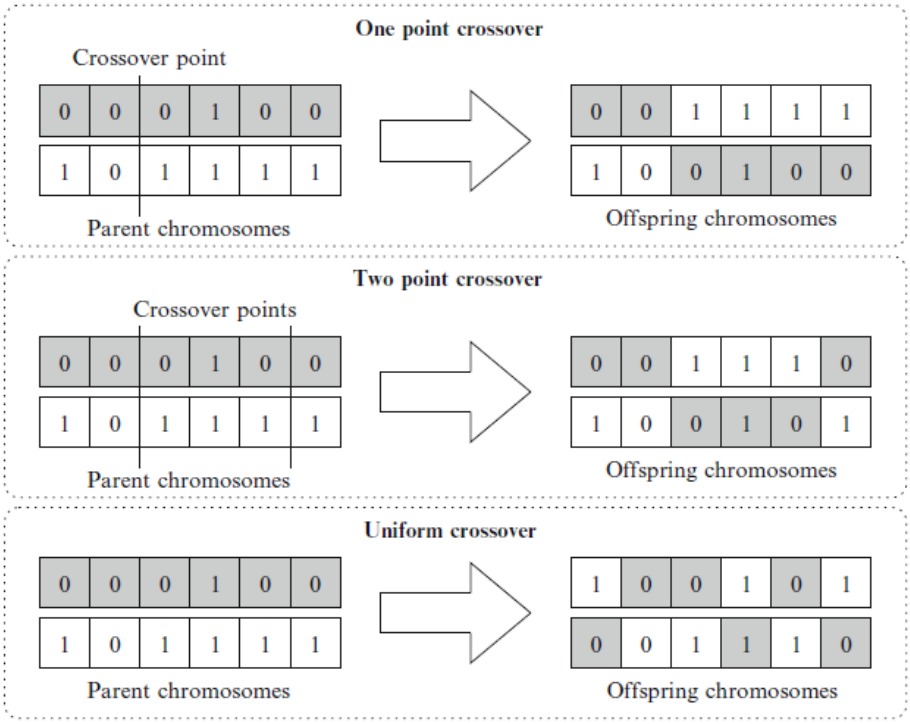


Figure 3: Illustration of recombination methods, taken from [131]

4.4 Fuzzy Inference Systems

A fuzzy inference system (FIS) is a nonlinear mapping from a given input to a given output established using fuzzy logic and fuzzy set theory [135]. A fuzzy set, in contrast to a crisp set, is a set such that membership is defined along some spectrum or degree [136]. An example of a crisp set might be pregnancy status. A person is either pregnant or not pregnant, and there is no continuum between the two. An example of a fuzzy set might be height status. A person can be considered tall, short, average height, or some other label. Furthermore, a specific individual's height, such as 5'- 6", might be considered tall in some contexts, short in some contexts, or average in some contexts. A kindergartner who is 5'- 6" tall would be considered tall; a professional basketball player that height would be considered short.

A Mamdani FIS [137] typically contains four elements: fuzzification, rules, inferencing, and defuzzification [138].

Fuzzification is the process by which a membership function is applied to a crisp input in order to determine the degree of membership in a fuzzy set. For notation purposes, $\mu_A(x)$ denotes the degree of membership in fuzzy set A and is represented by a value between 0 and 1.

Consider a simple example. Suppose that one wishes to use a FIS to label the comfort level of a room. Two important input variables are identified, temperature and humidity, where each can be set at two fuzzy linguistic descriptors: LOW and HIGH. Suppose that the output is the room's comfort level, which can also be categorized as LOW and HIGH.

The first step would be to define appropriate membership functions for the predictor variables. There are a number of different options for membership functions [139], including uniform, triangular, trapezoidal, or Gaussian. See Figure 4 for hypothetical membership functions for this example.

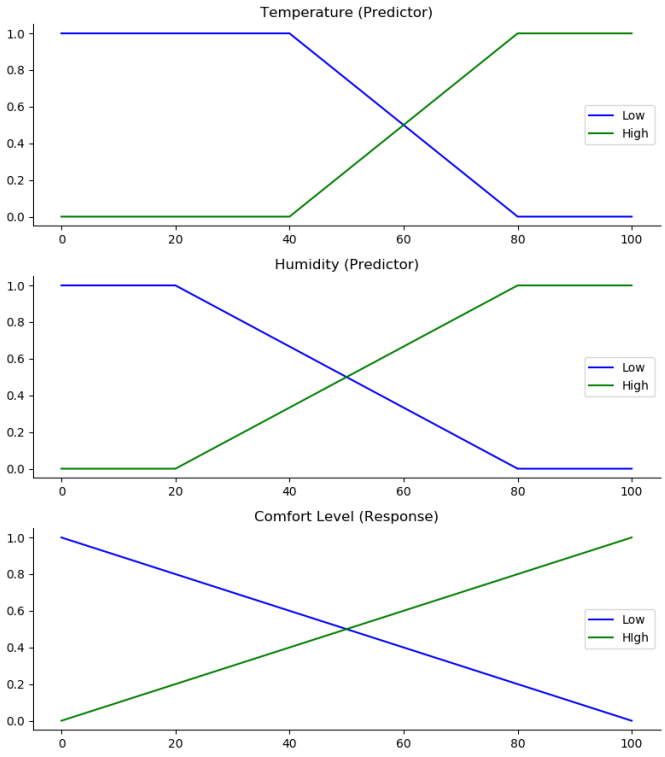


Figure 4: Membership functions

In Figure 4, the membership function for temperature is trapezoidal, where a temperature of 40 degrees or less represents full membership in LOW and no membership in HIGH. A temperature of 80 degrees or higher represents full membership in HIGH and no membership in LOW. Temperatures between 40 and 80 vary the degree of membership linearly between LOW and HIGH. A similar interpretation applies to humidity. The response variable, comfort level, is evaluated on a scale from 0 to 100, with a triangular membership function.

Suppose that it is of interest to assign a comfort level to 73 degrees and 65% humidity. Using the above membership functions, fuzzification maps a crisp temperature input of 73 degrees to 0.825 membership in HIGH and 0.175 membership in LOW. A humidity of 65% maps to 0.75 in HIGH and 0.25 in LOW.

Rules are statements in if-then form that relate the input variables to the output variable. For example, a rule might be:

"If the temperature is HIGH and the humidity is HIGH, then the comfort level of the room is LOW"

A second rule might be:

"If the temperature is LOW and the humidity is LOW, then the comfort level of the room is HIGH"

Inferencing is the application of rules to determine degree of membership in the output. Fuzzy inference systems use logical operators such as AND, OR, or NOT, with different options for how to interpret them. In this example, the operator AND will be defined as the minimum of the two degrees of membership, and OR will be defined as the maximum of the two degrees of membership.

Considering the first rule in our example, the temperature has a degree of membership of 0.825 in HIGH and humidity has a degree of membership of 0.75 in HIGH. This gives the comfort a degree of membership of 0.75 in LOW. Applying the second rule takes the minimum of 0.175 (the temperature's degree of membership in LOW) and 0.25 (the humidity's degree of membership in LOW) and assigns it to the degree of membership in HIGH comfort. Figure 5 contains a plot of the membership function of the output variable that accounts for the two rules.

Defuzzification is a process that takes the fuzzy output and translates it back to a single crisp value. Depending on the context or specific problem of interest, defuzzification may or may not take place. A number of different defuzzification techniques exist [140]. In this example, the centroid technique will be used. The centroid technique computes the center of gravity of the output distribution function and outputs the x-coordinate.

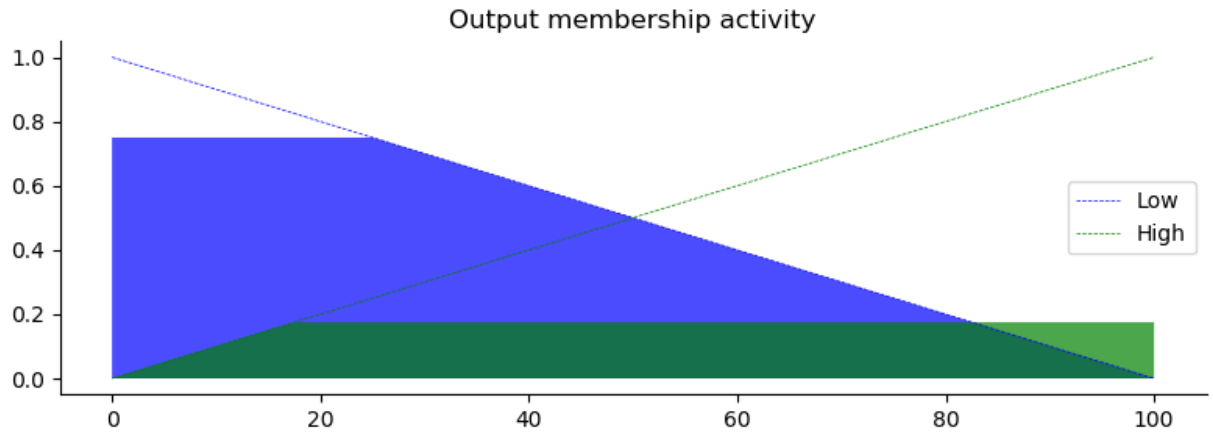


Figure 5: Membership function for output based on model rules

Note that the distributions for each level overlap. Aggregation of rules is typically performed as an OR operation, as illustrated in Figure 6. Figure 6 also inserts the crisp output based on defuzzification as a vertical line.

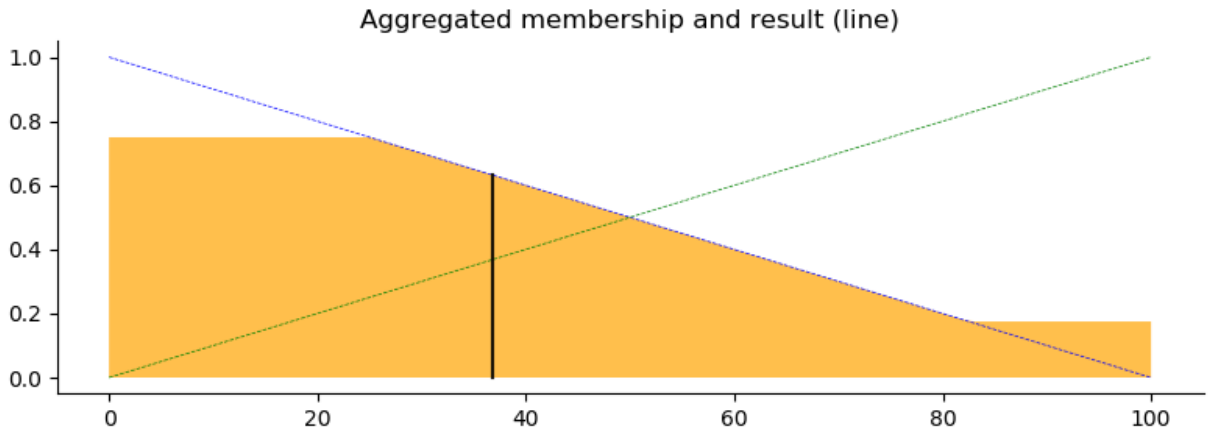


Figure 6: Aggregated output membership with defuzzified result

In our example, the crisp output following defuzzification via the centroid technique is approximately 36.87 out of 100, designated by the vertical line in Figure 6. This maps to a degree of membership of approximately 0.63 in LOW comfort.

It should be pointed out that the number 36.87 has little meaning in isolation. The contrived example sets an arbitrary scale from 0 to 100 for comfort level. The value would be in comparing the metric derived by 73 degrees and 65 humidity with the metric derived by a different temperature and humidity combination.

CHAPTER 5: PROPOSED FILTRATION APPROACH

Chapter 5 and Chapter 6 content is contained within, “A Hierarchical, Fuzzy Inference Approach to Data Filtration and Feature Prioritization in the Connected Manufacturing Enterprise”, submitted for publication in *Journal of Big Data* on 1 October 2018, accepted for publication on 13 November 2018, and published on 19 November 2018 [141].

The starting point for the proposed framework, shown in Figure 7, is a problem of interest and the existence of a population of available data for use both as feature(s) and response(s).

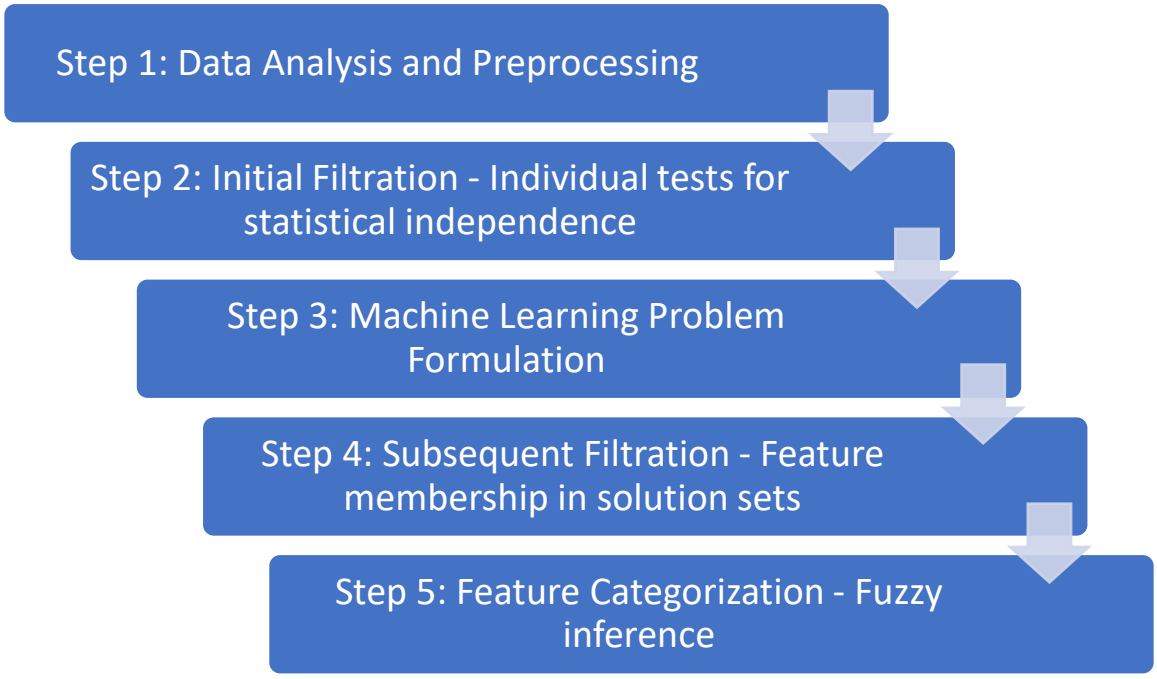


Figure 7: Proposed framework for data reduction and feature labeling

Step 1. Data analysis and preprocessing

The first step is initial data analysis and preprocessing. Of primary interest in this step is the initial identification of a base set of features to apply to the problem of interest. This necessarily involves a high degree of human, subject matter expert involvement. There may

arrive a point at which automated processes anticipate future questions and define their own future problems of interest, but, at this point, the initial formulation of the base problem and base set of features is difficult to separate from the human element.

Additional activities during this step may include putting data in a desired format, screening out features based on subject matter expertise, consolidating data from multiple sources, or extracting features of interest from predictor variables. For example, identification of whether any predictor information is time series and extracting time series features would be appropriate for this stage. The algorithm TSFRESH [102], [105] is an excellent tool for extracting features from time series data.

An additional action that may be necessary to perform in this stage is the identification and encoding of data containing categorical features. Different machine learning tools have degrees to which they are compatible with categorical features; some require that each level of the categorical feature be converted to a numeric quantity.

Options exists for how to encode categorical variables, and different techniques have different advantages and disadvantages. A few examples include one-hot encoding, label or ordinal encoding, target mean encoding, and leave-one-out encoding. One-hot encoding replaces N levels with N-1 binary features equal to 1 if the categorical features is at that level and 0 if not. Label or ordinal encoding replaces each level with an integer. Target mean encoding replaces each level with its proportion of occurrence in the target categories. For example, if the categorical variable is gender (M, F) and the response is binary (0, 1), then M and F would be replaced by the proportion of time each gender equals one. Leave-one-out encoding is similar to target mean encoding except that it leaves out the specific record when calculating the proportions.

The output of Step 1: Data Analysis and Preprocessing is a data set that satisfies the preconditions for whichever algorithm or tool is to be applied to the problem of interest. At this point, most frameworks for applied machine learning proceed to identifying the appropriate technique, developing the model, tuning the hyperparameters if necessary, obtaining the model's results, and then scoring the model using whichever metrics happen to be of interest.

In this case, however, the "problem of interest" is not the only "problem" of "interest". We are also interested in the data itself and how useful each piece of data is with respect to obtaining the ultimate solution. For this reason, the proposed framework inserts filters that successively eliminate factors from consideration for use in model training.

Step 2. Filter #1: Individual Strength of Association

The first filter is to apply strength of association hypothesis tests between each feature under consideration and the response variable. This filter individually tests each feature against the response, where the null hypothesis is that the two variables are statistically independent, having no relationship to each other. The hypothesis tests will each produce a p-value, which is then compared to the Benjamini-Hochberg threshold [103]; features whose p-values are in tolerance are retained and features whose p-values are not in tolerance are discarded.

The Benjamini-Hochberg test controls the false discovery rate (FDR) at a predefined threshold. The FDR is simply the rate by which random chance allows for the hypothesis test to indicate a significant relationship when there is none. For example, suppose that patients are tested for a disease, where the test has a significance level of .05. If 100 patients known to be disease-free are given the test, then by virtue of randomness one would expect approximately 5

patients to be erroneously flagged as ill. It is not possible to completely eliminate Type I error, but the Benjamini-Hochberg procedure controls the rate by which it occurs.

$$p_{BH} = \frac{r_i}{n} * Q \quad (1)$$

Equation (1) contains the formula for the Benjamini-Hochberg threshold, where r_i is the rank of the p-value associated with feature *i*, *n* is the number of tests performed, and *Q* is the required FDR, typically .05.

