APPLICATION OF MACHINE LEARNING IN MULTISTAGE QUALITY CONTROL IN AUTOMOTIVE INDUSTRY

A MULTI DISCIPLINARY DESIGN REPORT

Submitted by

ABHIMANYU WADHWA [RA1711003020595]

NIKHIL CHOPRA [RA1711003020591]

RITWICK BHADURI [RA1711003020596]

Under the guidance of

Ms. Vidhyavani,

(Assistant Professor, Department of Computer Science & Engineering)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

RAMAPURAM CAMPUS - CHENNAI 600 089, MAY 2020

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this [15CS303M] A MULTI DISCIPLINARY DESIGN report titled **"Application of Machine Learning in Multistage Quality Control in Automotive Industry",** is the bonafide work of **ABHIMANYU WADHWA [RA1711003020595], NIKHIL CHOPRA [RA1711003020591], RITWICK BHADURI [RA1711003020596],** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form any other project report or dissertation.

SIGNATURE	**SIGNATURE**
Ms. Vidhyavani,	**Dr. N.KANNAN Ph.D.**
Assistant Professor,	**Head of the Department,**
Department of Computer Science & Engineering,	Department of Computer Science & Engineering,
SRMIST, Chennai.	SRMIST, Chennai.

Submitted for End Semester Practical Examination held on__________________at SRM Institute of Science and Technology, Ramapuram, Chennai- 600 089

EXAMINER-1	**EXAMINER-2**

DECLARATION

We hereby declare that the entire work contained in this report entitled **"APPLICATION OF MACHINE LEARNING IN MULTISTAGE QUALITY CONTROL IN AUTOMOTIVE INDUSTRY"** is an authentic record of our own work as requirements of [15CS303M] A MULTI DISCIPLINARY DESIGN. It has been carried out by us at SRM Institute of Science and Technology, Ramapuram Campus, Chennai, under the efficient guidance of **Ms. Vidyavani, Assistant Professor, Department of Computer Science and Engineering.**

NAME	REGISTRATION NO	SIGNATURE
ABHIMANYU W.	RA1711003020595	
NIKHIL C.	RA1711003020591	
RITWICK B.	RA1711003020596	

ABSTRACT

The manufacturing process is a very crucial process in the automotive industry. The success of the automobiles produced largely relies on customer satisfaction and how the product is received in the market. The process of Quality Control is used after the production of components in order to ensure that there are no manufacturing defects.

In some factories, this process is done manually and the rejected components are discarder. This is a tedious and time-consuming process. The recent advances in information technologies and consequently the increased volume of data that has become readily available provide an excellent opportunity for the development of automated defect detection approaches that are capable of extracting the implicit complex relationships in these multivariate data-rich environments. In this paper, several machine learning classifiers were trained and evaluated on varied metrics to predict dimensional defects in a real automotive multistage assembly line. The line encompasses two automated inspection stages with several human-operated assembly and pre-alignment stages in between. One possible barrier to the success of such a predictive solution in the long term is the possibility of drastic changes in the underlying distributions of the dimensional characteristics of cars. This can happen for instance due to a change in the materials, suppliers or the replacement of parts in the stations, our objective is to overcome this limitation. A possible solution in the occurrence of this case during production would be through online monitoring and/or training of the models using, for instance, an architecture similar to the one showcased in based on the IDARTS framework. Hence by applying these various techniques of Machine Learning into the manufacturing process, it can be simplified and we can ensure the better utilization of available resources. Large manufacturing companies could ensure that the raw material is not being wasted and could ensure the best possible yield since the number of defective parts produced would reduce by a considerable amount.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The manufacturing process is a very crucial process in the automotive industry. The success of the automobiles produced largely relies on customer satisfaction and how the product is received in the market. The process of Quality Control is used after the production of components in order to ensure that there are no manufacturing defects. In some factories, this process is done manually and the rejected components are discarder. This is a tedious and time-consuming process.

In order to increase the speed and efficiency of this quality control process, some techniques of the computer science industry must be implemented within these processes. This will not only ensure better success rate, but also decrease the time consumed in the process of quality control. Thus, by using some machine learning techniques, the aim of this project is to increase the efficiency of the quality control department of a factory.

The recent advances in information technologies and consequently the increased volume of data that has become readily available provide an excellent opportunity for the development of automated defect detection approaches that are capable of extracting the implicit complex relationships in these multivariate data-rich environments.

In this paper, several machine learning classifiers were trained and evaluated on varied metrics to predict dimensional defects in a real automotive multistage assembly line. The line encompasses two automated inspection stages with several human-operated assembly and pre-alignment stages in between.

Drawbacks of the traditional system:

To predict deviations at the end of the line regardless of these interventions, indicating that some of these feature interactions are considerably hard to detect.

The correct assessment of the corrective actions that need to be carried out during the assembly operations (i.e. offsetting the jig) can be improved.

1.1.2 PROBLEM STATEMENT

The existing system does not make use of all the available resources and hence a large amount of resources are wasted, which results in a not so profitable yield for the manufacturing company. By introducing the concepts of Machine Learning in fault detection, the defects in the components may be identified easily and thus a better yield would be produced. This will increase the overall efficiency of the manufacturing process and hence benefit the manufacturing company largely.

To predict deviations at the end of the line regardless of these interventions, indicating that some of these feature interactions are considerably hard to detect. The correct assessment of the corrective actions that need to be carried out during the assembly operations (i.e. offsetting the jig) can be improved.

1.1.3 OBJECTIVE

One possible barrier to the success of such a predictive solution in the long term is the possibility of drastic changes in the underlying distributions of the dimensional characteristics of cars. This can happen for instance due to a change in the materials, suppliers or the replacement of parts in the stations, our objective is to overcome this limitation.

A possible solution in the occurrence of this case during production would be through online monitoring and/or training of the models using, for instance, an architecture similar to the one showcased in based on the IDARTS framework.

CHAPTER 2

LITERATURE SURVEY

2.1 Survey 1

AUTOMATED SURFACE DEFORMATIONS DETECTION AND MARKING ON AUTOMOTIVE BODY PANELS (2010) (IEEE)

Authors:

1) **Valentin Borsu :** Student, Computer Science
2) **Arjun Yogeswaran** : Student, Computer Science
3) **Pierre Payeur :** Student, Computer Science

Concept:

This paper proposes an integrated solution for automated surface deformations detection and marking on automotive body panels in the context of quality control in industrial manufacturing. Starting from a 3D image of the surface of the panel, deformations are extracted and classified automatically.

Algorithm:

Integrity Verification Algorithm

The system requires executing the application and rolling back to re-execute with different input providing no real-time protection and the algorithm to detect divergent output in network.

Limitations:

The computation of data is a slow process.

2.2 Survey 2

MACHINE LEARNING TECHNIQUES FOR QUALITY CONTROL IN HIGH CONFORMANCE MANUFACTURING ENVIRONMENT (2017) (IEEE)

Authors:

1) **Carlos A Escobar**: Student, Mechanical Engineering
2) **Ruben Morales-Menendez:** Student, Mechanical Engineering

Concept:

This article presents the learning process and pattern recognition strategy for a knowledge-based intelligent supervisory system, in which the main goal is the detection of rare quality events. Defect detection is formulated as a binary classification problem. The l1-regularized logistic regression is used as the learning algorithm for the classification task and to select the features that contain the most relevant information about the quality of the process. The proposed strategy is supported by the novelty of a hybrid feature elimination algorithm and optimal classification threshold search algorithm.

Algorithm:

Watermarking algorithm- The developed algorithm is a blind watermarking technique that meets the requirements of invisibility and robustness. Watermarking is performed by embedding a watermark in the middle-frequency coefficient block of three DWT levels.

Limitations:

The whole process is depend upon watermark process, if the sensitive data is not catch by watermark process there is a possibility of data leakage.

2.3 Survey 3

MACHINE LEARNING IN AUTOMOTIVE INDUSTRY (2018) (IEEE)

Authors:

1) **Baozhen Yao:** Student, Mechanical Engineering
2) **Tao Feng:** Student, Mechanical Engineering

Concept:

The approach integrated Support Vector Machine (SVM) into the type-2 fuzzy learning systems to generate optimal rules for the system nonlinear character and complicated formation state. Reinforcement learning was combined with type-2 fuzzy systems to deal with acoustic communication conditions during the formation process. The proposed methods' high performance was tested by simulations and experiments. In the paper ''Algorithmic design and application of feedback control for coiling temperature in hot strip mill,''8 the authors introduced a laminar cooling control system and designed a feedback control algorithm based on proportional–integral controller and Smith predictor, using the online adaptive algorithm to optimize parameters of proportional–integral controller.