Alternative criteria for determining a threshold p-value do exist, such as the Bonferroni adjustment. The Bonferroni adjustment seeks to control the familywise error rate by proportionately reducing the p-value threshold based on the number of tests performed [142]. If a familywise error rate of .05 is desired, and five tests are to be performed, then each test is independently performed with a p-value of .01. This is different from Benjamini-Hochberg in that it controls the probability that at least one test will produce a false discovery, in contrast to controlling the rate of false discovery. As a result, the Bonferroni adjustment is more restrictive than Benjamini-Hochberg, allowing fewer features to pass forward. Benjamini-Hochberg is selected for use in this framework because the presence of a second filter makes it less damaging for one or more poor features to be mistakenly lumped in with good features.

Step 3. Machine Learning Problem Formulation

Up to this point, the specific problem of interest has received only superficial treatment. In Step 1, subject matter expertise was employed to shape the initial set of features, and the data was prepared for use. Step 2 is agnostic to the larger problem of interest or machine learning technique to employ, rather focusing exclusively on any statistical dependence that may exist between an individual feature and the response variable.

At this point, it is necessary to formulate with some degree of specificity the machine learning model to apply to the problem of interest. This includes, at a minimum, designating the machine learning technique and specifying the metrics by which it will be scored. Some techniques, such as random forest, contain hyperparameters that may require tuning.

This step is technically outside the scope of this dissertation's primary interest but is included in the framework as a necessary prerequisite to the next filter. There is a substantial body of literature on formulating machine learning problems, and additional reading on the subject is left for the reader to explore as needed.

Step 4. Filter #2: Membership in Solution Groups

After initially testing each feature in isolation against the response variable to determine if there is a statistical dependence between the two, a second filter is applied to the remaining features that examines performance as a member of a collective group. A subset of features is selected, either randomly or according to some intentional process, and used as input for the machine learning algorithm selected in the previous step. The selected features may yield a high-quality solution or a low-quality solution, with quality measured according to those predetermined metrics of interest identified in Step 3. This process is then repeated some number of times until there exists a sufficiently substantial body of high-quality solutions and low-quality solutions to make inferences regarding the specific features that tend to appear in each group.

There are at least two potentially desirable outcomes from this step. One might simply be to obtain a high quality solution to the problem of interest. If the set of available features is large, the undertaking becomes non-trivial, as the problem to obtain an optimal subset is NP-Hard. In

this case, it may be advisable to employ a metaheuristic such as Tabu search, Simulated Annealing, or Genetic Algorithm to allow the solutions to evolve and improve. This approach might best apply when the problem of interest is relatively self-contained and there is not the expectation to extensively generalize its results. If the set of available features is not especially large, it may be economical simply to enumerate every possible combination of features and then select the combination that produces the best result.

A second outcome, more directly aligning with the purpose of this framework, is to make a quantified statement of knowledge regarding how useful each remaining feature is with respect to its contribution to the model. For this outcome, achieving a near-optimal subset presents only a partial picture. Suppose, for example, that application of Genetic Algorithm in this step produces an optimal or near-optimal subset of features. A feature's presence in or absence from that single superior subset does not tell us the same thing about its general usefulness as would the observation that that feature habitually appears in subsets that tend to be good versus subsets that tend to be bad. There will always be the nagging uncertainty as to whether a feature is in the superior subset because it is broadly useful or because its interaction with the other members of the subset produces a uniquely outstanding result.

Step 5. Feature Categorization through Fuzzy Inference

At this point, the initial set of features has been reduced once by eliminating all features with strong evidence of statistical independence from the response variable. The reduced set of features has subsequently been the population from which sampling with replacement has created a large number of subsets, which have been used as inputs into a machine learning model as appropriate for the problem of interest. As each model is scored by some quality metric, there

is a one to one mapping from each feature subset to a scoring metric. For the sake of illustration, suppose that classification accuracy is the metric of interest.

Step 5 groups the subsets based on their scoring, quantifies each feature's membership in the various groups, and employs that crisp value as an input to a FIS for the purpose of generating a label. This label, applied to each feature, will qualitatively describe the usefulness of the feature. Two sets of output labels are proposed for consideration, the first being general and the second being more detailed. The first set of labels is as follows:

- Strong Utility
- Moderate Utility
- Weak Utility

The strength of this set of labels is in its simplicity and applicability to relatively straightforward decisions. For example, the decision may be whether to migrate data from a particular feature to cloud storage for long term preservation and analysis. Features classified as Strong or Moderate would perhaps make the cut, while Weak features would not. Figure 8 illustrates the output membership function using these labels.

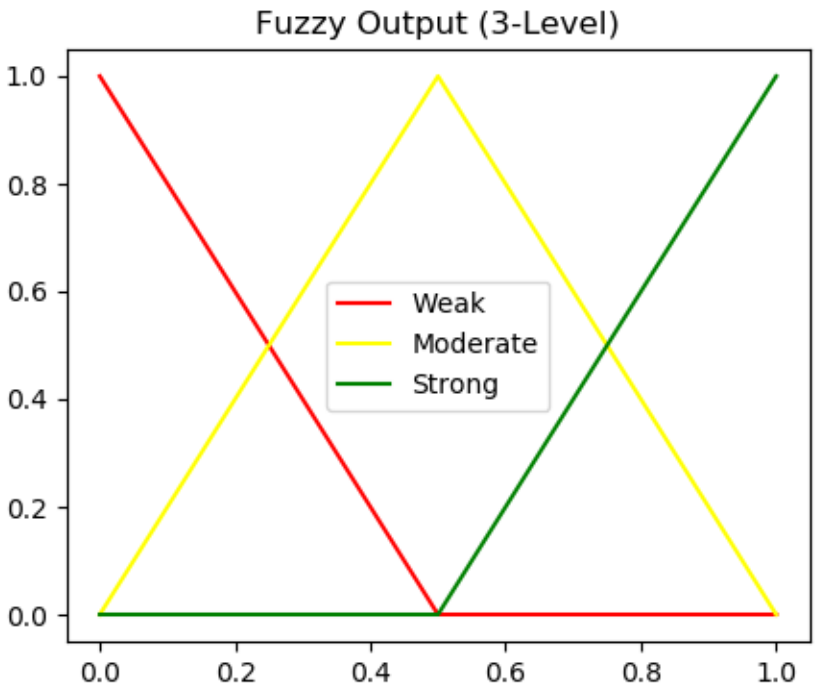


Figure 8: Membership function, 3-level output

The second set of labels provides additional descriptive information:

- Level 1 (L1). This label is for features which produce high quality solutions in single-variable models. Other features receiving this highest level would be those appearing consistently in high-quality solutions but exhibit low or nonexistent membership in low-quality solutions.
- Level 2 (L2). This label is for features that exhibit strong membership in high-quality solutions but also exhibit some non-trivial membership in low-quality solutions. This is an indicator that the value contributed by these features is in their interaction with other features in producing a high-quality result.
- Level 3 (L3). This label is for features tending not to appear in the best solutions but also tending not to appear in the worst solutions. Rather, these features tend to appear in the "squishy middle", the subsets of features producing generally unremarkable results.

- Level 4 (L4). This label is for features that exhibit strong membership in low-quality solutions but exhibit non-trivial membership in high-quality solutions.
- Level 5 (L5): This label is for features that perform poorly regardless of circumstance, exhibiting strong membership in low-quality solutions and low or nonexistent membership in high-quality solutions. These features only contribute to non-poor solutions when they happen to be combined with especially high-performing features.

The strength of this set of labels is in the information it provides. It may be useful to distinguish between features that perform well in isolation and features that only perform well when interacting with other features. Figure 9 contains a plot of the membership function for the more descriptive, 5-level output.

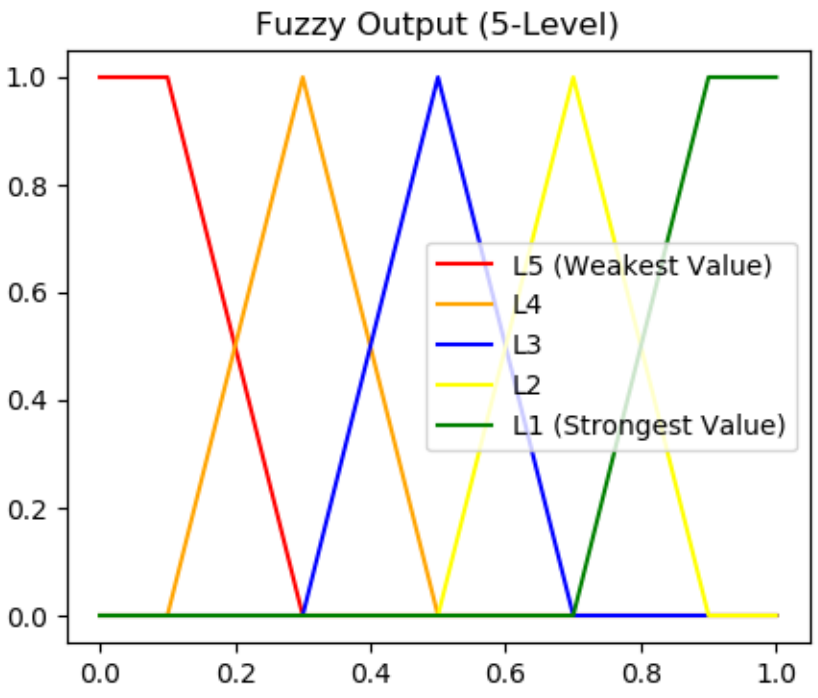


Figure 9: Membership function, 5-level output

The FIS in this step follows the following procedure.

(1) Identification of inputs

(2) Definition of input membership functions

(3) Definition of rules

(4) Fuzzification

(5) Activation of rules to create output distribution

(6) Defuzzification

FIS Step (1): Identification of Inputs

It is proposed to employ two fuzzy input variables for each metric of interest. One of the fuzzy input variables will be the feature's membership in feature subsets producing high quality results and the other fuzzy input variables will be the feature's membership in feature subsets producing low quality results. For example, suppose that the metrics of interest for a classification problem are precision and recall. This FIS would require four fuzzy input variables, as indicated by Table 6.

Table 6: Example fuzzy input variables

Fuzzy Input	Description
X_1	Feature's membership in High Precision group
X_2	Feature's membership in Low Precision group
X_3	Feature's membership in High Recall group
X_4	Feature's membership in Low Recall group

There is a relationship between the definition of a feature's membership in a group and the formulation of its membership function in Step (2). In some cases, the fuzzy input variable denotes something familiar, such as temperature or humidity, as in the example in Section 4.4. In those circumstances, the feature's crisp input can reasonably follow the appropriate familiar convention.

In the case of creating a quantified input for a feature's membership in a group, there is flexibility because this quantity does not create automatic associations in the mind of the reader. A reasonable convention is to create a quantity between 0 and 1 for group membership, with 1 being perfect membership and 0 being no membership at all.

$$x_i^{(j)} = \frac{\sum_k I_k^{(j)}}{K_i} \tag{2}$$

The proposed convention for computing a crisp quantity for feature membership in a group is presented in Equation (2), where $x_i^{(j)}$ represents the membership of feature j in fuzzy input X_i, K_i represents the number of subsets in group i, and $I_k^{(j)}$ represents an indicator function equal to 1 if feature j is contained in subset k, where $k = 1 \ldots K_i$ and 0 if it is not.

FIS Step (2): Definition of Input Membership Functions

Fuzzy input variables require membership functions, which take the crisp inputs computed in Equation (2) and convert them into degrees of membership in linguistic labels. For this framework, two linguistic labels are proposed: Strong and Weak. One or more middle-ground labels could certainly be considered, but they are omitted because the extra labels add unnecessary complexity to the rule generation step, as will be illustrated below.

As discussed in Section 4.4, analysts have a host of different membership functions from which to choose. It is proposed to use trapezoidal membership functions, where membership in Low (High) equals one (zero) for $x_i^{(j)} < 0.25$ and equals zero (one) for $x_i^{(j)} > 0.75$. Figure 10 illustrates for a generic, 2-level fuzzy input.

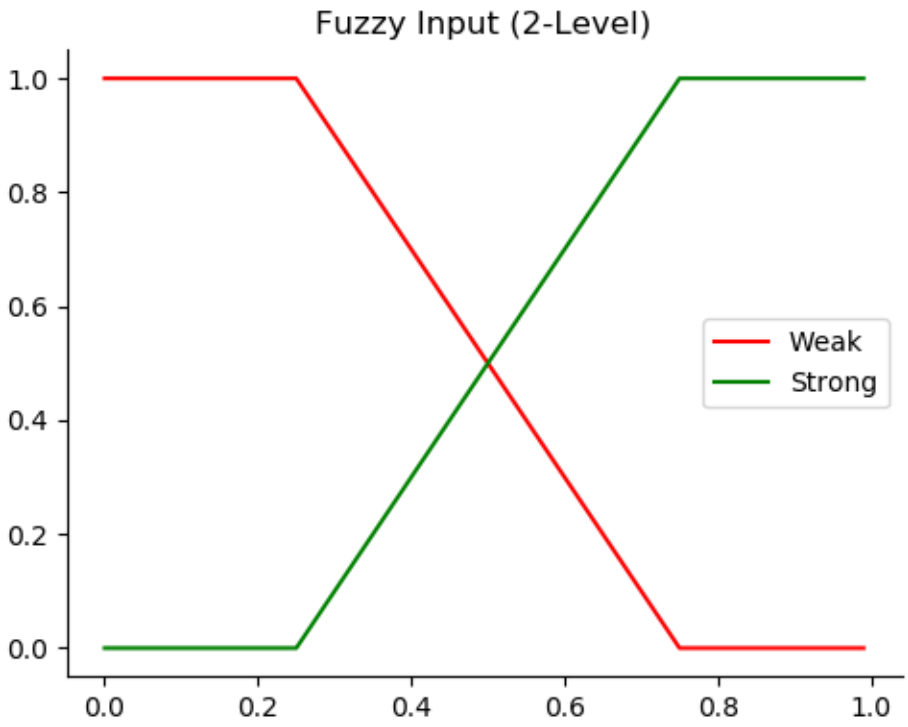


Figure 10: Membership function, 2-level input

The trapezoidal membership function is selected because it is reasonable to declare a feature as definitively weak or strong when its crisp input is outside of certain parameters.

FIS Step (3): Definition of Rules

The rules definition stage is arguably the most important piece of the FIS because the rules drive the construction of the output distribution. As discussed in Section 4.4, rules take the IF – THEN form and use logical operators such as AND, OR, and NOT. Table 7 contains the conventions to be employed for the fuzzy logical operators.

Table 7: Fuzzy logical operator conventions

Logical Operator	Convention
AND	$\min(\mu_A(x), \mu_B(x))$
OR	$\max(\mu_A(x), \mu_B(x))$
NOT	$1 - \mu_A(x)$

In Table 7, $\mu_A(x)$ refers to the degree of membership of fuzzy input variable A given a crisp input *x*.

Suggested rules for general outputs of Strong, Moderate, and Weak are as follows.

1. If (Metric=High) is STRONG and (Metric=Low) is NOT STRONG, then Utility is STRONG
2. If (Metric=High) is WEAK and (Metric=Low) is NOT WEAK, then Utility is WEAK
3. If (Metric=High) is STRONG and (Metric=Low) is STRONG, then Utility is MODERATE
4. If (Metric=High) is WEAK and (Metric=Low) is WEAK, then Utility is MODERATE

Suggested rules for the five-level output labels are as follows.

1. If (Metric=High) is STRONG and (Metric=Low) is WEAK, then Utility is Level 1
2. If (Metric=High) is STRONG and (Metric=Low) is NOT WEAK, then Utility is Level 2
3. If (Metric=High) is WEAK and (Metric=Low) is WEAK, then Utility is Level 3
4. If (Metric=High) is NOT STRONG and (Metric=Low) is NOT WEAK, then Utility is Level 4
5. If (Metric=High) is WEAK and (Metric=Low) is STRONG, then Utility is Level 5

FIS Step (4): Fuzzification

The proposed framework imposes no constraints regarding fuzzification. Fuzzification should be implemented as per established FIS conventions as described in Chapter 4.0.4.

FIS Step (5): Activation of Rules

The proposed framework imposes no constraints regarding activation of rules to generate the output distribution. Activation of rules should take place as per established FIS conventions as described in Chapter 4.0.4.

FIS Step (6): Defuzzification

The proposed framework imposes no constraints regarding defuzzification. Selection of defuzzification technique should take place as per established FIS conventions as described in Chapter 4.0.4.

In conclusion, the approach discussed in the preceding chapter has been coined and will be referred to as the Fuzzy Approach to Feature Reduction and Prioritization (FAFRAP) for conciseness. The following chapters will explore the FAFRAP approach as applied to a diverse array of machine learning datasets.

CHAPTER 6: FAFRAP APPROACH IMPLEMENTATION

This chapter contains three applications of the FAFRAP method on three datasets. The content in this chapter appears in, "A Hierarchical, Fuzzy Inference Approach to Data Filtration and Feature Prioritization in the Connected Manufacturing Enterprise" by LaCasse, et al., published in the *Journal of Big Data*, December 2018, Volume 5, Issue 1.

6.1 Example #1: Robot Execution Failures

Description of Data

The FAFRAP approach will be illustrated using a multivariate time series dataset that contains force and torque measurements on a robot after failure detection, obtained via the University of California Machine Learning Archive (https://archive.ics.uci.edu/ml/machine-learning-databases/robotfailure-mld/robotfailure.data.html) [104]. The dataset consists of fifteen instances for each of six different time series variables. There is a variable for force and a variable for torque in each of the x-, y-, and z- directions. The response variable is whether the failure detection turned out to be an actual failure or turned out to be nothing wrong. Out of 88 failures detected, there are 21 false alarms and 67 actual faults.

Step 1: Data Analysis and Preprocessing

For this dataset, it is necessary to address two elements. The first element is that the data is divided into two sources, a source for the time series data and a source for the outcome. The second element is that time series data requires an initial step of feature extraction. The Python TSFRESH library is used to extract features from time series data. The algorithm allows for a total of 794 potential features [105] to be extracted from a single time series. For detailed discussion and definitions of potential extracted features, the reader is referred to TSFRESH documentation (https://media.readthedocs.org/pdf/tsfresh/latest/tsfresh.pdf). For this example,

TSFRESH extracts a total of 4764 features from the six different time series. The 88 records for each of the new 4764 features are then linked to the corresponding 88 instances of the response variable and then split into training and test sets.

Because the extracted features are all numerical values, it is not necessary to encode any categorical variables.

Step 2: First Filter

Initial filtration through testing each feature individually for statistical dependence with the response variable and then applying the Benjamini-Hochberg threshold reduces the available features from 4764 to 441. This is a reduction of more than 90% in terms of the raw number of features and a reduction of approximately 80% in terms of the size of the data. As for feature extraction, the Python TSFRESH library has user-friendly functionality to quickly cycle through the hypothesis tests in this step [105].

Step 3: Machine Learning Problem Formulation

At this point, the dataset containing the 88 records for the reduced set of features could be used to train a classification model. For the proposed framework, formulation of the machine learning problem is a necessary precondition to applying the second filter.

Our example of attempting to predict robot execution failures is a classification problem with a binary response variable. Somewhat arbitrarily, support vector machines (SVM) is selected as the technique, with the software engine being Python's "sklearn" [143] library and the "svm" package. Because the proposed framework is concerned with features' relative value compared to each other, there was no attempt made in this example to tune or optimize hyperparameters; rather, default settings were used. In sklearn, the kernel function defaults to

radial bias function (RBF) and a kernel coefficient of 1 / (# features) [144]. The 88 records were divided 80% into the training set and 20% into the test set. Due to rounding, this translates to 70 records in the training set and 18 records in the test set.

Step 4: Second Filter

The number of features remaining after the first filter in Step 2 is 441, a number that is prohibitively large if the desire is to obtain the optimal subset of features. The number of subsets that could be generated by a set of this size is approximately 5×10^{133}. If it were possible to check one million subsets every second, it would still take approximately 1.8×10^{119} years to iterate through every possible subset.

A total of N = 1,000 subsets of size K = 5 each were randomly generated and used to train the machine learning algorithm described in Section 4.2.3. Applying the trained model to a test set and scoring the results in each of the 1,000 subsets using Cohen's Kappa, subsets were either designated as "High Quality", "Low Quality" or discarded. High Quality solutions were defined as those with a Cohen's Kappa statistic greater than 0.85. Low Quality solutions were defined as those with a Cohen's Kappa statistic less than 0.15. Out of 1,000 subsets, 58 were placed in the High Quality group and 446 were placed in the Low Quality group. The remaining 496 subsets scored between 0.15 and 0.85 and were not placed in either group. Crisp inputs for each feature in each group were obtained using Equation (2).

Due to the high number of subsets in the two groups, one adjustment was necessary to Equation (2). The large K_i for each group resulted in very small values for the $x_i^{(j)}$. In order to reward the better-performing features, the value obtained in Equation (2) was normalized by dividing each one by the highest $x_i^{(j)}$ attained, as indicated by Equation (3).

$$x'^{(j)}_i = \frac{x_i^{(j)}}{\max(x_i^{(j)})} \tag{3}$$

This adjustment better allows the crisp inputs to map to intuitive fuzzy labels in the FIS stage of the framework.

Fuzzy Inference System

The FIS for this problem contains two input variables. High Quality group membership and Low Quality group membership. For each variable, a 2-level membership function was employed as indicated by Figure 8. The crisp input for each input variable is calculated using Equation (4).

The output variable is denoted as Value and uses five levels as described in Chapter 5. Figure 11 contains an extract from a Microsoft Excel spreadsheet of the FIS crisp inputs, degrees of membership determined by fuzzification, rule results, crisp output calculated in the defuzzification step using the centroid method, and finally a degree of membership for the crisp output based on the aggregated output distribution generated by the FIS.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Feature (Index):	Crisp Input (High Group):	High_WEAK:	High_STRONG:	Crisp Input (Low Group):	Low_WEAK:	Low_STRONG:	Rule 1 Result:	Rule 2 Result:	Rule 3 Result:	Rule 4 Result:	Rule 5 Result:	Defuzzified Output (Centroid):	Output Degree of Membership:
2	0	0.4	0.7	0.3	0.266666667	0.966666667	0.033333333	0.3	0.033333333	0.7	0.033333333	0.033333333	0.59526102	0.523694901
3	1	0	1	0	0.333333333	0.833333333	0.166666667	0	0	0.833333333	0.166666667	0.166666667	0.427037537	0.635187683
4	2	0	1	0	0.4	0.7	0.3	0	0	0.7	0.3	0.3	0.382135231	0.410676157
5	3	0	1	0	0.933333333	0	1	0	0	0	1	1	0.204761905	0.523809524
6	4	0.4	0.7	0.3	0.266666667	0.966666667	0.033333333	0.3	0.033333333	0.7	0.033333333	0.033333333	0.59526102	0.523694901
7	5	0.4	0.7	0.3	0.066666667	1	0	0.3	0	0.7	0	0	0.612243346	0.43878327
8	6	0.6	0.3	0.7	0.333333333	0.833333333	0.166666667	0.7	0.166666667	0.3	0.166666667	0.166666667	0.63845343	0.3
9	7	0.2	1	0	0.333333333	0.833333333	0.166666667	0	0	0.833333333	0.166666667	0.166666667	0.427037537	0.635187683
433	...	...	...	...	...	...	...	...	...	...	...	...	...	...
436	433	0	1	0	0.2	1	0	0	0	1	0	0	0.5	1
437	434	0	1	0	0.4	0.7	0.3	0	0	0.7	0.3	0.3	0.382135231	0.410676157
438	435	0	1	0	0.333333333	0.833333333	0.166666667	0	0	0.833333333	0.166666667	0.166666667	0.427037537	0.635187683
439	436	0.2	1	0	0.066666667	1	0	0	0	1	0	0	0.5	1
440	437	0.2	1	0	0.133333333	1	0	0	0	1	0	0	0.5	1
441	438	0	1	0	0.133333333	1	0	0	0	1	0	0	0.5	1
442	439	0.2	1	0	0.133333333	1	0	0	0	1	0	0	0.5	1
443	440	0	1	0	0.266666667	0.966666667	0.033333333	0	0	0.966666667	0.033333333	0.033333333	0.483163887	0.915819433

Figure 11: FIS Results for Robot Failure example

Consider the output for Row 2, corresponding to Feature 0. Out of the 58 subsets in the High Group, exactly two contained Feature 0. The highest number of subsets in which any of the

441 features appears is five, giving a normalized crisp input of 0.4. Applying the trapezoidal membership function for the High Group fuzzy input, degrees of membership of 0.7 in WEAK and 0.3 in STRONG are obtained and found in cells C2 and D2 respectively.

Similar reasoning and calculations give the values in cells E2, F2, and G2. For cell E2, Feature 0 is present in 4 out of 446 Low Group subsets; this value is normalized by dividing by the maximum number for any feature, which happens to be 15 in this case.

Cells H2, I2, J2, K2, and L2 give the outputs for each rule applied to Feature 0. To illustrate, consider cell H2 for Rule 1. Rule 1 takes the minimum of the feature's degree of membership in High_STRONG (0.3) and Low_WEAK (0.967) and applies it to the L1 label as defined in Chapter 5. This applies a value of 0.3 to the output distribution for the L1 label. Similar application of rules two through five gives the remaining values.

Column M provides the crisp, defuzzified output, calculated using the centroid technique. The centroid technique computes the center of mass of the output distribution and returns the coordinate for the horizontal axis. For this model, defuzzification was performed using Python's skfuzzy [145] package, a fuzzy toolkit for the scikit-learn [143] library.

Finally, Column N gives the degree of membership of the crisp output in the output distribution. Let DOM_o be the output degree of membership. If this value is less than one, then the interpretation is to apply it to whichever label is indicated by the output distribution. In this problem, the output membership function is defined such that not more than two labels will overlap. It is possible to interpret the quantity $1 - DOM_o$ as be the degree of membership in the other label. However, caution should be applied when doing so because this could result in conflict with one or more rule results. Figure 12 illustrates for Feature 0.

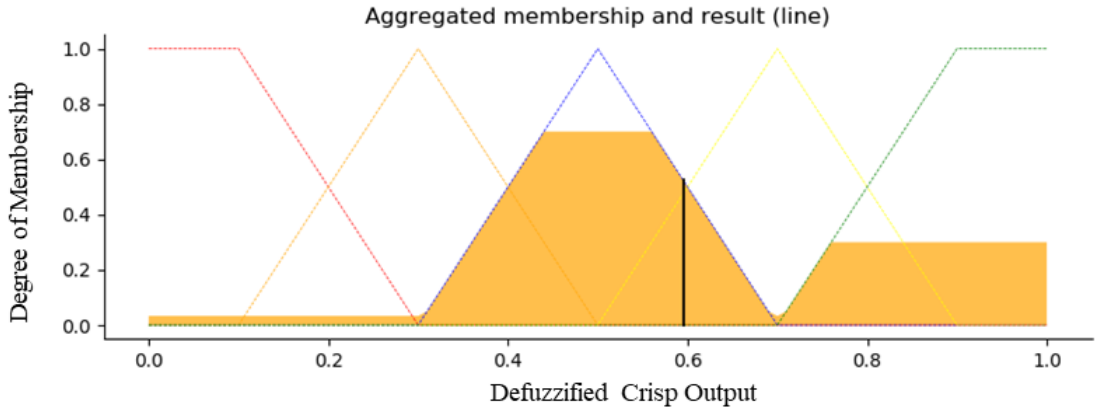


Figure 22: Output distribution and crisp output for Feature 0 in Robot Failure example

The black vertical line represents the crisp output, and its point of intersection with the L3 label boundary corresponds to a value of approximately 0.5237 on the vertical axis, as indicated in Cell N2 of Figure 11.

Note that the highlighted portions of the output distribution correspond to the rule outputs, with overlaps defaulting to the higher value. In this case, it is not advisable to interpret Feature 0 as having a degree of membership of 0.5237 in L3 and 0.4763 in L2 because that would conflict with Rule 2, which applies a value of only 0.033 to the output distribution for L2. Stated differently, because Rule 2 prohibits Feature 0 from having a degree of membership in L2 that exceeds 0.033, we cannot infer a degree of membership of 0.4763 from the defuzzified output.

Results

Out of 441 features, the model categorizes 15 at Level 1, or features of the highest value. The model categorizes 76 features at either Level 5 or Level 4, the two lowest-value labels. Table 8 summarizes the results.

Table 8: Feature Classification Summary for Robot Failure example

Label	Number of Features
Level 1 (highest value)	15
Level 2	2
Level 3	348
Level 4	76
Level 5 (lowest value)	0

Discussion

The first immediate observation from this example is that none of the 441 features were classified as L5 and only two are classified as L2. This may be a sound result or it may be indicative of an anomaly or suboptimal element in the FIS. It is possible that sampling only 1,000 subsets was not a sufficient number to ensure that the features are appropriately distributed across the output labels. This hypothesis may be supported by the large number of features in the "squishy middle", showing neither strong membership in the High Quality group nor weak membership in the Low Quality group. To test this, the model was run again, this time taking 200,000 random subsets instead of the initial 1,000. Results are displayed in Table 9 and are not substantially different from those obtained in Table 8, the most noticeable difference being a number of features shifted from Level 3 to Level 4.

Table 9: FIS Results (200,000 subsets)

Label	Number of Features
Level 1 (highest value)	13
Level 2	2
Level 3	312
Level 4	114
Level 5 (lowest value)	0

A second possible explanation is that a 2-level fuzzy input variable is not sufficiently descriptive to give adequate treatment to one or more rules. For example, Rule 2 attempts to capture the situation by which a feature shows strong presence in high quality solutions and a

not-weak presence in low quality solutions. However, with only two labels for the fuzzy input variables, a not-weak presence in low quality solutions is not distinguishable from a strong presence in low quality solutions.

Figure 13 contains an alternative membership function for the two fuzzy input variables, this time containing three levels: STRONG, MODERATE, and WEAK.

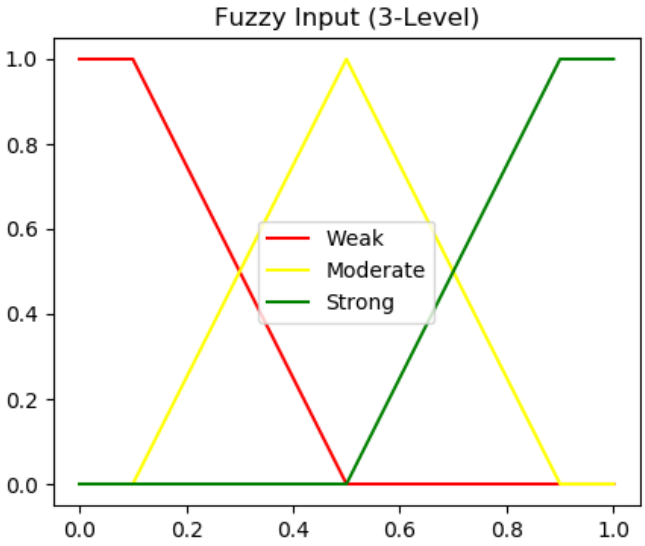


Figure 33: Membership function, 3-level fuzzy input variable

Table 10 contains model results for the modified FIS, generated using 1,000 subsets.

Table 10: Modified FIS Results, using 3-level input membership function

Label	Number of Features
Level 1 (highest value)	2
Level 2	13
Level 3	181
Level 4	245
Level 5 (lowest value)	0

This small change to the FIS results in a marked improvement in the results from a plausibility perspective. It is entirely reasonable to expect that the smallest number of features will reside at the extremes; in this case, two features are designated Level 1 and no features are

designated Level 5. It is also reasonable to expect that most features would not be especially useful. Considering the Pareto principle ([146]–[148]) – if most variation can truly be explained by a relatively small number of variables, then it is natural to expect increasing numbers of features to be classified in the lower value categories.

A second point of discussion is the sensitivity analysis. It has already been demonstrated that increasing the number of randomly generated subsets from 1,000 to 200,000 does not substantially change the numbers of features assigned to each level of value. However, upon closer examination, it is revealed that the two scenarios do not necessary flag the same features in the same way. Put in different words, it is not clear the extent to which model replications consistently assign the same labels to the same features. Table 11 lists the top 15 features for each scenario, rank ordered by their defuzzified crisp outputs.

Table 11: Comparison of best feature categorization, 2-level input membership functions

	Scenario 1: 1,000 Subsets		**Scenario 2: 200,000 Subsets**	
Rank Order	Feature	Crisp Output	Feature	Crisp Output
1	15	0.891666667	22	0.881925144
2	95	0.891666667	21	0.874986572
3	182	0.891666667	4	0.869815795
4	22	0.881298879	417	0.863758421
5	330	0.881298879	42	0.863278371
6	108	0.813182504	156	0.858847362
7	129	0.733510773	12	0.846923559
8	138	0.733510773	95	0.845564336
9	370	0.733510773	6	0.829632339
10	68	0.711872842	47	0.814060738
11	107	0.711872842	230	0.753711779
12	159	0.711872842	125	0.73260824
13	169	0.711872842	82	0.707993795
14	226	0.711872842	201	0.69445375
15	386	0.711872842	28	0.684691585

Table 11 shows that there is almost no overlap between the top fifteen features in each scenario, with only one feature, Feature 22, appearing in both lists. One possible culprit might, again, be the number of subsets in the two samples. Perhaps, for 441 features to consider, 1,000 subsets is too few to generate consistent results.

To test this hypothesis, the model was run n = 10 times with k = 10,000 subsets each run. Figure 14 contains an extract of Microsoft Excel output with the top performing features in each run and the aggregated results.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10		Index	Count
2	21	239	42	47	21	21	4	4	12	22		4	10
3	22	42	4	42	4	4	156	22	6	180		22	10
4	12	4	95	22	22	164	21	47	42	4		21	9
5	398	22	417	21	95	22	219	95	22	12		42	7
6	4	64	21	417	66	95	125	21	21	230		95	7
7	42	316	138	68	156	12	92	434	4	21		47	5
8	82	128	47	95	5	6	22	42	95	184		417	5
9	98	47	45	4	42	77	82	386	325	430		6	4
10	369	280	17	145	114	138	6	281	163	64		45	4
11	128	350	22	239	83	385	45	376	426	84		156	4
12	404	28	230	46	47	84	274	26	417	206		12	4
13	72	193	101	75	180	68	289	64	45	417		64	3
14	95	156	238	45	6	180	79	77	230	325		180	3
15	139	347	74	139	131	339	61	350	101	156		230	3
16	417	200	48	339	15	5	184	347	206	61		5	2

Figure 44: Highest Ranked FIS Output for 10 runs, 10,000 subsets per run

Columns L and M give the feature index and the aggregated counts for how many times the feature appears in the Top 15 performing features, summed over the ten runs. Out of 441 features, 427 appear fewer than three times, and only five features appear seven or more times. Those five features, which consistently score highly in the FIS, are candidates to retain.

A third point of discussion follows directly from the sensitivity analysis. Example #1 started with 4,764 features, reduced to 441 following Filter #1, and has been potentially reduced to five at the culmination of the proposed framework. However, this reduction is only acceptable if the remaining features produce acceptable solutions to the problem of interest.

Two questions are necessary to answer. The first is if solutions produced by the final, reduced set of features will generate comparable results to the solutions produced by the full set of features. The second is if those solutions are acceptable to be used for their intended decision-making purposes.