Algorithm:

Algorithm provide the hash function at the time of data upload with the reference to the third party agent and the to the actual cloud, this security feature is added to enhance the security of sensitive data transfer over the internet.

Limitations:

Generation of more number of fake records dynamically, according to the agent's request.

2.4 Survey 4

FAULT DIAGNOSIS OF MULTISTAGE MANUFACTURING PROCESS BY USING STATE SPACE APPROACH (2002) (IEEE)

Authors:

1) **Yu Ding:** Student, Industrial Engineering
2) **Dariusz Ceglarek:** Student, Industrial Engineering
3) **Jianjun Shi:** Student, Industrial Engineering

Concept:

This paper presents a methodology for diagnostics of fixture failures in multistage manufacturing processes (MMP). The diagnostic methodology is based on the state-space model of the MMP process, which includes part fixturing layout geometry and sensor location. The state space model of the MMP characterizes the propagation of fixture fault variation along the production stream, and is used to generate a set of predetermined fault variation patterns. Fixture faults are then isolated by using mapping procedure that combines the Principal Component Analysis (PCA) with pattern recognition approach.

Algorithm:

Minimal Generalization algorithm has been implemented in the proposed model of this journal.

Limitations:

Certain data cannot admit watermarks then it is possible to assess the likelihood that an agent is responsible for any violation.

2.5 Survey 5

RECENT ADVANCES AND TRENDS IN PREDICTIVE MANUFACTURING SYSTEMS IN BIG DATA ENVIRONMENT (2013) (IEEE)

Authors:

1) **Jay Lee:** Student, Mechanical Engineering
2) **Edzel Lapira:** Student, Mechanical Engineering
3) **Behrad Bagheri:** Student, Mechanical Engineering
4) **Hung-an Kao:** Student, Mechanical Engineering

Concept:

The globalization of the world's economies is a major challenge to local industry and it is pushing the manufacturing sector to its next transformation – predictive manufacturing. In order to become more competitive, manufacturers need to embrace emerging technologies, such as advanced analytics and cyber-physical system-based approaches, to improve their efficiency and productivity. With an aggressive push towards "Internet of Things", data has become more accessible and ubiquitous, contributing to the big data environment. This phenomenon necessitates the right approach and tools to convert data into useful, actionable information.

Algorithm:

SVM Algorithm is used.

Limitations:

This algorithm is not suitable for large data sets.

CHAPTER 3

SPECIFICATION

3.1 Introduction

Modern manufacturing and assembly facilities are becoming data-rich environments. For example, atypical automotive assembly line contains numerous data sources that provide the real-time status of programmable logic controllers (PLC) into a central system. Industries are investing in a variety of sensor technologies to increase operations' visibility (Kocet al. 2005, Brusey and McFarlane 2009). The availability of vast amounts of data and increased information visibility offer an unprecedented opportunity to model and simulate the real-time performance of an assembly line, including the distribution of work in process (WIP) and delays at various stations, throughput, yield, etc. Simulation techniques are widely used in the domain of manufacturing systems due to its capability to deal with variability and uncertainty, and the use of graphical user interfaces to facilitate communication with, and comprehension by, plant floor managers and strategic planners (Young et al. 1988). Discrete event simulation (DES) and system dynamics (SD) are two well-established simulation approaches in operational research (Reitman1974, Jahangirian et al. 2010). Discrete event simulation(DES) specifically focuses on the individual entities and their activities in the network of queues. System dynamics (SD) operate at a much more aggregated level by concentrating on system structures and the rates of change of populations of entities(Towill 1991). Hence, DES models have traditionally been used to answer specific questions at the operational or tactical level. System dynamics (SD) are used at a higher, more strategic level for deeper insights into the interrelations between the different parts of a complex manufacturing system.

In this investigation, we developed a novel continuous flow approach to model a real-world multistage assembly line system. Here, a production system is tracked using a set of state variables such as WIP levels, delays, production rates, and coupled temporal dynamics. The proposed continuous flow approach models the system as a series of buffer stocks and product flows, in which the state changes are continuous. The entities are viewed as a continuous fluid, flowing through a system of tanks connected by pipes. The rates of flow are controlled by valves, and the time spent in each tank is dependent on the rates of influx and outflux. Analogously, the part movement is treated as fluid flow, buffer stocks are water tanks, the conveyor belt is a water pipe, and manufacturing stations are the valves that control the rates of flow.

A set of ordinary differential equations (ODE) is derived to model the system of buffer stocks and production flows between the interacting machines. Our experimental results demonstrated that variations of Key Performance Indicators (KPI), such as processing velocity, throughput rate (which is usually determined by averaging the processing velocities over a time interval), WIP, and throughput loss, were effectively captured in the proposed continuous flow models. This paper is organized as follows: Section 2presents the background of DES and SD modelling in manufacturing systems. Section 3 details the proposed research methodology of the continuous flow model. Section 4 contains the model implementation. Section5 discusses the results of model performances. Section6 concludes this presented investigation.

3.2 Overall Description

Performance estimation involves harnessing information in the form of tractable models for the dynamic behaviours of manufacturing systems. Such information is usually buried in the measured data streams, and features need to be extracted based on the model to estimate performance. A typical automotive assembly-line contains hundreds of PLCs to record events such as the start and end of an operation, exceptions and errors, which can track even micro-motions in an assembly operation. The magnitude of data streams and the complex dynamics of the underlying manufacturing system poses significant challenges in deriving adequate models using these data streams for the real-time performance estimation.