The full model, using all 441 features, performs extremely poorly in this case. Likely due to overfitting, a model trained on 80% of the dataset and tested on 20% of the dataset makes the same prediction for all records in the test set. Table 12 contains the confusion matrix.

Table 12: Confusion matrix using all 441 features

Actual (Down) \ Predicted (Across)	**0**	**1**
0	11	0
1	7	0

In contrast, Table 13 contains the confusion matrix for a model generated using the top five features from Figure 12: Features 4, 21, 22, 42, and 95. The refined model perfectly classifies the 18-record test set.

Table 13: Confusion matrix using best-rated features from FIS

Actual (Down) \ Predicted (Across)	**0**	**1**
0	11	0
1	0	7

It should be clarified that, for this example, support vector machines was selected as the machine learning algorithm to use not because it is ideal or performs especially well, but rather to illustrate the contrast between model performance with the full set of features versus the reduced set of features. Had an alternative technique been chosen, such as classification trees or random forest, the contrast would have been less pronounced.

A final point of discussion will be the introduction of a novel metric to quantify the relationship between relative performance and relative size of the different models generated by the various stages and filters in the framework.

$$PSR = \frac{Model\ Relative\ Performance}{Model\ Relative\ Size} = \frac{M_i/M_0}{S_i/S_0} \quad (5)$$

This metric, denoted as PSR, is the Performance-Size Ratio for a given model at some stage in the framework. In Equation (5), M_i represents the performance of the model after Step i in the proposed framework, M_0 represents the performance of the base model with no feature reduction, S_i represents the size of the model after Step i, and S_0 represents the size of the base model with no reduction. The PSR will be applied to Example #2 in Section 6.2.

6.2 Example #2: Single Proton Emission Computed Tomography (SPECT) Images

Description of Data

A second example illustrates the FAFRAP method on a dataset describing the diagnosing of cardiac SPECT images, obtained from the University of California Machine Learning Archive (http://archive.ics.uci.edu/ml/machine-learning-databases/spect/SPECT.names) [104]. The dataset consists of 267 SPECT image sets, each image set corresponding to a single patient. The response variable is a binary classification of normal (0) or abnormal (1). A total of 22 binary feature patterns serve as the initial set of predictor variables.

Of the 267 SPECT image sets, 55 are classified as normal and 212 are classified as abnormal. The breakdown of data by classification after splitting the data into training and test sets is provided in Table 14.

Table 14: Composition of SPECT data for Example #2

Class	Training Set	Test Set	Full Set
0	42	13	55
1	171	41	212
Total	203	54	267

Problem Formulation

For Step 1: Data Analysis and Preprocessing, this dataset requires relatively little action. There is no time series component, requiring no feature extraction step. Likewise, there are no categorical features, requiring no encoding of categorical variables.

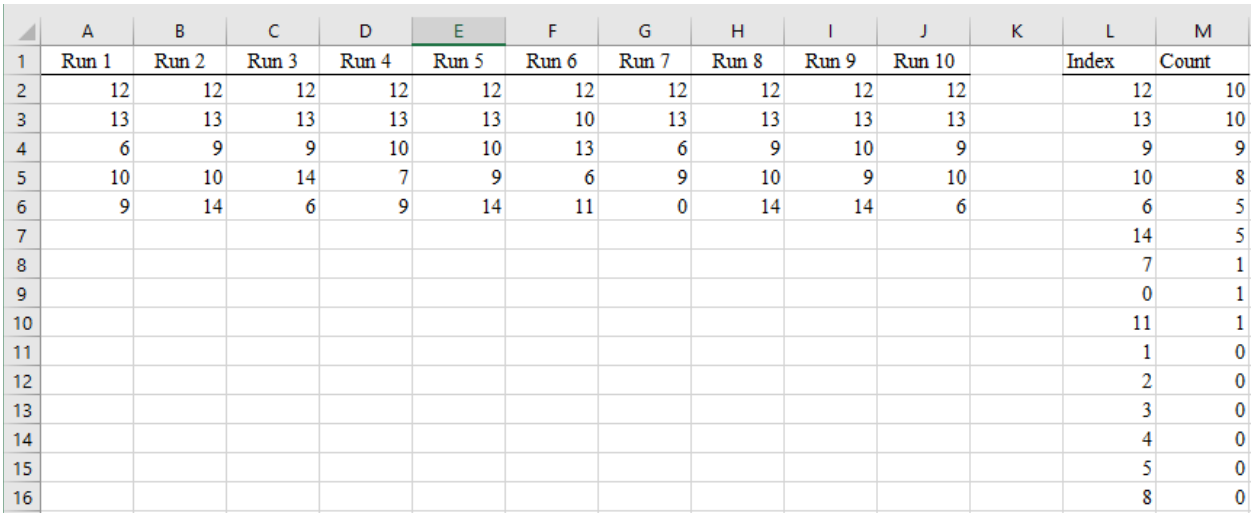
For Step 3, classification trees are selected as the machine learning technique, with default Python sklearn.tree.DecisionTreeClassifier() settings applied.

For Step 4, classification accuracy was employed as the metric for determining whether a subset falls into the High Quality group or the Low Quality group. Classification accuracy is defined as the number of correctly predicted test set records divided by the total number of test set records.

For 1,000 subsets containing five features per subset, the range of classification accuracies was approximately 0.6 to 0.85. Given these values, a threshold of 0.75 was set, where subsets whose solution's classification accuracy exceed 0.75 were placed in the High Quality group and those not were placed in the Low Quality group.

For Step 5, ten runs of N = 1,000 subsets with k = 5 features per subset were run, and the top five scoring features were recorded. This is largely the same as in Section 6.1, with the exception that the top 15 features in that example were truncated due to the large number of features remaining. FIS parameters such as membership functions and rules were unchanged.

Figure 15 contains a screen shot of a Microsoft Excel spreadsheet summarizing the results.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10		Index	Count
2	12	12	12	12	12	12	12	12	12	12		12	10
3	13	13	13	13	13	10	13	13	13	13		13	10
4	6	9	9	10	10	13	6	9	10	9		9	9
5	10	10	14	7	9	6	9	10	9	10		10	8
6	9	14	6	9	14	11	0	14	14	6		6	5
7												14	5
8												7	1
9												0	1
10												11	1
11												1	0
12												2	0
13												3	0
14												4	0
15												5	0
16												8	0

Figure 55: Consolidated FIS results

Summary of Results

Filter #1 reduced the initial set of 22 features to 15. After applying Filter #2 and FIS classification, six features stood out predominantly as consistently being scored by the FIS as among the top five features. Those six features are Features 6, 9, 10, 12, 13, and 14.

Table 15 summarizes the model performance for each of three cases: the original model with no feature reduction, the reduced feature set after Filter #1, and the final reduced feature set following the FIS.

Table 15: Summary of Model Performance

Model	Number of Features	Classification Accuracy	Relative Size	Relative Performance	PSR
Original	22	0.7778	1	1	1
Filter #1	15	0.7222	0.6818	0.9285	1.3618
Final	6	0.8148	0.2727	1.0476	3.8416

Using the original, unfiltered model as a baseline, Table 15 shows that the model, after applying Filter #1, displays a slight degradation in performance regarding classification accuracy but ends up with a higher PSR due to the reduced size required in the model. The final model produces a superior classification accuracy with 0.2727 the input size, resulting in a PSR of approximately 3.8416.

6.3 Example #3: Single Proton Emission Computed Tomography Image Features (SPECTF)

Description of Data

A third example uses the SPECTF dataset (http://archive.ics.uci.edu/ml/machine-learning-databases/spect/SPECTF.names), obtained using the same 267 SPECT images from Section 6.2. The difference being that, in this case, 44 continuous features instead of 22 binary features are extracted from each image. No change to the response variable. The SPECT image data that provides the source of the features in Section 6.2 and 6.3 was collected as part of an effort to semi-automate the diagnostic process associated with myocardial perfusion [149].

Problem Formulation

For Step 1: Data Analysis and Preprocessing, this dataset requires relatively little action. There is no time series component, requiring no feature extraction step. Likewise, there are no categorical features, requiring no encoding of categorical variables.

For Step 3, random forest is selected as the machine learning technique, with default Python sklearn.ensemble.RandomForestClassifier() settings applied. As with the previous two examples, the specific machine learning technique is not central to the model. Any algorithm for classification could have been chosen.

For Step 4, classification accuracy was again employed as the metric for determining whether a subset falls into the High Quality group or the Low Quality group.

For 1,000 subsets of five features per subset, the range of classification accuracies was approximately 0.62 to 0.87. Given these values, a threshold of 0.75 was set, where subsets whose solution's classification accuracy exceeds 0.75 are placed in the High Quality group and those not are placed in the Low Quality group.

For Step 5, ten runs of N = 1,000 subsets with k = 5 features per subset were run, and the top five scoring features were recorded. FIS parameters such as membership functions and rules are unchanged from the previous two examples.

Summary of Results

Filter #1 reduced the initial set of 44 features to 18. After applying Filter #2 and FIS classification, five features stood out predominantly as consistently being scored by the FIS as among the top five features. Those six features are Features 4, 5, 7, 9, 11, and 17.

Table 16 summarizes the model performance for each of three cases: the original model with no feature reduction, the reduced feature set after Filter #1, and the final reduced feature set following the FIS.

Table 16: SPECTF Example Results

Model	**Number of Features**	**Classification Accuracy**	**Relative Size**	**Relative Performance**	**PSR**
Original	44	0.8148	1	1	1
Filter #1	18	0.7407	0.4091	0.9091	2.2222
Final (Keep Best)	6	0.8703	0.1364	1.0681	7.8308
Final (Drop Worst)	10	0.8333	0.2273	1.0227	4.5

Using the original, unfiltered model as a baseline, Table 16 illustrates that the model after applying Filter #1 displays a slight degradation in performance. This is similar to the results observed in the SPECT example in Section 6.2. Examining the confusion matrices produced by the original model and the model after applying Filter #1, the difference amounts to four additional misclassified records. This near-attaining of the performance of the original model happens at an almost 60% reduction in the size of the feature set.

Two “final” models are defined and scored. The first keeps the best features, as identified by the FIS. This final model produces a slightly improved classification accuracy of 0.8703 with 0.1364 the input size, resulting in a PSR equal to approximately 7.8308. A second way to determine the final feature set to retain is to drop the worst-performing features as ranked by the FIS. This model produces a slightly superior classification accuracy of 0.8333, although the PSR is comparatively lower. This is an improvement over both the original feature set and equal to the performance of the set of best-performing FIS features. The PSR for the two “final” models favors the model obtained by keeping the best-performing features by virtue of the smaller size of its feature set. This result also demonstrates, for this dataset, that there is a middle-ground subset of features that are not classified as “worst” by the FIS but offer no added value to the overall scoring of the solution.

CHAPTER 7: APPLIED CASE STUDY

This chapter describes, implements, and discusses an applied machine learning case study to satisfy Research Objective #4 in the smart manufacturing environment, specifically in the context of electronic assembly of printed circuit boards (PCBs) as presented by Figure 16. Content from this chapter was incorporated into, "Predicting Contact-Without-Connection Defects on Printed Circuit Boards Employing Ball Grid Array Package Types: A Data Analytics Case Study in the Smart Manufacturing Environment", submitted to *International Journal of Data Science and Analytics* in March 2019.

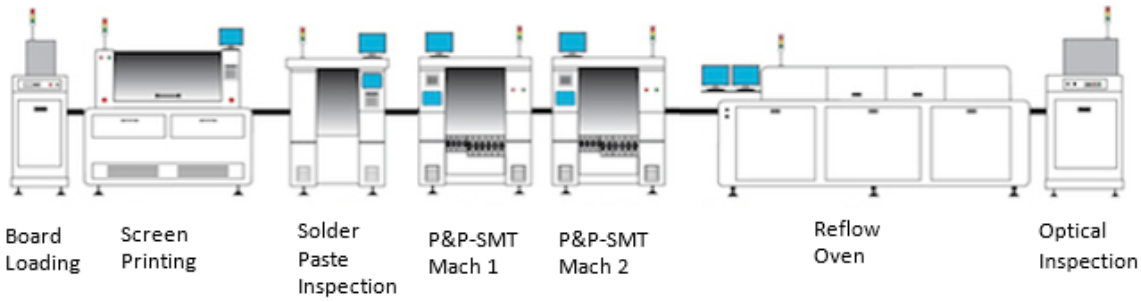


Figure 66: PCB assembly process, graphics adapted from [150]

7.1. Introduction to the Case Study

In PCB manufacture, a surface mount technology (SMT) line consists of several machines arrayed in series, with each machine performing either a mechanical step or an inspection function. The process begins with the application of solder paste to a clean PCB, employing a prefabricated stencil to ensure that correct quantities of solder paste are deposited in the correct places. Figure 17 (not to scale) illustrates a cross-section of this process, with solder paste wiped flush against the overlaid stencil.

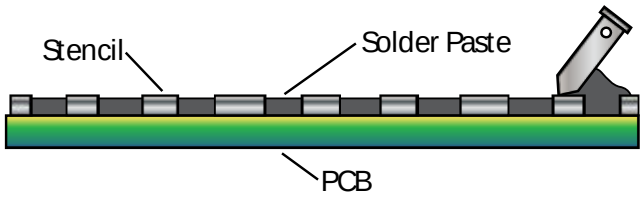


Figure 17: Solder Paste Application to PCB

After the stencil is removed, the solder paste deposits remain and a solder paste inspection (SPI) machine performs a visual inspection of the solder paste deposits, takes measurements of predefined parametric quantities, and compares those measurements against specifications that are particular to the type of PCB that is being produced. PCBs that meet specification continue along the line; PCBs failing to meet specification require a human operator to verify the data measurements and make the decision to advance the boards or reprocess them. Reprocessed boards are either reprinted or are removed from the line, washed, and reinserted at the start of the process. Time and cost associated with rework of defects identified at SPI are minimal, relative to those identified downstream in the production process. The time required to wash each board is a matter of seconds, and the cost associated with the lost solder paste is trivial. Figure 18, also not to scale, shows the same PCB cross-section, with the stencil removed and solder paste deposits remaining.

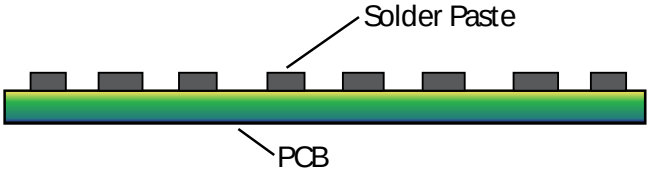


Figure 18: PCB and Solder Paste Deposits, After Removal of Stencil

After SPI, the boards move to a pick-and-place (P&P) machine that affixes components to the wet solder paste on the PCBs. Depending on the configuration of the line, there may be several P&P machines in series. After P&P, the boards moves to a reflow oven. The reflow oven activates flux to clean oxides and melts the solder spheres to form the mechanical and electrical connection between the components and PCBs.

The final station on the SMT line is an automated optical inspection (AOI) machine that checks quality of assembly and identifies visible defects such as missing or misaligned parts, solder volume, solder bridge, or insufficient solder. For a thorough orientation to SMT history, terminology, processes, and technology, see [151].

Boards passing AOI are designated as GOOD, while boards flagged for potential defects require a human operator to inspect. Those boards deemed acceptable by the operator are designated as PASS and advance indistinguishably from the GOOD boards. Boards deemed as defective are designated as FAIL and require rework to repair them. The time and cost associated with an AOI defect are typically orders of magnitude greater than those associated with an SPI defect. Whereas the time to correct an SPI-identified defect might be measured in seconds, it could take anywhere from minutes up to an hour of operator time to rework the failed board after the AOI stage depending on the complexity. This is because, by this point, the board has already passed through reflow and therefore cannot simply be reprinted or wiped clean and restarted.

After AOI, the boards move off the SMT line and are placed in a queue for in-circuit testing (ICT), which verifies electrical connectivity. Boards passing ICT are moved to final assembly (FA) and functional testing while boards not passing are troubleshot, diagnosed, reworked, and retested. As might be expected, the cost associated with a defect escaping from AOI to ICT could be orders of magnitude that what it might be had it been caught at AOI.

From a cost and rework perspective, it is always preferable to identify and repair defects as early in the process as possible, and the old military adage “bad news doesn’t get better with age” [152] is especially relevant for PCB manufacture. In one case study by Zarrow (1999), the cost associated with an undetected defect at downstream assembly stages was found to increase from $0.40 to $45. Figure 19 shows the cost increase as well the defect percentage at each PCB processing stage. The cost of defects increased by as much as 375% from SPI to AOI and another 3000% for a defect that escapes from AOI to FA [153]. Though dated, it is the only case study that was found that gives this cost estimation.

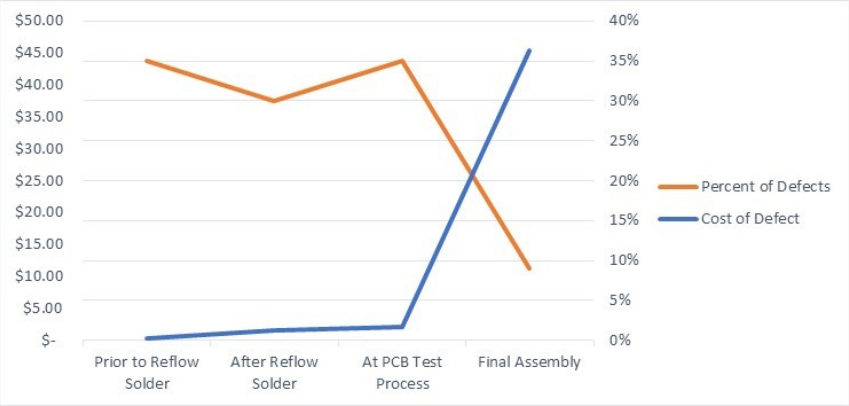


Figure 19: Cost and Percentage of Undetected Defects

The purpose of this research is to describe, implement, and discuss an applied machine learning case study in the smart manufacturing environment. The objective is to determine whether parametric data obtained by the optical SPI machine harbors hidden relationships that are indicators of downline defects at ICT. The scope of the case study is limited to ball grid array (BGA) package types. BGA packaging affixes components such as microprocessors to the PCB. Pins arrayed in a grid deposit small solder balls on the component that match a corresponding grid of solderable pads on the PCB. After connecting the component to the PCB at the P&P stage using the wet solder paste as an adhesive, as illustrated in Figure 20, the assembly passes through

a temperature profile in the reflow oven to activate the fluxes, melt the solder spheres, and form mechanical and electrical connection between the component and PCB.

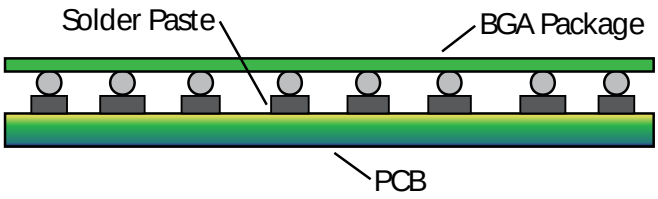


Figure 20: PCB with BGA Package, Prior to Reflow

BGA package types are of interest because, once the PCB has passed through reflow, it can be costly and require a higher skilled operator to rework defective components [154]. An especially challenging circumstance is the so-called "contact-without-connection" defect such as "head-in-pillow", in which there is enough contact for the unit to pass optical inspection and sufficient electrical connectivity for the component to function. Such a defect would pass AOI and may or may not fail ICT. However, throughout the life of the assembly, the component lacks mechanical strength and is high-risk to fail when encountering mechanical or thermal stress. In the worst-case scenario, these failures occur after distribution to the external customer.

Industry practitioners have explored the various potential contributing factors for BGA defects such as package warpage ([155], [156]), PCB and stencil printing parameters ([157], [158]), and solder paste or solder alloy composition ([156], [159], [160]).

In this case study, the process is a well-established, mature operation with defect rates far lower than one percent on average (M. Cvijetinovic, Senior Process Engineer, personal communication, November 29, 2018). As processes become increasingly mature with fewer and fewer defects, it becomes increasingly less cost effective to achieve reductions in the defect rate in terms of the resources required to lower the rates of defect occurrence. Machine learning

provides an opportunity to identify and harness previously unknown relationships to predict and preempt occurrences of downline defects. These relationships may then be utilized to establish a level of automation to help decision makers accelerate the discovery of core issues in the manufacturing process, while reducing the amount of resources in preparing datasets and infrastructure. The challenge herein presented conveys both academic and industrial interest.

The motivation for this case study, from an industry perspective, is the potential for cost savings in preempting ICT defects in BGA package types at the SPI stage. The broader impact of this work would enable practitioners to extend the results to correlate ICT defects to warranty returns, a case study for which will be presented in a subsequent research project.

There is also a purely academic benefit that is derived from this case study. As this chapter describes an exploratory case study, the collective body of knowledge is benefitted regardless of the result. A positive result is obviously preferable, for the practical benefits described above as well as for the potential to other researchers and practitioners to extend upon and improve the results. A negative result, however, is informative as well in that it identifies approaches not to pursue in future exploration of this important problem. Additionally, the chapter employs a novel modeling approach to extract features of interest to be used for model training that will be discussed in detail in Section 7.3. This adds a second academic interest in the outcome; any opportunity to employ a novel approach or extend an application beyond the context for which it was initially conceived adds to the collective body of knowledge and is worth exploring.

Two model formulations are contained in this chapter. The first formulation is a traditional machine learning approach, by which each solder paste deposit is treated as an individual record.

The second formulation aggregates all solder paste deposits on a single PCB location and analyzes them holistically.

The first formulation, described in Section 7.2, is focused primarily on the relative performance of models trained using FAFRAP-generated feature sets versus models trained using the full set of available features. It is less concerned with the standalone performance particular to defect prediction. In other words, if the full feature set produces a poor model but the FAFRAP-reduced feature set produces a slightly less poor model, then that result is pertinent to Section 5.1.

The second formulation, described in Section 7.3, employs an innovative approach to holistically analyze all solder paste deposits in a single location by applying a tool initially created for time series feature extraction to the arrays of solder paste parametric data. The focus, therefore, is twofold: to compare full and reduced feature set models; also, to analyze the utility of the novel modeling approach in predicting downline defects based on SPI parametric data.

7.2 Background and Related Work

It is believed that attempting to correlate SPI parametric data with downline electrical-testing defects has not previously been undertaken, and a brief survey of literature related to SMT manufacture of PCBs using BGA package types was conducted to confirm or deny this belief, with a specific focus on those that incorporate SPI data.

7.2.1 Survey Methodology

Keyword searches for relevant terms such as “BGA”, “ball grid array”, “SPI”, “solder paste inspection”, and “defect” were entered into academic databases ScienceDirect (SD), SpringerLink (SL), Engineering Village (EV), and Wiley (W). Search parameters were restricted

to journal articles or conference proceedings. Ten years was the rule of thumb for recency. Table 17 summarizes the searches with terms, source, total number of results, number of dated or irrelevant results, and number of discussed results.

Table 17: Search Summary

Index	Source	Search Terms	Total Results	Dated or Irrelevant Results	Relevant Results
S01	SD	"BGA", "SPI"	0	0	0
S02	SD	"BGA", "defect"	12	7	5
S03	SD	"SPI", "defect"	41	41	0
S04	SD	"BGA", "defect", "head-in-pillow"	0	0	0
S05	SD	"solder paste inspection", "defect"	1	0	1
S06	SL	"BGA", "defect", "solder paste inspection"	4	0	4
S07	SL	"BGA", "defect", "ball grid array"	9	6	3
S08	SL	"solder paste inspection", "ball grid array"	0	0	0
S09	EV	"BGA", "SPI", "defect"	1	0	1
S10	EV	"BGA", "defect", "head-in-pillow"	14	7	7
S11	EV	"solder paste inspection", "ball grid array"	7	7	0
S12	W	"ball grid array" "solder paste inspection", "defect"	1	1	0

7.2.2 Relevance Criteria

Table 1 lists counts for articles that were deemed either dated or irrelevant and articles that were selected for brief discussion to illustrate the nature of the related work. In some cases, articles were deemed irrelevant because the acronyms "BGA" and "SPI" contain more than one meaning. Search S03 is an example, where none of 41 results were deemed relevant because the "SPI" search criteria gathered results for "*Salmonella* pathogenicity island", "slow positron implantation spectroscopy", "soy-protein insulate", and the ETS transcription factors subfamily "SPI" in genetics. Search S05 modifies S03 by spelling out "solder paste inspection", with a total of one search result.

In deciding relevance, of primary consideration was any article directly attempting to correlate SPI parametric data with downline PCB defects. A secondary consideration was made

for articles discussing BGA defects and whether the context was in their identification, repair, or preemption.

7.2.3 Discussion of Related Work

Out of 21 articles deemed relevant from the 12 academic database searches, only one directly attempted to employ SPI parametric data with downline defects, using ANOVA-style statistical analysis to correlate SPI and PCB attribute data with defects identified at the AOI station [161]. The distinction between [161] and this research is that this research is concerned with ICT defects, not AOI defects. Additionally, the SPI data explored in this research is not attributional but rather parametric. Attributional SPI data might be information such as stencil thickness, solder paste type, or supplier. SPI parametric data is the body of measurements taken from the individual solder paste deposits.

The second type of articles considered was those discussing BGA defects in any manner whatsoever. Of the 20 articles in this group, eight focused on x-ray or other optical technology to detect existing defects ([162]–[169]), four explored the metallurgical properties of the solder ([156], [160], [170], [171]), three involved somewhat specific subsets of the BGA package type ([172]–[174]), three involved rework of BGA defects ([154], [155], [175]), and two focused on the stencil-printing process ([158], [176]). Table 18 summarizes the reviewed articles.

Table 18: Summary of Articles

Author(s)	*Article Title*	*Remarks*
Harter et al. (2016)	Comprehensive correlation of inline inspection data for the evaluation of defects in heterogeneous electronic assemblies [161]	SPI data to predict AOI defects
Bernard & Krastev (2008)	Modern 2D X-ray Tackles BGA Defects [162]	X-ray inspection for defect detection
Peng & Nam (2012)	Void defect detection in ball grid array X-ray images using a new blob filter [163]	X-ray inspection for void defect detection
Wang et al. (2014	Optik Microfocus X-ray printed circuit board inspection system [164]	X-ray inspection for select defect detection
Castellanos et al. (2014)	Head-in-Pillow X-ray Inspection [165]	X-ray inspection for head-in-pillow defect detection
Sumimoto et al. (2005)	Detection of defects of BGA by topography imaging [166]	X-ray inspection for BGA defect detection
Hui & Pang (2009)	Solder paste inspection using region-based defect detection [167]	Optical SPI inspection
Kuo et al. (2006)	Construction of 3D solder paste surfaces using multi-projection images [168]	Alternative to scanning to obtain SPI data
Chu & Pang (2007)	Solder paste inspection by special LED lighting for SMT manufacturing of printed circuit boards [169]	SPI technique
Scalzo (2009)	Addressing the challenge of head-in-pillow defects in electronics assembly [156]	Causes of BGA head-in-pillow defects
Pandher et al. (2010	Head-in-pillow defect – Role of the solder ball alloy [160]	Relation of solder properties to BGA head-in-pillow defects
Li et al. (2011)	Solder volume effects on the microstructure evolution and shear fracture behavior of ball grid array structure Sn-3.0Ag-0.5Cu solder interconnects [170]	Relation of solder metallurgical properties to BGA defects
Yang et al. (2014)	IMC growth and shear strength of Sn-Ag-Cu/Co-P ball grid array solder joints under thermal cycling [171]	Relation of solder metallurgical properties to BGA defects
Chiou et al. (2008)	The feature extraction and analysis of flaw detection and classification in BGA gold-plating areas [172]	Specific subset of BGA package type
Lee et al. (2017)	Temporal and frequency characteristic analysis of margin-related failures caused by intermittent nano-scale fracture of the solder ball in a BGA package device [173]	Specific subset of BGA package type
Lee & Park (2015)	Prediction enhancement of the J-lead interconnection reliability of land grid array sockets [174]	Specific subset of BGA package type
Wetterman (2017)	Top 5 BGA challenges to overcome [154]	BGA defect rework
Zhao et al. (2015)	Effects of package warpage on head-in-pillow defect [155]	BGA defect rework
Chen et al. (2014)	Characterization of after-reflow misalignment on head-in-pillow defect in BGA assembly [175]	BGA defect rework
Tsai & Liukkonen (2016)	Robust parameter design for the micro-BGA stencil printing process using a fuzzy logic-based Taguchi method [158]	Stencil-printing process
Yang & Tsai (2004)	A neurofuzzy-based quality-control system for fine pitch stencil printing process in surface mount assembly [176]	Stencil-printing process

A reasonable conclusion from the reviewed articles is that BGA defects are a known topic of interest among industry practitioners and that nearly all existing research on the topic is in detecting and reworking defects. Preempting these defects is largely unexplored, and the use of SPI parametric data to preempt electrical testing defects is an approach not previously attempted far as the authors can determine. The likely reason for the absence of research exploring the use

of SPI parametric data to preempt ICT defects is simply that the data may not be available. The data for this case study is available only due to a circa-2017 Industrial Internet of Things (IIOT) capability enhancement project at the manufacturing facility from which the data is obtained.

7.3 Model Formulation #1 – By Solder Paste Deposit

7.3.1 Description of Data

The data for this case study consists of 16 features and three binary response variables, corresponding to the three inspection stations that a PCB must pass through: AOI, ICT, and FA. The fifteen features are a mixture of continuous and categorical data types. Table 17 provides an overview and description of each of the 16 features.

Table 19: Feature descriptions

Feature	Data Type	Description
F00	Categorical	Barcode – Unique designator associated with a single panel, upon which several PCB modules might exist, each module to be manufactured into a single PCB
F01	Categorical	Location reference ID for each specific location on a single PCB
F02	Integer	Designator for a module on a panel; the combination of F00, F01, and F02 allow precise mapping of SPI parametric data with defect(s) identified at AOI, ICT, and FA
F03	Categorical	Component pin number – Designates a specific pin and solder paste deposit or joint
F04	Integer	Component pin size (x-axis)
F05	Integer	Component pin size (y-axis)
F06	Float	Solder paste offset (x-axis) – Deviation of solder paste deposit from target location as per specifications; parametric data
F07	Float	Solder paste offset (y-axis) – Same as F06 but considering the y-axis
F08	Integer	Pad stencil height
F09	Float	Solder paste deposit volume, as a percentage of component-specific benchmark
F10	Float	Solder paste deposit height
F11	Float	Solder paste deposit area, as a percentage of component-specific benchmark
F12	Categorical	Component part number – A unique designator for any component placed on PCB

F13	Float	Stencil surface area ratio
F14	Categorical	PCB name – A unique designator for type of PCB produced; not a unit serial number designation
F15	Categorical	Package type – Identifier for specific BGA package employed
R01	Integer	Binary response variable equal to 1 if an AOI defect is discovered on the F00-F01-F02 combination and 0 if it is not
R02	Integer	Binary response variable equal to 1 if an ICT defect is discovered on the F00-F01-F02 combination and 0 if it is not
R03	Integer	Binary response variable equal to 1 if a FA defect is discovered on the F00-F01-F02 combination and 0 if it is not

Six models were created and run using six different datasets, with datasets formed by conditioning upon the levels of feature F14 and using R02 as the response variable. This was due to the sparsity of defect data available in R01 and R03. Summary information for each dataset is provided in Table 20. Note that the counts for records and defects denote individual solder paste deposits. This means that the counts in the three defects columns are not indicative of the number of defective products but rather the number of solder deposits that are present on a defective product. This topic will receive additional treatment in Section 7.3.3 Result and Discussion.

Table 20: Dataset summary information

Index	Number of Records	ICT Defects
DS01	187264	2144
DS02	1058852	9811
DS03	1146089	153431
DS04	1004748	15332
DS05	1114966	53270
DS06	1070664	22000

7.3.2 Model Formulation

Random Forest models were created, using Python's sklearn RandomForestClassifier() [143] default settings with the exception that the number of trees in each forest was raised to 20

from the default setting of 10. Categorical features were encoded using Ordinal Encoding with Python's category encoders library[1].

Because of the applied nature of this case study, it is preferable for metrics of interest to have seamless interpretation that requires little modification or translation when briefed to decision makers at various levels of the organization. For this reason, four metrics are worth considering: accuracy, precision, recall, and f1 score.

Table 21: Confusion Matrix Template

	Actual – Non-Defective	***Actual – Defective***
Predicted – Non-Defective	A	B
Predicted – Defective	C	D

To illustrate the distinction between the various metrics, consider the confusion matrix template in Table 21. Quantity A refers to those records which are correctly classified as defective. Quantity B represents records incorrectly classified as non-defective. A record in this category can be thought of as an "escape", where the model fails to predict that a defect is present. These records incur a cost, as discussed in Section 7.1.

Quantity C represents records that the model classifies as defective but are not defective. A record in this category might be thought of as a "false alarm", which incurs unnecessary costs from the parts and labor associated with diagnosing, troubleshooting, or replacing defects. Finally, Quantity D represents records that are correctly classified by the model as defective.

Accuracy is the proportion of correctly classified records and is computed using Equation (4). If the dataset is highly unbalanced, with only a tiny fraction of records exhibiting a defect,

[1] For documentation and definitions of encoding techniques, see http://contrib.scikit-learn.org/categorical-encoding/

then accuracy may not be the ideal metric because the model may simply predict all records to be non-defective.

$$Accuracy = \frac{A + D}{A + B + C + D} \quad (4)$$

Precision is accuracy conditioned upon a specific prediction and, for defective records, is computed using Equation (5). If the cost associated with a false alarm is prohibitive, then precision might be an appropriate metric.

$$Precision = \frac{D}{B + D} \quad (5)$$

Recall is a metric that quantifies the proportion of total defects that the model flags; a high recall score indicates that the model does a good job of preventing escapes. As escapes typically incur increasing levels of cost the longer that they are undetected, recall is an important metric to consider. Recall is computed using Equation (6).

$$Recall = \frac{D}{C + D} \quad (6)$$

The decision to use precision or recall boils down to the relative cost associated with an escape versus a false alarm.

Finally, f1 score is computed by calculating the harmonic mean of precision and recall. If the desire is to keep both escapes and false alarms to a minimum, then f1 score is a possible metric as it captures the elements of precision and recall in a single metric.

7.3.3 Results and Discussion

In all six scenarios, the models produced using the feature reduction and prioritization framework produced results comparable to models produced using the full feature set. Table 22 summarizes each model's performance.

Table 22: Summary of results

Dataset	Relative f1 score	Relative Size	Top FAFRAP Features
DS01	1.0144	0.556	F02, F03, F07, F09, F10
DS02	1.0237	0.875	F01, F03, F09, F10, F11, F12, F15
DS03	2.201	0.625	F03, F05, F07, F13, F15
DS04	0.9083	0.40	F01, F03, F09, F10
DS05	0.9632	0.3636	F01, F07, F09, F11
DS06	0.9268	0.3846	F07, F09, F10, F11, F12

As indicated in Table 22, all FAFRAP models have an f1 score of at least 0.90 of the f1 score obtained by the full model, with that performance achieved using no more than 0.875 the full-size model. In three models, the f1 score improves with the reduced set of features, and in three models the f1 score is slightly degraded. The tradeoff for the reduced model performance is in the model size.

As previously discussed, this research is not explicitly geared towards answering the underlying question of what if any predictive power lies in SPI parametric data to predict downline defects. Rather, it is to make an apples-to-apples comparison of models produced by the FAFRAP framework versus models produced by the full set of available features.