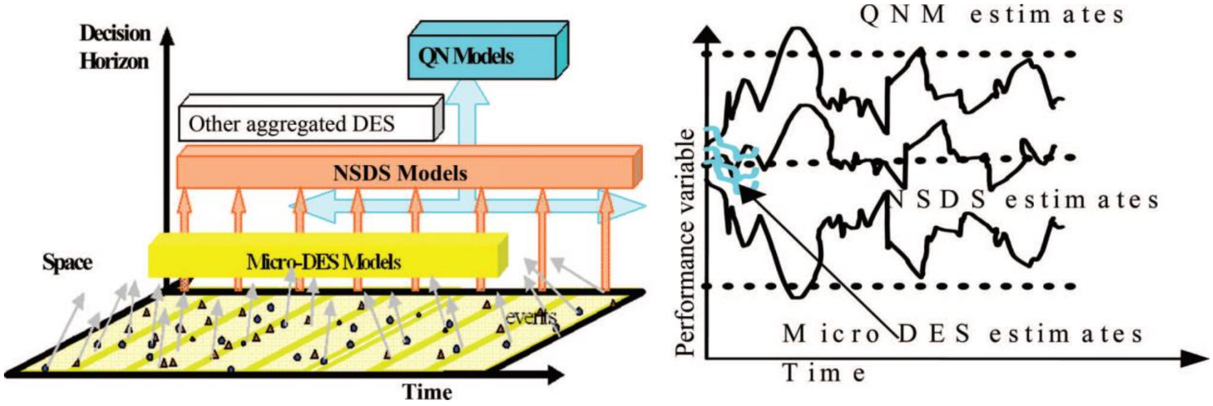


Figure 3.1 DES estimation

As shown in Figure 1, there are three broad classes of models for real-time performance estimation and prediction using online data, i.e. queueing network models (QNM), detailed DES models and nonlinear system dynamics simulation (NSDS) models. Each class of models is suitable for a particular decision horizon. In QNM models, network systems contain a finite number of queues. A queue is a waiting line of customers. The system performance will be described by the number or types of customers in the waiting lines (Almeida et al. 1998). In a multistage manufacturing assembly line, customers are work orders (jobs to be completed). Orders must visit one or more workstations before they are completed and shipped. Each workstation is viewed as a simple queueing system. The entire set of workstations is viewed as a queueing network (Jackman and Johnson 1993). Queueing network models (QNM) estimate performance by parameterizing the model through aggregating temporal data streams into a single distribution under the assumptions of a steady-state. As shown in figure 1, these assumptions are reasonable for longterm(days through weeks) decision-horizons. Queueing network models (QNM) are widely used to provide long-term distributions and performance bounds.
Discrete event simulation (DES) models the systems as networks of queues and activities, where state changes occur at discrete points of time (Weigert and Henrich 2009). It captures the detailed changes in the status of multistage manufacturing systems resulting from, for example, the movement of individual workpieces, the releasing, queuing, and processing of work and capacity. Discrete event simulation (DES) models are generally more appropriate when the operational details need to be modeled, especially when individual items need to be tracked. Such DES models are parsimonious when the system evolves through a small and finite set of states and external stimuli are given at a particular time or at a finite set of time. However, it will be time-consuming and computationally expensive to reduce the variance of simulation results on a large scale and highly variable system. Because each entity possesses characteristics that determine its activities using probability distributions, and each simulation run or iteration only represents the one realization of the system. Therefore, DES models require many iterations to generate large ensembles of datasets to extract statistically significant metrics including the meantime to repair (MTTR), mean time between failure (MTBF), processing time averages, and variances (Tako and Robinson 2009,2010). Besides, DES models tend to construct a more lifelike representation of real-world systems, which consequently results in more detailed and complex models.
Detailed DES models, also called as micro-DES models in the industry, use online data to estimate near-term (typically, minutes) performance look ahead (Wu and Wysk 1989). The aggregation of events is necessary to yield performance estimates over the medium-term decision horizon (typically, hours)(Armbruster et al. 2004b). As shown in Figure 1(b), multiple realizations of micro-DES models (light cyan lines) are aggregated to estimate the system behaviors in the medium-term horizons.
System dynamics (SD) models consider the system of buffer stocks and production flow instead of the individual work parts. In the continuous flow models, state changes continuously at small segments of time. It may be noted that such continuous flow models operate at a much more aggregated level by concentrating on system structures and dynamic tendencies involved in the change of populations of entities. The specification of the model structure consists of the explicit representation of the feedbacks and causal relationships that generate

the dynamic behaviors of the system. The continuous flow models are more suitable whenever the stimulus is a continuous function of time, and the response is time-varying over a continuum of states. The potential limitations of continuous flow models include: (1) The materials are represented as a continuous quantity, and individual parts cannot be tracked through the system. (2) The continuous flow model operates at a more aggregated level focusing on the SD and material flow rate, which is better for answering questions at the macro level instead of the micro level. (3) Because of continuous state variations and dynamic tendencies involved, the continuous flow model is more sensitive to the external stimuli (e.g. demand variations).

However, a dynamic system view will aid in deriving closed-loop distributed control, feedback structure and coordination policies, akin to making fast and informed decisions, such that the system can respond expeditiously and effectively to external stimuli (e.g. demand dynamics, arrival processes,etc.) (Wiendahl and Scheffczyk 1999, Kumara et al. 2003, Papakostas et al. 2009). Accurate and real-time performance estimation and prediction based on the models derived using these data streams are the basis of these closed-loop control and coordination strategies (AlDurgham and Barghash 2008, Shin et al.2010). However, DES models are traditionally used in the 'what-if' experimentation, in which the effects of various alternatives are investigated (Scholz-Reiteret al. 2005, Brailsford 2008). Despite the differences between DES and continuous flow models, both simulation approaches are aimed at understanding how systems behave over time and estimating system performance under different external conditions.

While QNM and micro-DES are standard practices in the research community, NSDS models have been marginal in the manufacturing systems domain(Armbruster et al. 2001, 2004a, 2004c, Westman, and Hanson 1999). Many previous approaches, such as asQNM, and micro-DES, allude to the potential of time aggregated models for performance estimation. A gap remains in synthesizing these models for real-world systems using online data. It is noteworthy that whereas most of the behaviors emerging due to nonlinear dynamics in a manufacturing system have little role to play in QN (temporal dynamics is assumed to be a stationary, usually second-order process) and micro-DES (linearity assumptions can hold over such short time-scales), the consideration of NSDS models becomes critical. The use of a continuous flow modeling approach was introduced by Newell (1965) to approximately solve queuing problems. These models consider the length of a queue and WIP as a continuous variable but model machines as individual discrete quantities. Furthermore, Armbruster et al. (2004a) developed a heuristic model with differential equations for a truly continuous description of the production process using a state equation that relates the speed of the product moving through the factory to the amount of product in the factory (i.e. relating throughput and WIP). Table 1 summarises the comparisons of the three aforementioned classes of models in the field of manufacturing systems. This present investigation aims to model the dynamics of multi-stage manufacturing systems using continuous flow modeling approaches. We derived a large system of ODEs for modeling the interactions between machines through flux conservation. The variations of buffer stocks are dependent on the differences in throughput rate between adjacent machines. This model will be parameterized using real-world data from plant floor systems (PFS).

3.2 Project Features

The investigated assembly line operation is a classical example of a 'push' system. It may be noted that although the segment of 18 machines runs according to a 'push' system, the production system is not designed as a pure push or pure pull system. For example, the stock point of raw materials (i.e. in the beginning of this segment) may run according to a pull system. That is, the order for refilling the stock point is released when the level of a particular item reaches its reorder point. In addition, the continuous flow modelling approach is adaptable for both make-to-order (i.e. push system) and make-to-stock (i.e. pull system) scenarios. Specifically, each manufacturing station has both influx and outflux production flows resulting from the interaction of production machine networks. The statuses of machines considered include processing, blocked, starved/idle and failure/down. The machines are allowed to process a job as long as it is not in the following conditions:

starved (i.e. its input buffer is empty), blocked (i.e. the output buffer reaches the limit) or down (i.e. needs repair or under repair).

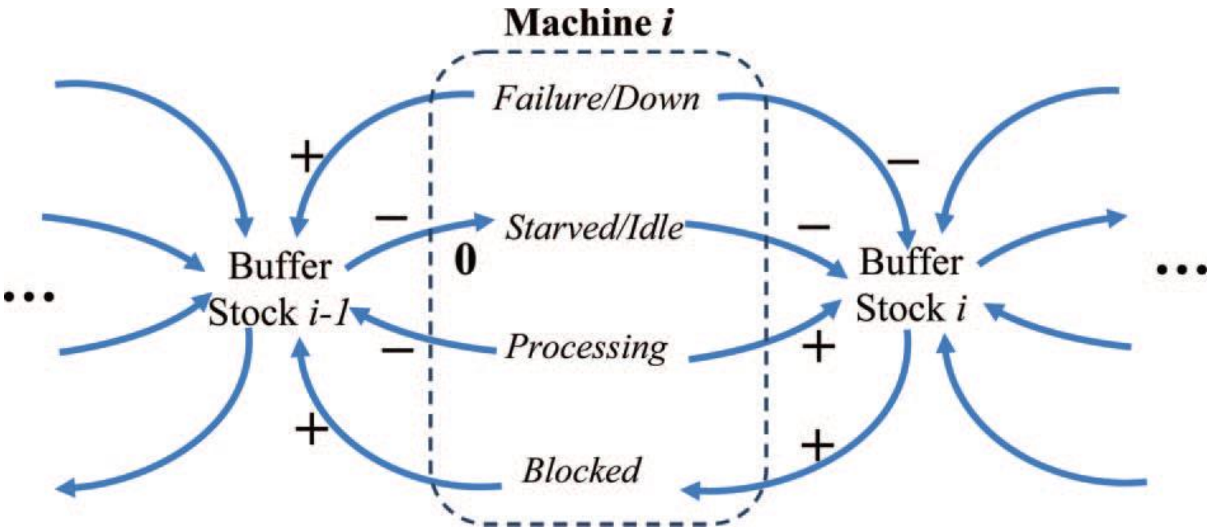


Figure 3.2 Insights to variables interact

The proposed continuous flow modelling approach consists of both qualitative and quantitative aspects, which is aimed at enhancing the understanding of system structures and inter-relationships between relevant subsystems. The qualitative analysis identifies the elements, state variables in the system that have an influence on the modelling and simulation. As shown in Figure 2, the identified elements are connected by arrows. The 'þ' and '7' signs denote the direction of the influence, but do not show the magnitude of the influence. For example, the processing in machine i reduces the upstream buffer stock, shown by a '7', and increases the downstream buffer stock, shown by a 'þ'. When the downstream buffer stock increases to its limit, this machine will be blocked and consequently result in the increase of upstream buffer stock. Similarly, the decrease of upstream buffer stock to '0' will cause the machine to be starved or idle, and the downstream buffer stock will decrease if the next machine is operating normally. The machine failure/ down will increase the upstream buffer stock, shown by a 'þ' and decrease the downstream buffer stock, shown by a '7'. In this way, the informative influence diagrams will be constructed to represent the problem being investigated and provide insights into how the variables interact (Brailsford 2008).