The initial results indicate at least the possibility that underlying relationships exist that can be exploited to reduce solder, process, or testing defects. Notably, the models consistently prioritize parametric data associated with solder paste offset, solder paste height, and solder paste

area or volume. This is consistent with subject matter expert expectations. With this initial insight, one can foresee an automated application to generate early-process recommendations which would guide the operators to make the necessary corrective actions on the process in real-time.

A limitation of the first model formulation is in the unbalanced nature and the potential for the model's predictive power to be exaggerated due to the data records being generated by solder paste deposit and not by panel. For example, DS01 shows 2144 ICT Defects in Table 18; this does not mean that 2144 printed circuit boards in DS01 had an ICT defect. Rather, it means that there were 2144 solder paste deposits on printed circuit boards that went on to have an ICT defect identified. Likely, there were no more than a handful of PCBs with an identified defect. The reason for this formulation is the desire to identify hidden relationships within this highly mature process, where state-of-the art test machines cannot reach with optical inspection. The caution is simply not to forget that model results and corrective actions must include an added step to translate the by-solder-paste-deposit results back to the PCB level. This step is outside the scope of this section but is certainly crucial from an application and automation standpoint.

7.4 Model Formulation #2 – By PCB Location

7.4.1 Description of Data

The data for this formulation consists of five parametric features measured at SPI and defect data extrapolated from ICT. The five features of interest are:

- X01: Deviation of solder paste deposit from target location (x-direction).
- X02: Deviation of solder paste deposit from target location (y-direction).
- X03: Solder paste deposit volume, as a percentage of component-specific benchmark.

- X04: Solder paste deposit height.
- X05: Solder paste deposit area, as a percentage of component-specific benchmark.

Defect data consists of a binary response variable, equal to 1 if a defect is detected at ICT and 0 of not.

To organize the data, the following additional features were collected but not incorporated into any predictive models as input variables:

- F01: Unique barcode designator associated with a single panel, upon which several PCB modules might exist. Each module on a panel will be manufactured into the same style PCB.
- F02: Location reference ID for modules on a panel.
- F03: Location reference ID for component location on a PCB.
- F04: Unique designator for a specific component pin, which creates a specific solder paste deposit. This feature is used for sorting purposes in organizing the data for model inclusion.

The concatenation of each unique F01_F02_F03 combination provides a distinct location for a component, within a certain module, at a certain position on the panel. This information allows a specific defect to be mapped to specific records of the five parametric features.

Two datasets were collected, each containing data for a single type of PCB Assembly. The first dataset, DS01, contains smart manufacturing data collected from an approximately six-month period from April 2018 through September 2018. This circuit board is not sold as a standalone unit but rather mates to several other circuit boards in the final assembly.

DS01 was selected for this research because it is a high-volume product, which hopefully affords sufficient records to train a predictive model. It is not a foregone conclusion that SPI

parametric data harbors any predictive relationship whatsoever; the best chances for extracting such a relationship, if it exists, is in higher-volume datasets.

The second dataset, DS02, represents a different PCB assembly. This product is the only board in the final assembly product, but it merges with other components in final assembly. As with the PCB assembly in DS01, this this PCB assembly is not sold as a standalone unit. Data in DS02 was obtained in the same time window as DS01.Table 23 summarizes the two datasets.

Table 23: Data Summary

Name	Records	Total Defects	Training Set Non-Defect	Training Set Defects	Test Set Non-Defect	Test Set Defects
DS01	4421	233	3359	177	829	56
DS02	6916	41	5500	32	1375	9

As shown in Table 23, each dataset is highly unbalanced, with a fraction of the overall records exhibiting an identified defect. This presents the immediate concern that there may not be sufficient defect data for model training. Of the two datasets, DS01 might have the better prospects at first glance due to the higher proportion of defective records.

7.4.2 Model Approach

The nature of the available data presents some challenges associated with a traditional modeling approach in that defect information is available only to the precision of the PCB location, but SPI parametric data is available to the precision of the BGA pin level. This means that there could be scores or even hundreds of records associated with a single defect. The concern from a data perspective is that solder paste deposits that are fully within specification would be assigned a defective outcome in the training set if they reside in the same PCB location

as an identified defect. This would be incorrect and could lead to confusion with interpretation of model results. A second concern, this time from a process perspective, is that a test already exists to test the individual solder paste deposits for conformity to specifications; the traditional approach to model line by line at the record level would be redundant, unless the goal is to validate the specifications, which is a different analysis altogether. Finally, and most concerning, the traditional approach of examining each record in isolation fails to capture the interaction between parametric data from different pins on the same location.

An alternative approach, employed in this model formulation, is to consider all pins in a location simultaneously by creating an array and sorting it in some fixed manner for each of the five features. The model is indifferent to the method of sorting, so long as the arrays are sorted in the same way every time. The default for this case study was to sort by F04 alphanumerically.

Using this approach, each record would be assigned a label of 0 or 1 depending on whether a defect was identified at that location, and that label would map to five arrays, one for each of the five parametric SPI features. The uniqueness of this approach is in applying the TSFRESH time series feature extraction algorithm to each of the five arrays, thus generating up to 794 * 5 = 3970 features for each record. Even though TSFRESH is designed with time series in mind, there is no reason why the features that it generates could not be applied to problems in other contexts. The FAFRAP framework would then be applied to those 3970 features.

Decision trees were selected as the machine learning algorithm to provide the maximum clarity to the interpretation of results. If model results are encouraging, then decision trees present the greatest flexibility to clearly identify rules or thresholds to track or visualize as PCBs are run through the SMT lines in real time.

7.4.3 Results and Discussion

Results from this case study fall into two domains of interest. The first domain is the bottom-line, direct answer to the defect prediction question. The second domain is the application of FAFRAP to the case study and any framework-specific lessons that might be learned.

Table 24 summarizes the confusion matrices for each dataset at each level of filtration in the FAFRAP framework. The results indicate comparable results from one level of the framework to the next, suggesting that the features being removed make little to no contribution to the overall model.

Table 24: Aggregated model predicted results

Dataset	FAFRAP Filtration Level	Number of Features	Correctly Predicted Non-Defective	Incorrectly Predicted Non-Defective (Escape)	Correctly Predicted Defective	Incorrectly Predicted Defective (False Alarm)
DS01	None	3970	808	32	24	21
DS01	Filter #1	2210	810	33	23	19
DS01	Final	15	807	34	22	22
DS01	Best Accuracy	9	817	21	35	12
DS01	Best Precision	9	821	29	27	8
DS01	Best Recall	7	805	21	35	24
DS01	Best F1	9	817	21	35	12
DS02	None	3970	1370	6	3	5
DS02	Filter #1	901	1372	7	2	3
DS02	Final	15	1371	5	4	4
DS02	Best Accuracy	8	1375	4	5	0
DS02	Best Precision	8	1375	4	5	0
DS02	Best Recall	7	1367	3	6	8
DS02	Best F1	8	1375	4	5	0

Table 25 contains scoring metrics, selected because they each carry a clear physical interpretation. Any one of them may be the metric of choice depending on the situation. The four metrics described in Section 7.3.2 are calculated and displayed: accuracy, precision, recall, and F1 score.

Table 25: Model comparison at FAFRAP levels

Dataset	Filter Level	Number of Features	Accuracy	Precision	Recall	F1 Score
DS01	None	3970	0.9401	0.5333	0.4286	0.4752
DS01	Filter #1	2210	0.9412	0.5476	0.4107	0.4694
DS01	Final – Top 15	15	0.9367	0.5	0.3929	0.44
DS01	Best Accuracy	9	0.9627	0.7447	0.6250	0.6796
DS01	Best Precision	9	0.9582	0.7714	0.4821	0.5934
DS01	Best Recall	7	0.9492	0.5932	0.625	0.6087
DS01	Best F1	9	0.9627	0.7447	0.6250	0.6796
DS02	None	3970	0.9921	0.375	0.3333	0.3529
DS02	Filter #1	901	0.9928	0.4	0.2222	0.2857
DS02	Final – Top 15	15	0.9935	0.5	0.4444	0.4706
DS02	Best Accuracy	8	0.9971	1.0	0.5556	0.7143
DS02	Best Precision	8	0.9971	1.0	0.5556	0.7143
DS02	Best Recall	7	0.9921	0.4286	0.6667	0.5217
DS02	Best F1	8	0.9971	1.0	0.5556	0.7143

Note that the number of features in the ‘best’ rows are subsets of the 15 features in the Final – Top 15 row. The final step in the FAFRAP framework, after identifying some number of top performing features, is to combinatorically iterate through all possible subsets of the final list. Note the optimal subset of features is not necessarily the same for every metric. For DS01, the same optimal subset gives the best accuracy and F1, but a different subset gives the best precision and the best recall. For DS02, the same subset of features gives optimal values for all metrics except for recall.

The first point of discussion is that the exploratory nature of this problem has revealed that there do appear to be relationships hidden within SPI parametric data that can be harbingers of downline ICT defects. The use of the TSFRESH package to extract as diverse a feature set as possible resulted in models that achieve encouraging results. For DS01, the top accuracy achieved was 0.9627, the best precision achieved was 0.7714, the best recall achieved was 0.625, and the best f1 score achieved was 0.6796. For DS02, results are comparable and slightly superior for each metric.

While these initial results are encouraging, the limited scope of this initial case study makes it necessary to build and run models on increasingly diverse datasets before broad conclusions can be drawn. This will produce two benefits. The first benefit is continued validation or confirmation of the featured modeling approach. The second benefit is added depth of knowledge regarding which of the 3,970 TSFRESH features continue to be selected for the best producing models. It is not known if the underlying relationships correlate to only a few TSFRESH features or many of them. If subsequent model results continue to gravitate towards the same set of best features, then those features can possibly be the centerpiece for development of metrics for data tracking and visualization.

A second, related point, is that the specific features calculated by TSFRESH and retained in the final models require additional analysis and interpretation by subject matter experts. Because these features were extracted by a tool initially conceptualized for time series, the physical interpretation of the features will likely have little direct transfer to this manufacturing process. If a detailed drill-down can be made into those specific features, it will inform decisions as to how best to proceed regarding utilization of these model results in a practical sense. If there is no direct connection between any element of the physical process, it may be best for practical

application of this model simply to extract the features real-time as boards come through the SPI machine and execute the trained models to obtain a prediction. If, however, there is a direct connection to some tangible physical piece, then there are more options. Metrics can be created and visualized via some dashboard or other means. Then, instead of executing a trained model, the metric is simply monitored, with corrective action taken if it exceeds the identified threshold.

A final point relevant to the first domain is that, depending on the continued performance of this modeling approach, the decision on how to apply the results requires exploration into the nontrivial question of where, architecturally, to perform this analysis with respect to the data infrastructure. In smart manufacturing, it is possible to perform analysis offline, at the local machine level, at the cloud level, or at some intermediate edge level. In this case study, there is a sharp increase in the cost associated with rework after the board passes through reflow. For this reason, it is ideal for model predictions to be made prior to that stage. The best-case scenario is for model predictions to be made before the board enters the P&P machine. This, however, gives a window of approximately 30 seconds to five minutes for an analysis container to receive the SPI parametric data, extract the TS features, and execute the trained model using the appropriate feature subset before the board enters the P&P machine. This necessitates edge-level computing because by the time data migrates to the cloud level and predictions are routed back, the window of opportunity to take corrective action will have likely passed.

With respect to the second domain, the results are encouraging and consistent with previous validation tests in [141]. The filtration mechanisms in FAFRAP worked as intended and consistent with what might be expected from this case study. Because the TSFRESH features were initially conceived to be extracted in a time-series context, there was the expectation that many of the extracted features would have little or no relationship to ICT defects. However, the

question that the case study explored was if there were underlying or hidden factors that were systemically contributing to defects. If that is the case, then it is natural to expect that that factor might appear in the behavior of at least one of the 794 features extracted by the software.

Given these expectations, the FAFRAP filter results make intuitive sense. In DS01, the first filter reduced the feature set from 3970 to 2210. It is unlikely that all 2210 features individually exhibiting statistical dependence with ICT defect identification do so based on a factor that is distinct from all the others. Rather, some underlying factor might be manifesting itself in many or all those 2210 features. The final feature set bears this out because the model produced by the final feature set scores comparably to the models at previous filtration levels for each of the metrics computed. In some cases, the reduced feature set performed better. Similar reasoning may be applied to the DS02 results

It is also of value to consider the input data required to make defect predictions and associated follow-on decisions. Models involving large datasets require more time to train than models involving smaller datasets, and the difference in time required might be the difference between performing the calculations at the edge or the cloud computing level. This can translate to decisions regarding what data is designated for long-term cloud storage and what data can be discarded. The FAFRAP framework can assist analysts in making decisions regarding the capture and storage of smart manufacturing data based on its relative value in answering the important analytical questions facing the organization.

CHAPTER 8: CONCLUDING REMARKS AND FUTURE RESEARCH

The novel Fuzzy Approach to Feature Reduction and Prioritization (FAFRAP) approach presented in this dissertation attempts to rank and classify potential predictor variables according to their value for use in training machine learning models for some particular problem of interest. It provides a quantitative metric that is suitable for feature ranking and it also provides a qualitative description that assists human data scientists and decision makers in understanding the nature of the feature's usefulness. This information has immediate value in building problem-specific models and it also possesses a knowledge management value that, when developed, can inform long term decisions in data capture, storage, and retention.

This research contributes to the body of knowledge by integrating the diverse techniques and disciplines of statistical independence tests, applied machine learning, and fuzzy inference systems in such a way that, as yet, has not been undertaken previously. Additionally, the applied case study described in Chapter 7 contributes twofold: it validates the FAFRAP method in a real-world context and it employs the novel problem approach by treating the arrays of SMT-SPI parametric data as time series for the purpose of feature extraction. To the knowledge of the author, this is a unique approach to the problem of defect prediction in electronic assembly manufacture.

The approach employed in this research to model SPI parametric data holistically by PCB location and extract features using a Python library initially intended for time series has been shown to produce results that merit continued exploration of the approach. Additionally, the use of the FAFRAP algorithm to reduce and prioritize features performed as intended. From these perspectives, the case study might cautiously be deemed a tentative success, while leaving room for scalability to other manufacturing case examples.

However, additional work is necessary to truly harness the possible benefits from this approach, and additional knowledge is needed to maximize the potential of the FAFRAP. Thus, future research may be compartmentalized into that which applies specifically to the SPI case study and that which applies specifically to the FAFRAP.

With respect to the applied case study, the first area of additional work is in the interpretation of results. This study intentionally employed decision trees because they provide clear interpretability. Splits in the trees can be programmed into data tracking or visualization tools, and dashboards that correspond to the model can be created and monitored in lieu of continually computing the extracted features and running the models. However, the tradeoff in using decision trees is that they typically underperform other algorithms such as random forest or deep neural networks. This is a known, intentional limitation and tradeoff in the approach taken in this dissertation.

Therefore, it is necessary to analyze the specific features highlighted by FAFRAP, identify their physical interpretations as applied to time series problems, and determine what if any physical interpretation can be extended to manufacturing. This will inform subsequent decisions on how to use these models, any KPIs that may be necessary to create, or any existing KPIs to track.

The second area of additional work is in generalizability of results. The study should be performed again on additional datasets with a different mix of PCBs. Not only would it be value added to use additional classes (models) of PCBs, but there are reasons to condition datasets on factors other than PCB type. For example, each PCB contains a variety of components. Conditioning on component part number instead of PCB ID may allow better generalizability of the results.

With respect to FAFRAP, a known area of future research is the tuning of its hyperparameters, as alluded to in Chapter 5. This study revealed an area of potential improvement associated with the hyperparameters related to grouping the features into high-performing and low-performing solutions. There is no known technique to determine the minimum number of subsets to generate so that there is sufficient representation of each feature for the FIS to effectively rank it. Up to this point, the analyst's discretion has been used, with the rule of thumb being that every feature should appear in at least 20 subsets and the bottom-line model result being the discriminant. However, additional knowledge would speed up the process and free the data scientist from performing redundant work.

A second hyperparameter for which a general principle would be helpful is where to set the threshold for what constitutes a "top" feature. In this case study, there were 2210 features that survived the first filter and were input into the FIS. The threshold was tested at different values such as 50, 100, or 200, with the guiding principle being within the top 10%.

Finally, continued replication of the study is necessary to identify if the same features highlighted by FAFRAP will continue to be prioritized in subsequent models using different PCBs or component parts. If the true underlying factor is not directly calculated by any of the time-series-style features, which is a reasonable working hypothesis, then it is possible that the algorithm will have many different "final" feature sets that all achieve comparable results.

In conclusion, the approach outlined in the preceding section could be whimsically described as a "poor analyst's deep learning", where the TSFRESH-extracted features operate conceptually similarly to a single hidden layer of neurons. Like deep learning models, this approach transforms the input features into some number of new features. Unlike deep learning models, those new features can be identified and interpreted precisely because they correspond to

some known calculation or physical interpretation. Also, unlike deep neural networks, model training is relatively straightforward and the lack of successive nonlinear transformations between hidden layers gives weaker prospects for uncovering hidden relationships in the data. The tradeoff between the two approaches is one of model performance for interpretability.

REFERENCES

[1] Rockwell Automation, *The Connected Enterprise eBook: Bringing People, Processes, and Technology Together*. Rockwell Automation, 2015.

[2] W. Otieno, M. Cook, and N. Campbell-Kyureghyan, "Novel approach to bridge the gaps of industrial and manufacturing engineering education: A case study of the connected enterprise concepts," in *2017 IEEE Frontiers in Education Conference (FIE)*, 2017, vol. 2017–Octob, no. November, pp. 1–5.

[3] S. J. Qin, "Process data analytics in the era of big data," *AIChE J.*, vol. 60, no. 9, pp. 3092–3100, Sep. 2014.

[4] S. K. Gill, P. Nguyen, and G. Koren, *Adherence and tolerability of iron-containing prenatal multivitamins in pregnant women with pre-existing gastrointestinal conditions*, vol. 29, no. 7. 2009.

[5] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information.," *Psychol. Rev.*, vol. 63, no. 2, pp. 81–97, 1956.