The influence diagram shows how variables change over time, allowing their behaviours to be modelled and simulated in the continuous way. The continuous flow modelling approach treats increasing and decreasing rate of materials (i.e. working parts through the assembly line). The machine statuses (i.e. breakdown, starved/idle, processing, blocked) will determine the variations of the continuous flow. The continuum assumption is rationalised for the cases of mass manufacturing, e.g. an automotive assembly line where Table 1. Summary of the relevant alternative manufacturing systems modelling approaches. Approach Remarks Stochastic system and classical queuing network model (QNM) . Capture simple steady-state behaviours, ignore emerging behaviours in nonlinear and other complex systems Discrete event simulation (DES) . Detailed, captures nonlinearities, slow and unwieldy for estimating performance and synthesising control . Decision making in the operational and tactical level Nonlinear system dynamics simulation (NSDS) model (e.g. continuous flow model) . Fast, yet mostly qualitative . Aggregated, captures nonlinearities, performance predictor and controller not directly derivable . Predictors and controllers derivable for simple systems, nonlinearity ignored and/or grossly simplified . Policy implementing in the strategic level Figure 2. Influence diagram of a manufacturing station. 404 H. Yang et al. Downloaded by [University of South Florida] at 13:39 04 February 2015 the inter-release time scales of working parts are much smaller than those for other events. This approach essentially models the movement of materials in a manufacturing system as a fluid flow and offers an advantage to estimate the aggregated dynamic patterns of the multistage manufacturing system.

3.3 Operating Environment

As shown in Figure 3, machine/station operation in a multistage assembly line includes material flow and information flow. Material flow is following the assembly sequence of work pieces. Information flow indicates the signal flows about requesting the supply from upstream machines, authorising a station to start processing work pieces including machine starved, down, blocked information, etc. The material flow into the buffer stock i is denoted ui(t) and the flow out is denoted by uiþ1(t). These flow are regulated by the upstream and downstream machines to maintain an optimised buffer level, yi(t). Since the level of buffer stock can never be negative and it is limited by some storage capacity Li, the variable yi(t) is constrained by 0 _ yi(t) _ Li for all time instants t. For a simple N-stage manufacturing system, the change in the level of the buffer yi at the downstream of ith operation is given as follows: y_1ðtÞ ¼ u1ðtÞ _ u2ðtÞ . . . y_iðtÞ ¼ uiðtÞ _ uiþ1ðtÞ . . . y_NðtÞ ¼ uNðtÞ _ uNþ1ðtÞ where y_iðtÞ ¼ d½yiðtÞ_=dt is the rate of buffer level variations and yi(t) is the buffer level that is subject to the constraint 0 _ yi(t) _ Li. The effects of this constraint are as the following: (1) If yi(t) ¼ 0, then the buffer stock remains empty and unchanged (i.e. y_iðtÞ ¼ 0), until the inflow exceeds the outflow (i.e. ui(t) 4 uiþ1(t), the throughput rate of upstream machine is greater than the downstream machine). (2) Similarly, if yi(t) ¼ Li, then buffer stock remains unchanged until ui(t) 5 uiþ1(t).

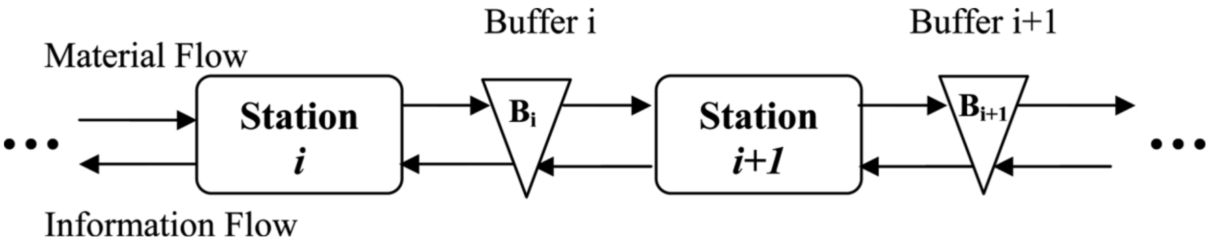


Figure 3.3 Flow diagram

This present investigation about data-driven continuous low model aims to capture the SD of the multistage assembly line. As shown in Figure 4, two models, continuous flow model and DES model, are built based upon the actual data (i.e. processing velocity, MTBF and MTTR of each machine) from a real-world automotive assembly line. In this proposed continuous flow model, by considering the production flow and density of working parts, the variation in external stimuli (e.g. increase of buffer limits or decrease of input flow in the first station) will result in the change of production material density instead of dealing with individual working parts. Statistical analysis of both model outputs, i.e. variations of throughput rate and buffer levels, is used to compare performance statistics from simulation model and realworld assembly line. The basic nomenclature and relevant relationships pertinent to the continuous flow modelling of assembly line operations are delineated in the following implementation section.

3.4 Design & Implementation Constraints

In an automotive manufacturing plant, an assembly line has various layout configurations based on the material flow between departments. Examples of assembly line layout configurations include serial lines, U-shaped lines, parallel stations, work-centres, onesided and two-sided assembly lines, etc. It may be noted that two-sided assembly lines are typically found in producing large-sized products such as trucks and buses. The proposed continuous flow approach treats the part movement as a fluid flow, buffer stocks as water tanks, the conveyor belt as water pipe and manufacturing stations as the valves which control the rates of flow. Specifically, we do not account for the individual parts in different structures of assembly lines, but the flow rates and processing velocities that will be determined by the structure of production system.

In the present investigation, the assembly line is to manufacture the powertrain system, which is the most important component in automobiles. The powertrain system includes the engine, transmission, drive shafts and drive wheels, etc. The structure of this assembly line segment consists of 18 stations of which 17 are allocated in tandem. One pair of stations is located in a parallel arrangement. The main model implementation in Simulink environment is illustrated in Figure 5. The continuous flow model captures the flow of materials through each machine as well as the effects of various state transitions in the flow. The buffer stock Bi will inform the upstream machine (i.e. station i) to block the operation when the buffer limit is reached. The downstream machine (i.e. station i þ 1) will be starved if buffer stock Bi is empty. The arriving velocity of materials depends on the processing velocity of upstream machine and the variations of real-time buffer levels. In spite of these simple rules, such a continuous flow model results in complex SD behaviours, especially when external stimuli (e.g. demand dynamics, buffer limit adjustments, etc.) are presented.

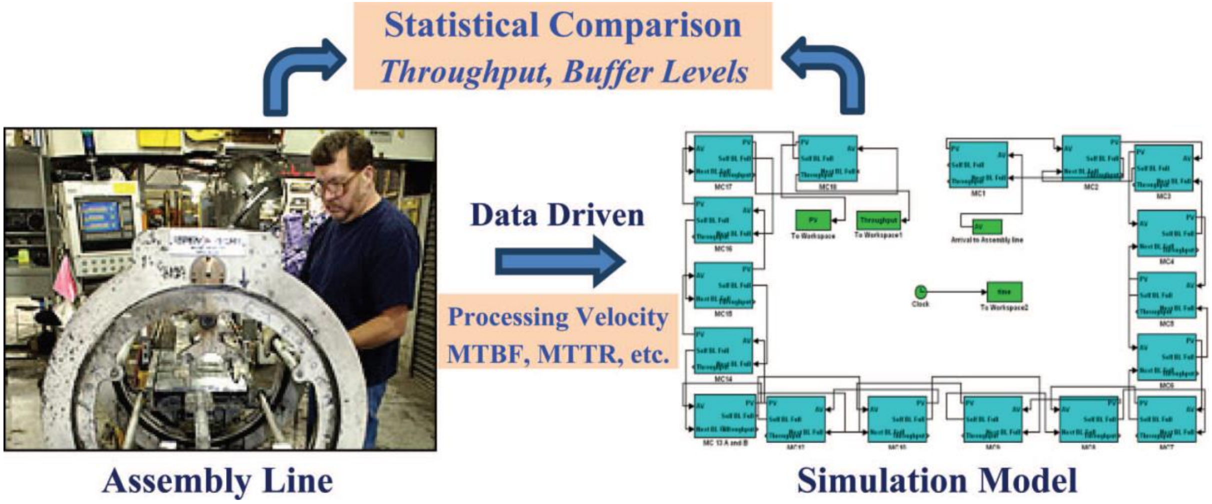


Figure 3.4 Overview

Parallel station operations, as shown in Figure 6, are encapsulated into a single station in the segment. The individual components of the station are designed so that material flows from the upstream are forked through all the parallel stations. The continuous flow is splitted according to the processing rates of individual stations, e.g. station iA will receive the proportion PVA/(PVA þ PVB) of the material flow and station iB will receive the proportion of PVB/(PVA þ PVB) in the case of two parallel machines. In addition, stations iA and iB will share the same buffer stock Bi, which will inform the two parallel machines of the block information. Both station iA and iB will be starved if the upstream buffer stock is empty.

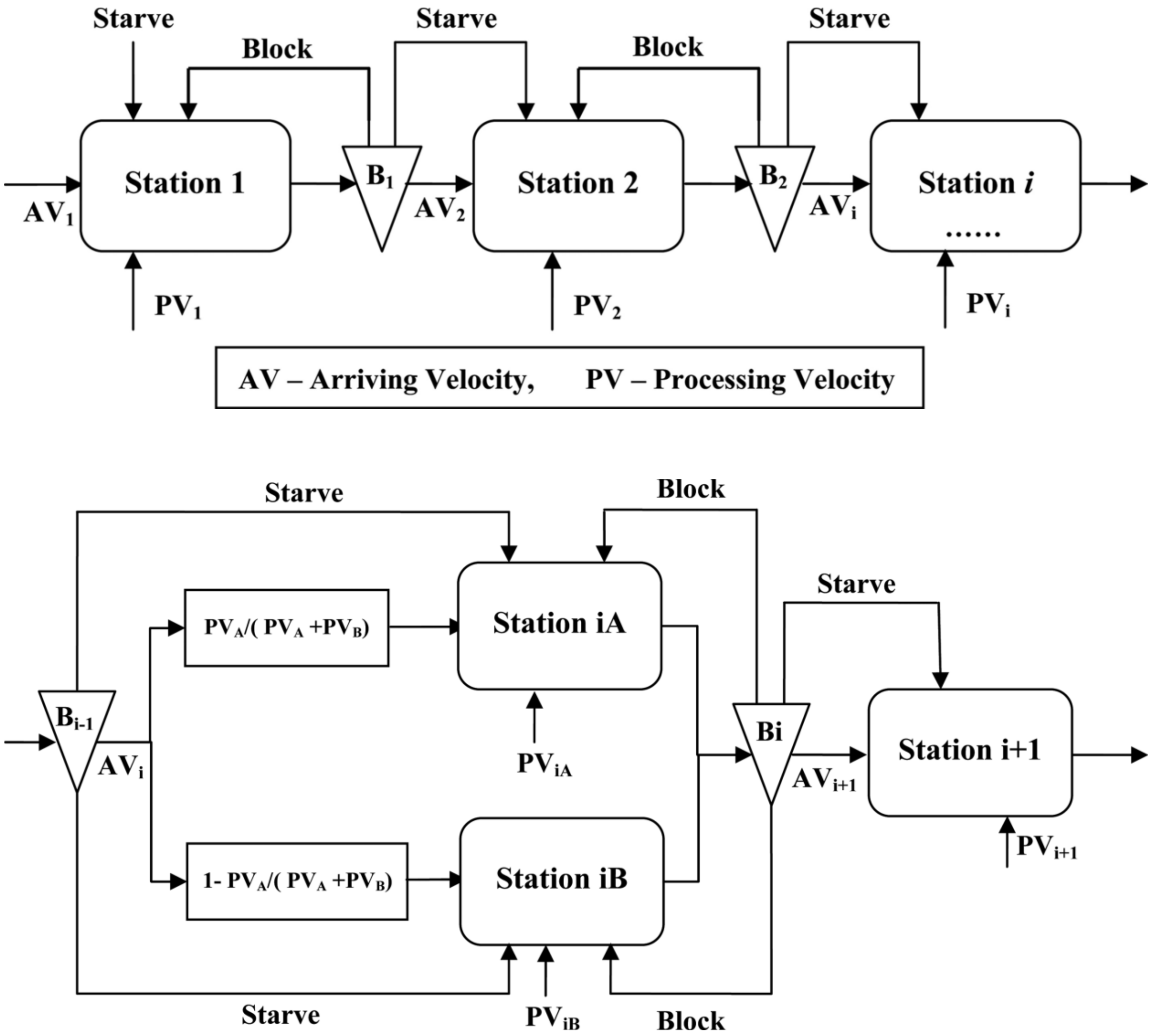


Figure 3.5 Machine Subsystem

For each machine subsystem i, the throughput is calculated as the integration of the rate of buffer level changes y_iðtÞ. Starve states are determined locally depending on whether the upstream buffer is empty. Blocking is indicated by the current station's buffer level (see Figure 6). If the maximum of buffer level Li is reached, then the buffer stock remains unchanged until ui(t) 5 uiþ1(t). In other words, the following conditions hold: (1) If ui(t) 7 uiþ1(t) 4 0, then the buffer stock simply fills up at some time TL and yi(t) ¼ Li for all t 4 TL. (2) If ui(t) 7 uiþ1(t) 5 0, then the buffer stock becomes empty at some time T0 and yi(t) ¼ 0 for all t 4 T0. (3) If ui(t) 7 uiþ1(t) ¼ 0, then the buffer stock remains unchanged.

The up and down times of various stations are taken from historical data or by generating random variables TTTR (Time To Repair) and TTBF (Time Between Failure) from the actual MTBF and MTTR statistics over a specified simulation epoch. The basic logic is summarised in Figure 7. Here, TST is the total simulation time, Tperiod(i,j) is the time point at the end of machine i's jth period consisting of Time Between Failure TTBF(i,j) and Time To Repair TTTR(i,j), TWork(i,j) is the time point when status changing from working to failure in machine i's jth period. This part is executed prior to the simulation of the line segment and those random generated parameters are used as inputs to the Simulink simulation model. If one has access to actual downtime realisations, the TWork(i,j) and TPeriod(i,j) arrays can be computed from actual values instead of generating exponentially distributed random variable sequences.

3.5 Proposed System

In this subsystem, based on the recorded TWork(i,j) and TPeriod(i,j) sequences, we calculate the real time throughput rate ui(t) by comparing the current time clock of simulation with TWork(i,j) and TPeriod(i,j) sequences. The essence of this model involves partitioning the time axis into non-overlapping TPeriod(i,j). Each period consists of a time location TWork(i,j) that marks the uptime of a station between two successive breakdowns. During this time interval, ui is set to the actual processing velocity mi for the specified station. The remaining time between the TWork(i,j) and TPeriod(i,j) of a work cycle marks the downtime. Here, ui is set to zero.