[6] H. a. Simon, "Designing organizations for an information-rich world," *Comput. Commun. public Interes.*, vol. 72, p. 37, 1971.

[7] M. J. Adler and C. Van Doren, *How to Read a Book: The Classic Guide to Intelligent Reading*. New York: Simon and Schuster, 1972.

[8] E. L. Kaufman, M. W. Lord, T. W. Reese, and J. Volkmann, "The Discrimination of Visual Number," *Am. J. Psychol.*, vol. 62, no. 4, pp. 498–525, 1949.

[9] D. Mourtzis, E. Vlachou, and N. Milas, "Industrial Big Data as a Result of IoT Adoption in Manufacturing," *Procedia CIRP*, vol. 55, pp. 290–295, 2016.

[10] D. Bollier and C. M. Firestone, *The Promise and Peril of Big Data*. 2010.

[11] Q. P. He and J. Wang, "Statistical process monitoring as a big data analytics tool for smart manufacturing," *J. Process Control*, 2017.

[12] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, Aug. 2014.

[13] "Everything You Need to Know About the Industrial Internet of Things," *G.E. Digital*, 2016. [Online]. Available: https://www.ge.com/digital/blog/everything-you-need-know-about-industrial-internet-things. [Accessed: 01-May-2018].

[14] A. Gilchrist, *Industry 4.0*. Berkeley, CA: Apress, 2016.

[15] S. Schneider, "THE INDUSTRIAL INTERNET OF THINGS (IIoT)," in *Internet of Things and Data Analytics Handbook*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2016, pp. 41–81.

[16] "Industrial Internet Consortium." [Online]. Available: https://www.iiconsortium.org/. [Accessed: 02-May-2018].

[17] "OpenFog." [Online]. Available: https://www.openfogconsortium.org/. [Accessed: 02-May-2018].

[18] A. Kusiak, "Smart manufacturing must embrace big data," *Nature*, vol. 544, no. 7648, pp. 23–25, 2017.

[19] Rockwell Automation, “The Connected Industrial Enterprise,” 2015.

[20] “Benefits of cloud computing,” *Queensland Government: Business Queensland*, 2017. [Online]. Available: https://www.business.qld.gov.au/running-business/it/cloud-computing/benefits. [Accessed: 30-Jul-2018].

[21] Rockwell Automation, “The Connected Enterprise Maturity Model,” p. 12, 2014.

[22] P. LaCasse, W. Otieno, and F. Maturana, “A Survey of Feature Set Reduction Approaches for Predictive Analytics Models in the Connected Manufacturing Enterprise,” *Appl. Sci.*, vol. 9, no. 5, p. 843, 2019.

[23] J. Lenz, T. Wuest, and E. Westkämper, “Holistic approach to machine tool data analytics,” *J. Manuf. Syst.*, vol. 48, pp. 180–191, 2018.

[24] K. Thoben, S. Wiesner, and T. Wuest, “‘ Industrie 4 . 0 ’ and Smart Manufacturing – A Review of Research Issues and Application Examples,” vol. 11, no. 1, 2017.

[25] A. Oussous, F. Benjelloun, A. Ait Lahcen, and S. Belfkih, “Big Data technologies: A survey,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 30, no. 4, pp. 431–448, Oct. 2018.

[26] N. Honest, “A Survey of Big Data Analytics,” *Int. J. Inf. Sci. Tech.*, vol. 6, no. 1/2, pp. 35–43, Mar. 2016.

[27] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V Vasilakos, “Big data analytics: a survey,” *J. Big Data*, vol. 2, no. 1, p. 21, Dec. 2015.

[28] N. Spangenberg, M. Roth, and B. Franczyk, “Evaluating new approaches of big data analytics frameworks,” in *International conference on business information systems*, 2015, no. June.

[29] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, “Machine learning in manufacturing: advantages, challenges, and applications,” *Prod. Manuf. Res.*, vol. 4, no. 1, pp. 23–45, 2016.

[30] D. J. Dingli, “The Manufacturing Industry – Coping with Challenges (Working Paper No . 2012 / 05),” *Working Papers from Mastricht School of Management*, 2012. [Online]. Available: https://econpapers.repec.org/paper/msmwpaper/2012_2f05.htm. [Accessed: 27-Feb-2018].

[31] J. Gordon and A. S. Sohal, “Assessing manufacturing plant competitiveness - An empirical field study,” *Int. J. Oper. Prod. Manag.*, vol. 21, no. 1/2, pp. 233–253, 2001.

[32] L. E. Shiang and S. Nagaraj, “Impediments to innovation: Evidence from Malaysian manufacturing firms,” *Asia Pacific Bus. Rev.*, vol. 17, no. 2, pp. 209–223, 2011.

[33] A. J. Thomas, P. Byard, and R. Evans, “Identifying the UK’s manufacturing challenges as a benchmark for future growth,” *J. Manuf. Technol. Manag.*, vol. 23, no. 2, pp. 142–156, 2012.

[34] S. B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques,” *Informatica*, vol. 31, pp. 249–268, 2007.

[35] K. Yang and J. Trewn, *Multivariate statistical methods in quality management*. New York: McGraw-Hill, 2004.

[36] E. Alpaydin, *Introduction to Machine Learning*, 3rd ed. Cambridge, MA: MIT Press, 2014.

[37] S. Doltsinis, P. Ferreira, and N. Lohse, “Reinforcement learning for production ramp-up: A Q-batch learning approach,” *Proc. - 2012 11th Int. Conf. Mach. Learn. Appl. ICMLA 2012*, vol. 1, pp. 610–615, 2012.

[38] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, “Deep learning for smart manufacturing: Methods and applications,” *J. Manuf. Syst.*, pp. 1–13, 2018.

[39] F. Tao, Q. Qi, A. Liu, and A. Kusiak, “Data-driven smart manufacturing,” *J. Manuf. Syst.*, 2018.

[40] B. Butler, “What is edge computing and how it’s changing the network,” *Network World*, 2017. [Online]. Available: https://www.networkworld.com/article/3224893/internet-of-things/what-is-edge-computing-and-how-it-s-changing-the-network.html. [Accessed: 07-Mar-2018].

[41] D. Linthicum, “Responsive Data Architecture for the Internet of Things,” *Computer (Long. Beach. Calif).*, vol. 49, no. 10, pp. 72–75, 2016.

[42] R. Mahmud, R. Kotagiri, and R. Buyya, “Fog Computing: A Taxonomy, Survey and Future Directions,” in *Internet of Everything*, B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, Eds. Springer Singapore, 2018, pp. 103–130.

[43] G. Andreadis, “A collaborative framework for social media aware manufacturing,” *Manuf. Lett.*, vol. 3, pp. 14–17, 2015.

[44] D. Mourtzis, “Internet based collaboration in the manufacturing supply chain,” *CIRP J. Manuf. Sci. Technol.*, vol. 4, no. 3, pp. 296–304, 2011.

[45] D. Mourtzis, M. Doukas, and C. Vandera, “Mobile apps for product customisation and design of manufacturing networks,” *Manuf. Lett.*, vol. 2, no. 1, pp. 30–34, 2013.

[46] J. Lee, E. Lapira, B. Bagheri, and H. an Kao, “Recent advances and trends in predictive manufacturing systems in big data environment,” *Manuf. Lett.*, vol. 1, no. 1, pp. 38–41, 2013.

[47] K. D. Bouzakis, G. Andreadis, A. Vakali, and M. Sarigiannidou, “Automating the manufacturing process under a web based framework,” *Adv. Eng. Softw.*, vol. 40, no. 9, pp. 956–964, 2009.

[48] C. M. Flath and N. Stein, “Towards a data science toolbox for industrial analytics applications,” *Comput. Ind.*, vol. 94, pp. 16–25, 2018.

[49] A. Kumar, R. Shankar, A. Choudhary, and L. S. Thakur, “A big data MapReduce framework for fault diagnosis in cloud-based manufacturing,” *Int. J. Prod. Res.*, vol. 54, no. 23, pp. 7060–7073, 2016.

[50] N. Japkowicz and S. Stephen, “The class imbalance problem: A systematic study,” *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, 2002.

[51] R. Longadge, S. S. Dongre, and L. Malik, “Class imbalance problem in data mining: review,” *Int. J. Comput. Sci. Netw.*, vol. 2, no. 1, pp. 83–87, 2013.

[52] A. Bahga and V. K. Madisetti, “Analyzing massive machine maintenance data in a computing cloud,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1831–1843, 2012.

[53] M. Devaney and B. Cheetham, “Case-Based Reasoning for Gas Turbine Diagnostics,” *18th Int. FLAIRS Conf. (FLAIRS-05)*, 2005.

[54] H. Timmerman, “SKF WindCon Condition Monitoring System for Wind Turbines,” in *New Zealand Wind Energy Conference*, 2009.

[55] P. Tamilselvan and P. Wang, “Failure diagnosis using deep belief learning based health state classification,” *Reliab. Eng. Syst. Saf.*, vol. 115, pp. 124–135, 2013.

[56] G. E. Hinton, “A Practical Guide to Training Restricted Boltzmann Machines,” in *Computer*, vol.

9, no. 3, 2012, pp. 599–619.

[57] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, “Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data,” *Mech. Syst. Signal Process.*, vol. 72–73, pp. 303–315, 2016.

[58] J. Schmidhuber, “Deep Learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.

[59] T. Banerjee, S. Das, J. Roychoudhury, and A. Abraham, “Implementation of a New Hybrid Methodology for Fault Signal Classification Using Short -Time Fourier Transform and Support Vector Machines,” *Soft Comput. Model. Ind. Environ. Appl. 5th Int. Work. (SOCO 2010)*, vol. 73, pp. 219–225, 2010.

[60] T. P. Banerjee and S. Das, “Multi-sensor data fusion using support vector machine for motor fault detection,” *Inf. Sci. (Ny).*, vol. 217, pp. 96–107, 2012.

[61] L. B. Jack and A. K. Nandi, “Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms,” *Mech. Syst. Signal Process.*, vol. 16, no. 2–3, pp. 373–390, 2002.

[62] M. Rychetsky, S. Ortmann, and M. Glesner, “Support vector approaches for engine knock detection,” *IJCNN’99. Int. Jt. Conf. Neural Networks. Proc. (Cat. No.99CH36339)*, vol. 2, no. 1, 1999.

[63] Y. Altintas, “In-process detection of tool breakages using time series monitoring of cutting forces,” *Int. J. Mach. Tools Manuf.*, vol. 28, no. 2, pp. 157–172, Jan. 1988.

[64] H. Wang, J. Zhoui, I. He, and J. Sha, “An uncertain information fusion method for fault diagnosis of complex system,” pp. 1505–1510, 2003.

[65] J. Xiong, Q. Zhang, G. Sun, X. Zhu, M. Liu, and Z. Li, “An Information Fusion Fault Diagnosis Method Based on Dimensionless Indicators with Static Discounting Factor and KNN,” *IEEE Sens. J.*, vol. 16, no. 7, pp. 2060–2069, 2016.

[66] A. P. Dempster, “A Generalization of Bayesian Inference,” *J. R. Stat. Soc.*, vol. 30, no. 2, pp. 205–247, 1968.

[67] M. Khakifirooz, C. F. Chien, and Y. J. Chen, “Bayesian inference for mining semiconductor manufacturing big data for yield enhancement and smart production to empower industry 4.0,” *Appl. Soft Comput. J.*, 2017.

[68] H. Lee, “Framework and development of fault detection classification using IoT device and cloud environment,” *J. Manuf. Syst.*, vol. 43, pp. 257–270, 2017.

[69] V. Gunes, S. Peter, T. Givargis, and F. Vahid, “A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems,” *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 12, pp. 120–132, Dec. 2014.

[70] R. (Raj) Rajkumar, I. Lee, L. Sha, and J. Stankovic, “Cyber-physical systems,” in *Proceedings of the 47th Design Automation Conference on - DAC ’10*, 2010, p. 731.

[71] M. Saez, F. Maturana, K. Barton, and D. Tilbury, “Modeling and Analysis of Cyber-Physical Manufacturing Systems for Anomaly Detection and Diagnosis,” 2018. .

[72] M. Saez, F. Maturana, K. Barton, and D. Tilbury, “Anomaly detection and productivity analysis for cyber-physical systems in manufacturing,” in *2017 13th IEEE Conference on Automation*

Science and Engineering (CASE), 2017, pp. 23–29.

[73] J. Wan, S. Tang, D. Li, S. Wang, C. Liu, H. Abbas, and A. V. Vasilakos, "A Manufacturing Big Data Solution for Active Preventive Maintenance," *IEEE Trans. Ind. Informatics*, vol. 13, no. 4, pp. 2039–2047, Aug. 2017.

[74] S. Munirathinam and B. Ramadoss, "Big data predictive analtyics for proactive semiconductor equipment maintenance," *Proc. - 2014 IEEE Int. Conf. Big Data, IEEE Big Data 2014*, pp. 893–902, 2015.

[75] J. Franklin, "Signalling and anti-proliferative effects mediated by gonadotrophin-releasing hormone receptors after expression in prostate cancer cells using recombinant adenovirus," *J. Endocrinol.*, vol. 176, no. 2, pp. 275–284, Feb. 2003.

[76] W. Ji and L. Wang, "Big data analytics based fault prediction for shop floor scheduling," *J. Manuf. Syst.*, vol. 43, pp. 187–194, 2017.

[77] B. F. Rolfe, Y. Frayman, G. L. Kelly, and S. Nahavandi, "Recognition of Lubrication Defects in Cold Forging Process with a Neural Network," in *Artificial Neural Networks in Finance and Manufacturing*, no. December 2015, IGI Global, 2006, pp. 262–275.

[78] M. Perzyk and A. W. Kochański, "Prediction of ductile cast iron quality by artificial neural networks," *J. Mater. Process. Technol.*, vol. 109, no. 3, pp. 305–307, Feb. 2001.

[79] E. Kilickap, A. Yardimeden, and Y. H. Çelik, "Mathematical Modelling and Optimization of Cutting Force, Tool Wear and Surface Roughness by Using Artificial Neural Network and Response Surface Methodology in Milling of Ti-6242S," *Appl. Sci.*, vol. 7, no. 10, p. 1064, 2017.

[80] C. Huang, X. Jia, and Z. Zhang, "A modified back propagation artificial neural network model based on genetic algorithm to predict the flow behavior of 5754 aluminum alloy," *Materials (Basel).*, vol. 11, no. 5, 2018.

[81] Á. Arnaiz-González, A. Fernández-Valdivielso, A. Bustillo, and L. N. López de Lacalle, "Using artificial neural networks for the prediction of dimensional error on inclined surfaces manufactured by ball-end milling," *Int. J. Adv. Manuf. Technol.*, vol. 83, no. 5–8, pp. 847–859, 2016.

[82] L. N. López de Lacalle, A. Lamikiz, M. A. Salgado, S. Herranz, and A. Rivero, "Process planning for reliable high-speed machining of moulds," *Int. J. Prod. Res.*, vol. 40, no. 12, pp. 2789–2809, 2002.

[83] L. N. L. De Lacalle, A. Lamikiz, J. A. Sánchez, and M. A. Salgado, "Effects of tool deflection in the high-speed milling of inclined surfaces," *Int. J. Adv. Manuf. Technol.*, vol. 24, no. 9–10, pp. 621–631, 2004.

[84] S. García, J. Luengo, and F. Herrera, "Tutorial on practical tips of the most influential data preprocessing algorithms in data mining," *Knowledge-Based Syst.*, vol. 98, pp. 1–29, 2015.

[85] H. Liu and R. Setiono, "A Probabilistic Approach to Feature Selection - A Filter Solution," *Proc. Int. Conf. Mach. Learn.*, pp. 319–327, 1996.

[86] R. Battiti, "Using Mutual Information for Selecting Features in Supervised Neural-Net Learning," *Ieee Trans. Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.

[87] K. Kira and L. Rendell, "A practical approach to feature selection," *Proceedings of the Ninth International Conference on Machine Learning*. pp. 249–256, 1994.

[88] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.

[89] P. Hart, “The condensed nearest neighbor rule (Corresp.),” *IEEE Trans. Inf. Theory*, vol. 14, no. 3, pp. 515–516, 1968.

[90] D. L. Wilson, “Asymptotic Properties of Nearest Neighbor Rules Using Edited Data,” *IEEE Trans. Syst. Man Cybern.*, vol. 2, no. 3, pp. 408–421, 1972.

[91] D. R. Wilson and T. R. Martinez, “Reduction Techniques for Instance-Based Learning Algorithms,” *Mach. Learn.*, vol. 38, no. 3, pp. 257–286, 2000.

[92] H. Brighton and C. Mellish, “Advances in Instance Selection for Instance-Based Learning Algorithms,” *Data Min. Knowl. Discov.*, vol. 6, no. 2, pp. 153–172, 2002.

[93] P. Stanula, A. Ziegenbein, and J. Metternich, “Machine learning algorithms in production: A guideline for efficient data source selection,” *Procedia CIRP*, vol. 78, pp. 261–266, 2018.

[94] M. H. U. Rehman, V. Chang, A. Batool, and T. Y. Wah, “Big data reduction framework for value creation in sustainable enterprises,” *Int. J. Inf. Manage.*, vol. 36, no. 6, pp. 917–928, 2016.

[95] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, “Fog Computing: Focusing on Mobile Users at the Edge,” pp. 1–11, 2015.

[96] X. Ma and R. J. Cripps, “Shape preserving data reduction for 3D surface points,” *CAD Comput. Aided Des.*, vol. 43, no. 8, pp. 902–909, 2011.

[97] I.-S. Jeong, H.-K. Kim, T.-H. Kim, D. H. Lee, K. J. Kim, and S.-H. Kang, “A Feature Selection Approach Based on Simulated Annealing for Detecting Various Denial of Service Attacks,” *Converg. Secur.*, vol. 2016, no. 1, pp. 1–18, 2016.