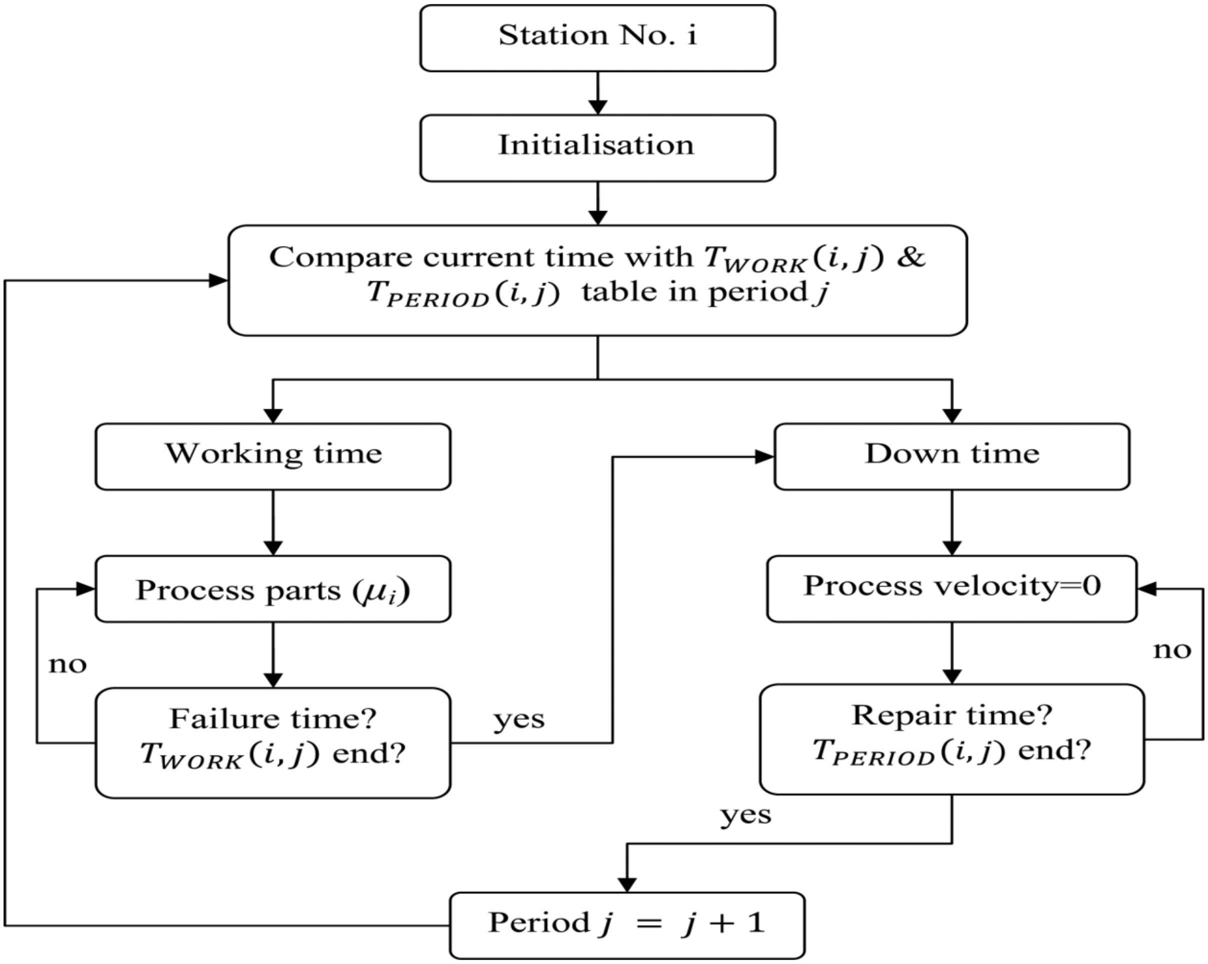

Station No. i
Initialisation
Compare current time with $T_{WORK}(i,j)$ & $T_{PERIOD}(i,j)$ table in period j
Working time
Down time
Process parts (μ_i)
Process velocity=0
no
no
Failure time? $T_{WORK}(i,j)$ end?
yes
Repair time? $T_{PERIOD}(i,j)$ end?
yes
Period $j = j + 1$

Figure 3.6 Proposed System

As the algorithm flow chart shown in Figure 8, real-time throughput rate ui(t) is adjusted based on the TWork(i,j) and TPeriod(i,j) matrix. In the initialisation stage, those input parameters (i.e. MTBF, MTTR, initial buffer length, processing velocity) are imported into the simulation model and the first period is started. Tim

CHAPTER 4

SYSTEM DESIGN

4.1 Introduction

In an automotive manufacturing plant, an assembly line has various layout configurations based on the material flow between departments. Examples of assembly line layout configurations include serial lines, U-shaped lines, parallel stations, work-centers, one-sided and two-sided assembly lines, etc. It may be noted that two-sided assembly lines are typically found in producing large-sized products such as trucks and buses. The proposed continuous flow approach treats the part movement as fluid flow, buffer stocks as water tanks, the conveyor belt as a water pipe, and manufacturing stations as the valves which control the rates of flow. Specifically, we do not account for the individual parts in different structures of assembly lines, but the flow rates and processing velocities that will be determined by the structure of the production system. In the present investigation, the assembly line is to manufacture the powertrain system, which is the most important component in automobiles. The powertrain system includes the engine, transmission, drive shafts and drives wheels, etc. The structure of this assembly line segment consists of 18 stations of which 17 are allocated in tandem. One pair of stations is located in a parallel arrangement. The main model implementation in Simulink environment is illustrated in Figure 2. The continuous flow model captures the flow of materials through each machine as well as the effects of various state transitions in the flow. The buffer stock Bi will inform the upstream machine (i.e. station i) to block the operation when the buffer limit is reached. The downstream machine (i.e. station I þ 1) will be starved if buffer stock Bi is empty. The arriving velocity of materials depends on the processing velocity of the upstream machine and the variations of real-time buffer levels. Despite these simple rules, such a continuous flow model results in complex SD behaviors, especially when external stimuli (e.g. demand dynamics, buffer limit adjustments, etc.) are presented.

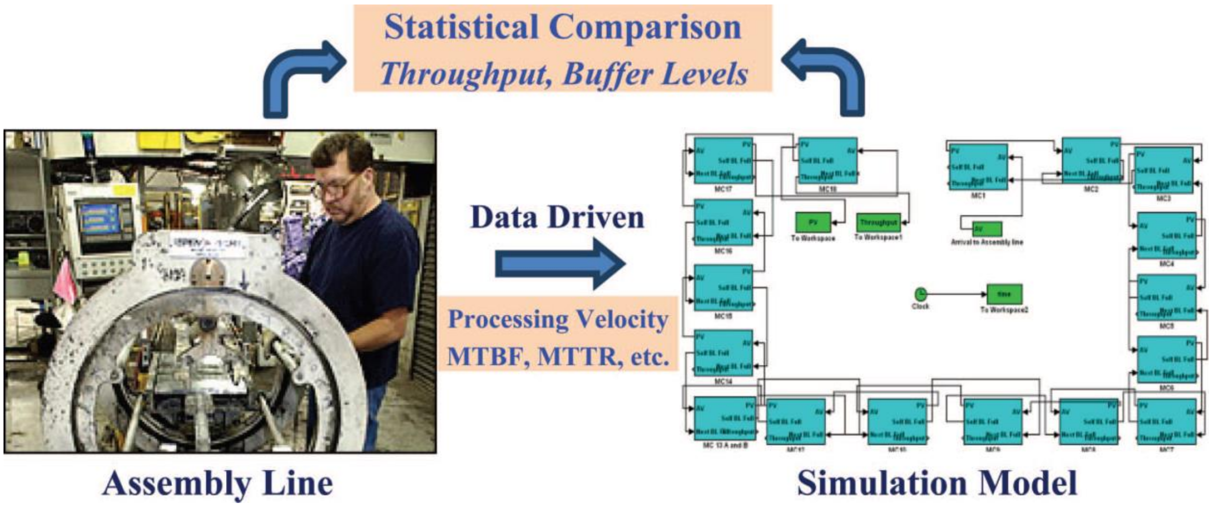


Figure 4.1 Manufacturing Structure

4.2 System Architecture

Parallel station operations, as shown in Figure 3, are encapsulated into a single station in the segment. The individual components of the station are designed so that material flows from the upstream are forked through all the parallel stations. The continuous flow is split according to the processing rates of individual stations, e.g.

station iA will receive the proportion PVA/(PVA þ PVB) of the material flow and station iB will receive the proportion of PVB/(PVA þ PVB) in the case of two parallel machines. Also, stations iA and iB will share the same buffer stock Bi, which will inform the two parallel machines of the block information. Both station iA and iB will be starved if the upstream buffer stock is empty.

4.3 ER Diagram

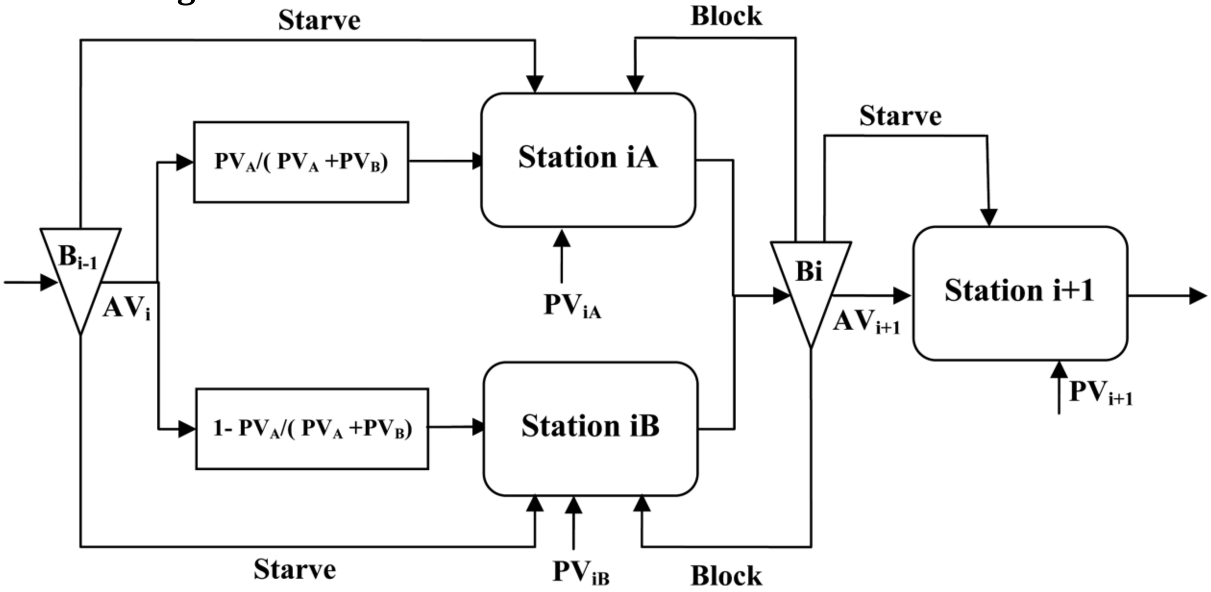


Figure 4.2 ER Diagram

4.4 Summary

For each machine subsystem i, the throughput is calculated as the integration of the rate of buffer level changes y_iðtÞ. Starve states are determined locally depending on whether the upstream buffer is empty. Blocking is indicated by the current station's buffer level (see Figure 6). If the maximum of buffer level Li is reached, then the buffer stock remains unchanged until ui(t) 5 uiþ1(t). In other words, the following conditions hold: (1) If ui(t) 7 uiþ1(t) 4 0, then the buffer stock simply fills up at some time TL and yi(t) ¼ Li for all t 4 TL. (2) If ui(t) 7 uiþ1(t) 5 0, then the buffer stock becomes empty at some time T0 and yi(t) ¼ 0 for all t 4 T0. (3) If ui(t) 7 uiþ1(t) ¼ 0, then the buffer stock remains unchanged.

Algorithm

The up and downtimes of various stations are taken from historical data or by generating random variables TTTR (Time To Repair) and TTBF (Time Between Failure) from the actual MTBF and MTTR statistics over a specified simulation epoch. The basic logic is summarised in Figure 7. Here, TST is the total simulation time, Tperiod(i,j) is the time point at the end of machine i's jth period consisting of Time Between Failure TTBF(i,j) and Time To Repair TTTR(i,j), TWork(i,j) is the time point when status changing from working to failure in machine i's jth period. This part is executed before the simulation of the line segment and those randomly generated parameters are used as inputs to the Simulink simulation model. If one has access to actual downtime realizations, the TWork(i,j) and TPeriod(i,j) arrays can be computed from actual values instead of generating exponentially distributed random variable sequences.

Results

An equivalent DES model is also implemented in Arena for the assembly line segment of 18 stations (see Figure 4). The DES model consists of the following five sub-components, i.e. entity arriving, first machine, middle 16 machines, end machine, and release. Parallel machines are inside the submodel of MidMachine. The Arena view of the complete DES model is shown in Figure 9. For a single machine, the operational logic is implemented with a ‘seize-delay-hold-release’ mechanism. In other words, the machine will ‘seize’ the entity from the upstream buffer, and the processing results in the ‘delay’ of an entity in the machine. The machine will ‘hold’ the operation due to block, starve, or breakdown, i.e. the model logic will hold the entity inside the machine while the downstream buffer is full. Inside the expression module, 18 6 4 matrices are used to store the relevant parameters including MTBF, MTTR, processing time, and buffer limit for each machine. The DES model was replicated 30 times during the simulation and the average statistics were analyzed for the total jobs passing through each machine. In the continuous flow models, the choice of integration routine and the step-size has a significant effect on the speed and quality of simulations. It is pertinent to note that the models are nonlinear due to the presence of hard-type saturations (captured in the switch functions).

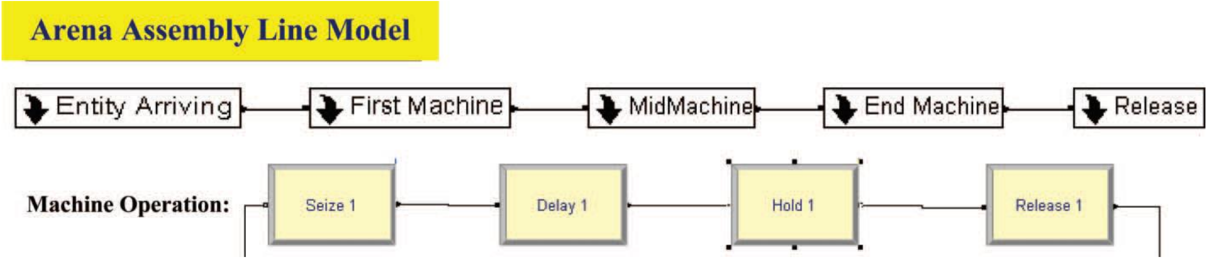
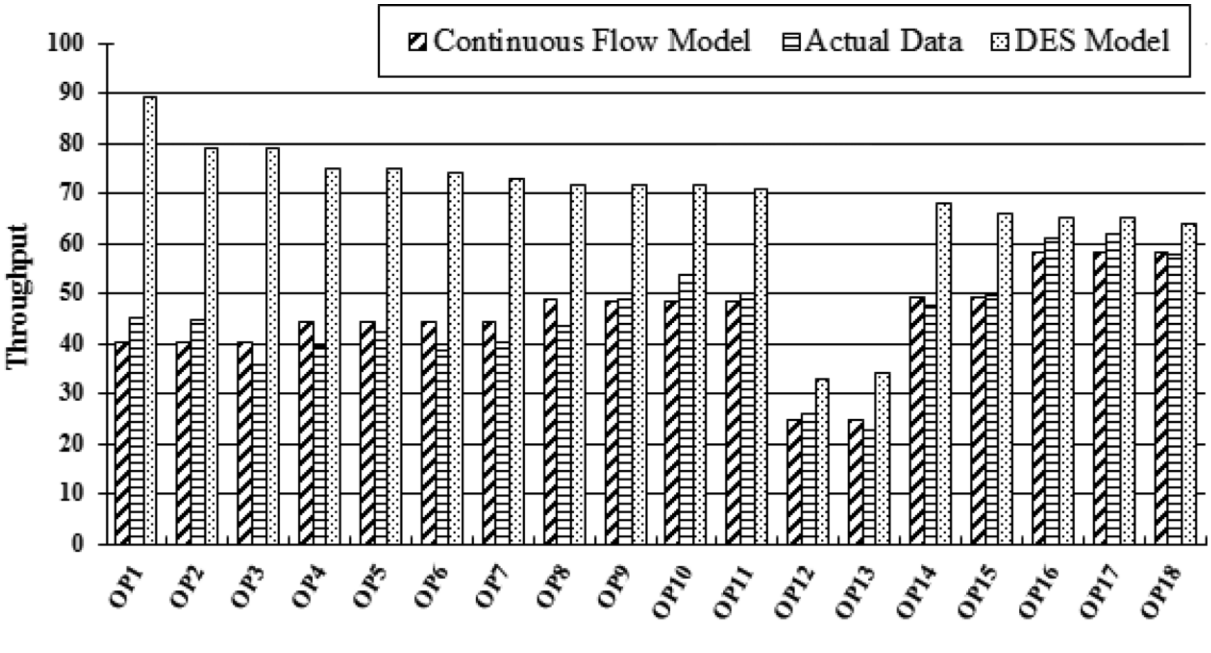
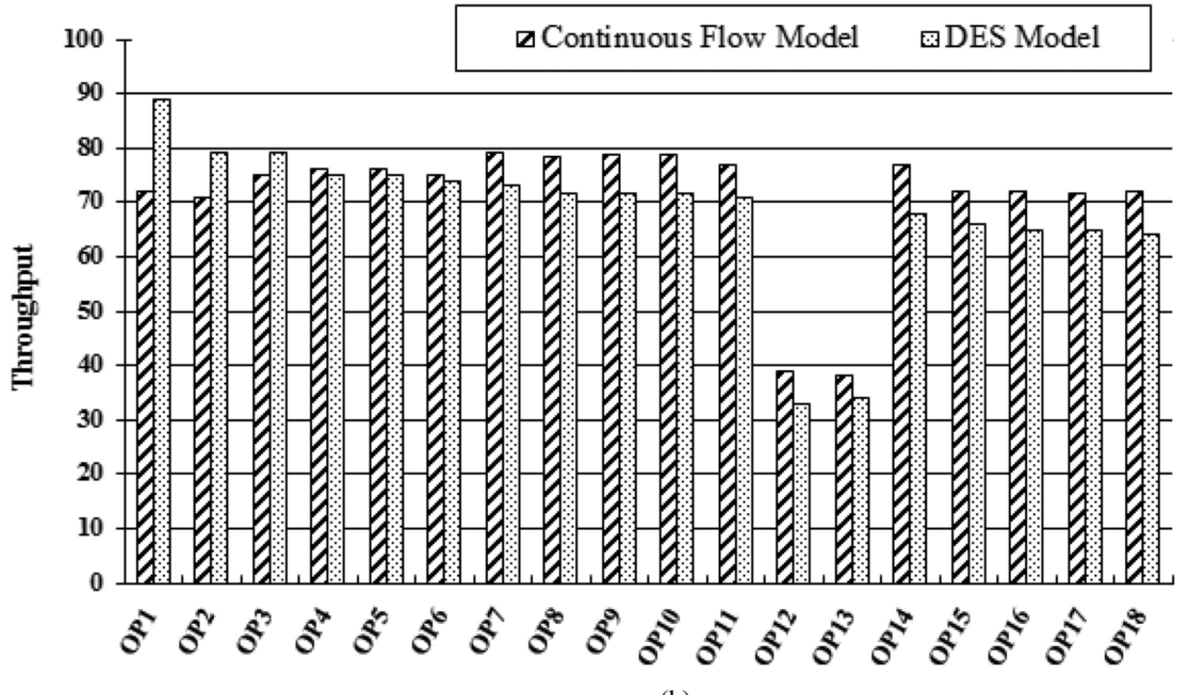