[98] S.-H. Kang and K. J. Kim, “A feature selection approach to find optimal feature subsets for the network intrusion detection system,” *Cluster Comput.*, vol. 19, no. 1, pp. 325–333, 2016.

[99] K. L. Du and M. N. S. Swamy, “Search and optimization by metaheuristics: Techniques and algorithms inspired by nature,” *Search Optim. by Metaheuristics Tech. Algorithms Inspired by Nat.*, pp. 1–434, 2016.

[100] A. Lalehpour, C. Berry, and A. Barari, “Adaptive data reduction with neighbourhood search approach in coordinate measurement of planar surfaces,” *J. Manuf. Syst.*, vol. 45, pp. 28–47, 2017.

[101] A. A. Ul Haq, K. Wang, and D. Djurdjanovic, “Feature Construction for Dense Inline Data in Semiconductor Manufacturing Processes,” *IFAC-PapersOnLine*, vol. 49, no. 28, pp. 274–279, 2016.

[102] M. Christ, A. W. Kempa-Liehr, and M. Feindt, “Distributed and parallel time series feature extraction for industrial big data applications,” Oct. 2016.

[103] Y. Benjamini and D. Yekutieli, “The control of the false discovery rate in multiple testing under dependency,” *Ann. Stat.*, vol. 29, no. 4, pp. 1165–1188, 2001.

[104] D. Dheeru and E. Karra Taniskidou, “UCI Machine Learning Repository.” University of California, Irvine, School of Information and Computer Sciences, 2017.

[105] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, “Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package),” *Neurocomputing*, vol. 307, pp.

72–77, 2018.

[106] W. Wang, J. Liu, G. Pitsilis, and X. Zhang, “Abstracting massive data for lightweight intrusion detection in computer networks,” *Inf. Sci. (Ny).*, vol. 433–434, pp. 1339–1351, 2018.

[107] J. Fan, F. Han, and H. Liu, “Challenges of Big Data analysis,” *Natl. Sci. Rev.*, vol. 1, no. 2, pp. 293–314, 2014.

[108] J. Campos, P. Sharma, U. G. Gabiria, E. Jantunen, and D. Baglee, “A Big Data Analytical Architecture for the Asset Management,” *Procedia CIRP*, vol. 64, pp. 369–374, 2017.

[109] K. Nikolaidis, J. Y. Goulermas, and Q. H. Wu, “A class boundary preserving algorithm for data condensation,” *Pattern Recognit.*, vol. 44, no. 3, pp. 704–715, 2011.

[110] S. Marsland, *Machine Learning: An Algorithmic Perspective*, 2nd ed. Boca Raton, FL: CRC Press, 2015.

[111] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, “An Overview of Machine Learning,” in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds. Palo Alto, CA: Tioga Publishing Company, 1983, pp. 3–23.

[112] K. Pearson, “X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” *Philos. Mag. Ser. 5*, vol. 50, no. 302, pp. 157–175, 1900.

[113] L. Vinet and A. Zhedanov, “Chi-Square Test,” in *Encyclopedia of Research Design*, vol. 44, no. 8, 2455 Teller Road, Thousand Oaks California 91320 United States: SAGE Publications, Inc., 2011, p. 085201.

[114] R. A. Fisher, “On the Interpretation of X2 from Contingency Tables, and the Caluclation of P,” *J. R. Stat. Soc.*, vol. 85, no. 1, pp. 87–94, 1922.

[115] R. A. Fisher, *Statistical Methods for Research Workers*, Fourteenth. Edinburgh: Oliver & Boyd, 1970.

[116] A. Agresti, “A Survey of Exact Inference for Contingency Tables,” *Stat. Sci.*, vol. 7, no. 1, pp. 131–153, 1992.

[117] F. J. J. Massey, “The Kolmogorov-Smirnov Test for Goodness of Fit,” *J. Am. Stat. Assoc.*, vol. 46, no. 253, pp. 68–78, 1951.

[118] M. G. Kendall, “A New Measure of Rank Correlation,” *Biometrika*, vol. 30, no. 1, pp. 81–93, 1938.

[119] M. Binshtok, R. I. Brafman, S. E. Shimony, A. Martin, and C. Boutillier, “Computing optimal subsets,” *Proc. 22nd Natl. Conf. Artif. Intell.*, pp. 1231–1236, 2007.

[120] M. Gendreau and J.-Y. Potvin, “Tabu Search,” in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, 2nd ed., E. K. Burke and G. Kendall, Eds. New York: Springer, 2014, pp. 243–264.

[121] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, Jan. 1986.

[122] F. Glover, “Tabu Search—Part I,” *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, Aug. 1989.

[123] F. Glover, “Tabu Search—Part II,” *ORSA J. Comput.*, vol. 2, no. 1, pp. 4–32, Feb. 1990.

[124] F. Glover, M. Laguna, and R. Marti, “Principles of tabu search,” *Approx. Algorithms Metaheuristics*, vol. 23, pp. 1–12, 2007.

[125] D. Kopeliovich, “Basic Principles of Heat Treatment,” *SubsTech: Substances & Technologies*, 2012. [Online]. Available: http://www.substech.com/dokuwiki/doku.php?id=basic_principles_of_heat_treatment. [Accessed: 24-Jul-2018].

[126] E. Aarts, J. Korst, and W. Michiels, “Simulated Annealing,” in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, 2nd ed., E. K. Burke and G. Kendall, Eds. New York: Springer, 2014, pp. 265–286.

[127] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, May 1983.

[128] D. Henderson, S. H. Jacobson, and A. W. Johnson, “The Theory and Practice of Simulated Annealing,” in *Handbook of Metaheuristics*, F. Glover and G. A. Kochenberger, Eds. Boston: Kluwer Academic Publishers, 2003, pp. 287–319.

[129] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. New York: Wiley, 1989.

[130] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, 1987.

[131] K. Sastry, D. E. Goldberg, and G. Kendall, “Genetic Algorithms,” in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, New York: Springer US, 2014, pp. 93–117.

[132] D. E. Goldberg and K. Deb, “A Comparative Analysis of Selection Schemes Used in Genetic Algorithms,” *Found. Genet. Algorithms*, vol. 1, pp. 69–93, 1991.

[133] D. E. Golberg, *Genetic Algorithms in Search Optimization & Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[134] G. Syswerda, “Uniform Crossover in Genetic Algorithms,” in *Proceedings of the Third International Conference on Genetic Algorithms: George Mason University*, San Mateo, CA: M. Kaufmann Publishers, 1989, pp. 2–9.

[135] J. M. Mendel, “Fuzzy logic systems for engineering: a tutorial,” *Proc. IEEE*, vol. 83, no. 3, pp. 345–377, 1995.

[136] L. a. Zadeh, “Fuzzy sets,” *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.

[137] E. H. Mamdani, “Application of fuzzy algorithms for control of simple dynamic plant,” *Proc. Inst. Electr. Eng.*, vol. 121, no. 12, p. 1585, 1974.

[138] Z. C. Yildiz, “A Short Fuzzy Logic Tutorial,” no. Figure 2, pp. 1–6, 2010.

[139] T. P. Hong and C. Y. Lee, “Induction of fuzzy rules and membership functions from training examples,” *Fuzzy Sets Syst.*, vol. 84, no. 1, pp. 33–47, 1996.

[140] D. Saletic, D. Velasevic, and N. Mastorakis, “Analysis of basic defuzzification techniques,” *6th WSES Int. Multiconference Circuits, Syst. Telecomunications Comput.*, no. January 2002, pp. 7–14, 2002.

[141] P. M. LaCasse, W. Otieno, and F. P. Maturana, “A hierarchical, fuzzy inference approach to data

filtration and feature prioritization in the connected manufacturing enterprise," *J. Big Data*, vol. 5, no. 1, p. 45, Dec. 2018.

[142] D. C. Howell, "Multiple Comparisons Among Treatment Means," in *Statistical Methods for Psychology*, 8th ed., Cengage Learning, 2012, pp. 384–387.

[143] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2012.

[144] "sklearn.svm.SVC," *Scikit-Learn Version 0.19.2 Onlne Documentation*. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC. [Accessed: 21-Aug-2018].

[145] J. Warner, J. Sexauer, scikit-fuzzy, twmeggs, A. M. S., A. Unnikrishnan, G. Castelão, F. Batista, T. G. Badger, and H. Mishra, "JDWarner/scikit-fuzzy: Scikit-Fuzzy 0.3.1," Oct. 2017.

[146] S. A. Flaum, "Pareto's Principle," *Parmaceutical Executive*, vol. 27, no. 2, Duluth, MN, pp. 54–56, Feb-2007.

[147] H. B. Harvey and S. T. Sotardi, "The Pareto Principle," *J. Am. Coll. Radiol.*, vol. 15, no. 6, p. 931, 2018.

[148] P. Gupta, "The Pareto Principle," *Printed Circuit Fabrication*, vol. 24, no. 1, San Francisco, pp. 62–63, Jan-2001.

[149] L. S. Goodenday, K. J. Cios, M. Ogiela, R. Tadeusiewicz, and L. A. Kurgan, "Knowledge discovery approach to automated cardiac SPECT diagnosis," *Artif. Intell. Med.*, vol. 23, no. 2, pp. 149–169, 2002.

[150] "SMT Manufacturing," *actsource: Experts in Electronics Manufacturing*, 2010. [Online]. Available: http://www.act-source.com/pcb-assembly/smt-manufacturing. [Accessed: 03-Jul-2018].

[151] W. J. Trybula and M. Trybula, "Surface Mount Technology," *Encyclopedia of RF and Microwave Engineering*. John Wiley & Sons, Inc., pp. 2058–5067, 2005.

[152] G. Sharman, "Nobody Calls Me General Anymore!," *McKinsey Q.*, p. 106, 1996.

[153] P. Zarrow, "Reflow Soldering of Through-hole Components," *J. Surf. Mt. Technol.*, no. 12, pp. 13–16, 1999.

[154] B. Wettermann, "Top 5 BGA challenges to overcome," *SMT Surf. Mt. Technol. Mag.*, vol. 32, no. 9, pp. 25–29, 2017.

[155] Z. Zhao, C. Chen, C. Y. Park, Y. Wang, L. Liu, G. Zou, J. Cai, and Q. Wang, "Effects of package warpage on head-in-pillow defect," *Mater. Trans.*, vol. 56, no. 7, pp. 1037–1042, 2015.

[156] M. Scalzo, "Addressing the Challenge of Head-In-Pillow Defects in Electronics Assembly," in *APEX EXPO Technical Conference*, 2009.

[157] A. A. Primavera, "Influence of PCB Parameters on Chip Scale Package Assembly and Reliability," 1999. [Online]. Available: https://pdfs.semanticscholar.org/7e83/37740819374e0b3dd72213a1448c03d5358c.pdf. [Accessed: 15-Nov-2018].

[158] T. N. Tsai and M. Liukkonen, "Robust parameter design for the micro-BGA stencil printing

process using a fuzzy logic-based Taguchi method," *Appl. Soft Comput. J.*, vol. 48, pp. 124–136, 2016.

[159] S. Cheng, C. M. Huang, and M. Pecht, "A review of lead-free solders for electronics applications," *Microelectron. Reliab.*, vol. 75, pp. 77–95, 2017.

[160] R. Pandher, N. Jodhan, R. Raut, and M. Liberatore, "Head-in-pillow defect - Role of the solder ball alloy," *2010 12th Electron. Packag. Technol. Conf. EPTC 2010*, pp. 151–156, 2010.

[161] S. Harter, T. Klinger, J. Franke, and D. Beer, "Comprehensive correlation of inline inspection data for the evaluation of defects in heterogeneous electronic assemblies," *2016 Pan Pacific Microelectron. Symp. Pan Pacific 2016*, 2016.

[162] D. Bernard and E. Krastev, "Modern 2D X-ray Tackles BGA Defects," *SMT Surf. Mt. Technol. Mag.*, vol. 22, no. 7, pp. 22–24, 2008.

[163] S. Peng and H. Do Nam, "Void defect detection in ball grid array X-ray images using a new blob filter," *J. Zhejiang Univ. Sci. C*, vol. 13, no. 11, pp. 840–849, 2012.

[164] Y. Wang, M. Wang, and Z. Zhang, "Optik Microfocus X-ray printed circuit board inspection system," *Opt. - Int. J. Light Electron Opt.*, vol. 125, no. 17, pp. 4929–4931, 2014.

[165] A. Castellanos, Z. Feng, D. Geiger, and M. Kurwa, "Head-in-Pillow X-ray Inspection," *SMT Surf. Mt. Technol. Mag.*, vol. 29, no. 5, pp. 16–29, May 2014.

[166] T. SUMIMOTO, T. MARUYAMA, Y. AZUMA, S. GOTO, M. MONDOU, N. FURUKAWA, and S. OKADA, "Detection of Defects of BGA by Tomography Imaging," *J. Syst. Cybern. INFORMATICS*, vol. 3, no. 4, pp. 10–14, 2005.

[167] T. W. Hui and G. K. H. Pang, "Solder paste inspection using region-based defect detection," *Int. J. Adv. Manuf. Technol.*, vol. 42, no. 7–8, pp. 725–734, 2009.

[168] C. H. Kuo, F. C. Yang, J. J. Wing, and C. K. Yang, "Construction of 3D solder paste surfaces using multi-projection images," *Int. J. Adv. Manuf. Technol.*, vol. 31, no. 5–6, pp. 509–519, 2006.

[169] M. H. Chu and G. K. H. Pang, *Solder paste inspection by special led lighting for SMT manufacturing of printed circuit boards*, vol. 8, no. PART 1. IFAC, 2007.

[170] X. P. Li, J. M. Xia, M. B. Zhou, X. Ma, and X. P. Zhang, "Solder volume effects on the microstructure evolution and shear fracture behavior of ball grid array structure Sn-3.0Ag-0.5Cu solder interconnects," *J. Electron. Mater.*, vol. 40, no. 12, pp. 2425–2435, 2011.

[171] D. Yang, J. Cai, Q. Wang, J. Li, Y. Hu, and L. Li, "IMC growth and shear strength of Sn–Ag–Cu/Co–P ball grid array solder joints under thermal cycling," *J. Mater. Sci. Mater. Electron.*, vol. 26, no. 2, pp. 962–969, 2014.

[172] Y. C. Chiou, C. S. Lin, and B. C. Chiou, "The feature extraction and analysis of flaw detection and classification in BGA gold-plating areas," *Expert Syst. Appl.*, vol. 35, no. 4, pp. 1771–1779, 2008.

[173] H. Lee, S. Baeg, N. Hua, and S. Wen, "Temporal and frequency characteristic analysis of margin-related failures caused by an intermittent nano-scale fracture of the solder ball in a BGA package device," *Microelectron. Reliab.*, vol. 69, pp. 88–99, Feb. 2017.

[174] J. Lee and H. W. Park, "Prediction enhancement of the J-lead interconnection reliability of land grid array sockets," *J. Mech. Sci. Technol.*, vol. 29, no. 5, pp. 2187–2193, 2015.

[175] C. Chen, J. Cai, Q. Wang, Y. Wang, G. Zou, Z. Zhao, and C. Y. Park, "Characterization of after-

reflow misalignment on Head-in-Pillow defect in BGA assembly," in *2014 15th International Conference on Electronic Packaging Technology*, 2014, pp. 1177–1180.

[176] T. Yang and T. N. Tsai, "A neurofuzzy-based quality-control system for fine pitch stencil printing process in surface mount assembly," *J. Intell. Manuf.*, vol. 15, no. 5, pp. 711–721, 2004.

CURRICULUM VITAE

Phillip M. LaCasse

Place of birth: Neenah, WI

Education

B.S., United States Military Academy, May 2000
Major: Mathematics

M.S., Industrial Engineering, University of Wisconsin – Madison, May 2010

Ph.D., Engineering, University of Wisconsin – Milwaukee, May 2019

Dissertation Title: A Hierarchical, Fuzzy Inference Approach to Data Reduction and Feature Prioritization in the Connected Manufacturing Enterprise

Scholastic Award: Nominee for the 2019 UWM College of Engineering and Applied Sciences Distinguished Graduate Student Award, May 2019.

Peer Reviewed Publications and Papers in Review

(1) LaCasse, P., Otieno, W., Maturana, F., Operationalization of Defect Prediction Case Study in a Holonic Manufacturing System, *9th International Conference on industrial Applications of Holonic and Multi-Agent Systems (HoloMAS)*, 2019, Linz, Austria, under 1st review.
(2) LaCasse, P., Otieno, W., Maturana, F., Predicting Contact-Without-Connection Defects on Printed Circuit Boards Employing Ball Grid Array Package Types: A Data Analytics Case Study in the Smart Manufacturing Environment, *J. of Data Sci. and Analytics*, under 1st review.
(3) P. LaCasse, W. Otieno, and F. Maturana, "A Survey of Feature Set Reduction Approaches for Predictive Analytics Models in the Connected Manufacturing Enterprise," *Appl. Sci.*, vol. 9, no. 5, p. 843, 2019
(4) Otieno, W., Garantiva, J., LaCasse, P., Optimal One-Dimensional Free-Replacement Warranty Period for AGM Batteries, *IEEE-Explore, Annual Reliability and Maintainability Symposium Proceedings*, January 2019
(5) P. M. LaCasse, W. Otieno, and F. P. Maturana, "A hierarchical, fuzzy inference approach to data filtration and feature prioritization in the connected manufacturing enterprise," *J. Big Data*, vol. 5, no. 1, p. 45, Dec. 2018

Teaching Experience

MA205: Integral Calculus and Introduction to Differential Equations, United States Military Academy

- Fall 2010
- Fall 2011 (Assistant Course Director)
- Spring 2012 (Course Director)

- Fall 2012 (Course Director)

MA206: Probability and Statistics, United States Military Academy

- Spring 2011
- Summer Term Academic Program (STAP) 2011
- Spring 2013

IME 575: Design of Experiments, University of Wisconsin – Milwaukee

- Spring 2018
- Spring 2019