Figure 4.3 Arena Assembly Line Model

The random processing, down and repair times render the dynamics of processing velocities and the inventory levels stochastic. In this context, the integration routines are used to invert nonlinear stochastic differential equations. The usual smoothness assumptions that underpin the variable step integration routines used in the Simulink implementations may not hold in this context. As a result, the use of variable step built-in numerical integration routines have led to significant slowing down of the simulations due to the need for extremely small step sizes to hedge against random perturbations. Table 2 shows a comparison of simulation running time for various built-in solver types. We selected the fixed step-size integrator with a step time of 1 min. As shown in Table 2, the ode1 (Euler) solver yielded a minimum running time of 10 s for the simulation of 20 h operation. Also, the simulation of a 1-week operation takes 324 s for the fixed step time of 1 min and 161 s for the 2-min step time. Thus, the Euler solver is chosen to enable a fast simulation and the fixed step time of 1 min is used in this investigation.



(a)



(b)

Figure 4.4 Flow Model

Conclusions

The present work attempted to use the SD approach to model the real-world multistage assembly line. We developed the data-driven continuous flow model using continuous flow approximation for assembly work transfers. The part movement is treated as fluid flow, buffer stocks are water tanks, the conveyor belt is a water pipe, and manufacturing stations are the valves that control the rates of flow. Machine states, i.e. breakdown, starved, and blocked, are treated as discrete events that cause variations in the material flow rate. The statistics of throughput rates match closely with those from previous DES implementations and real-world historical results. It is shown that the proposed continuous flow model can capture the instantaneous (dynamic) throughput rate in a faster manner, and also closely match the DES output during the steady-state. The continuous flow model offers flexibility in choosing the initial states and the actual breakdown and repair times. The model for most cases reached a steady-state within 100 min of simulation time during different initial conditions. The mean throughput values for the presented continuous flow model were within 5% of those from the real world assembly historical data. Future work includes improving computational efficiency, performing characterizations to yield appropriate prediction model structures, implementing real-time prediction routines, and using continuous flow models for performance estimation in the real-world operation of the multistage assembly line.

CHAPTER 5

MODULE DESCRIPTION

5.1 Introduction

The automobile industry, along with the auto components industry, is one of the core industries in India. A well developed transportation system plays a key role in the development of an economy, and India is no exception to it. Automobile is one of the largest industries in the global market. Owing to its strong forward and backward linkages with several key segments of the economy. Automobile Sector occupies a prominent place in the fabric of Indian Economy. Automobile sector is leader in product and process technologies in the manufacturing sector. It has been recognized as one of the drivers of economic growth and the domestic automobile industry is believed to be the barometer of the economy. Such a belief is in line with international trends since in most mature economies the automobile industry’s performance is viewed as a reflection of the economy’s health. This sector has emerged as sunrise sector in the Indian economy. According to data published by Department of Industrial Policy and Promotion (DIPP), ministry of Commerce, the amount of cumulative foreign direct investment (FDI) inflow into the auto sector from April 2000 to November 2012 was worth US$7,518 million. The auto sector accounts for 4 per cent of the total FDI Inflows (in terms of US $) in India. According to the recent data released by Society of Indian Automobiles Manufacturers (SIAM) India’s scooter and motorcycle manufacturers have registered 4 per cent growth during April-November, 2012. The Global and Indian manufacturers are focusing their efforts to develop innovative products, technologies and supply chains. India is one of the key markets for Global Manufacturers for hybrid and electronic vehicles, which is the new development in automobile sector. With a turnover of almost $59 Million US Dollars, Automobile industry Provides employment to 13 million people in the India Work-class. The automobiles sector is divided into four segments - two-wheelers, passenger vehicles, commercial vehicles and three wheelers. Two wheelers India is one of the world’s fastest growing passenger car markets it is second largest two wheeler manufacturer and fifth largest commercial vehicle manufacturer. It is also home for the largest motor cycle manufacturer. Moreover, India is fourth largest passenger car market in Asia.

The auto sector in India has achieved a growth rate of 26% in last two years (2010-12). However, it has shown a sluggish growth of 12 percent in 2012. The trend is likely to stay with a 10 percent growth outlined for 2013. The main reason are high ownership costs (fuel costs, cost of registration, excise duty, road tax) and slow rural income growth. Over the next few Years solid but cautious growth is expected. The Macquarie equities research reveals that the sale of passenger vehicles is expected to double in the next four years and growth anticipated is higher than the 16% achieved in the past 10 years. The automotive Mission Plan 2016 launched by the Government of India seeks to grow the industry to a size of $145 billion by 2016 and make it contribute 10 per cent to the nation’ GDP. The growth for automotive industry is important for growth in economy, particularly because the automotive industry has strong multiplier effect. It is capable if being the driver if economic growth. High direct to indirect employment ratio of about 1:10 Is estimated for the automobile industry, because automobile industry has potential to generate employment for about 10 more for every person employed directly in automobile manufacturing industry. These indirect employments includes employments in ancillary and component industries, automobile service stations mechanics, loaders and cleaners of commercial vehicles, institutions financing purchase of vehicles and people who drive commercial vehicles and hired vehicles. There is a symbiotic relationship between the growth of economy and the demand for vehicles.

5.2 Baseline Model

At first, several models were implemented without any hyperparameter tuning to create a baseline. The model training was performed on a machine with an Intel Core i7-9700K, 2x8GB 4000MHz DDR4 memory and an NVIDIA GeForce RTX 2070. As hypothesized during the exploratory data analysis from Section III-B, the two linear models, Logistic Regression and Naive Bayes, performed significantly worse than the worst performing non-linear classifier, suggesting a stronger non-linear relationship between the features and the target. Based on this, 5-fold cross validation was performed on the four non-linear classifiers in order to obtain a more realistic measure of accuracy and avoid over writing on the training data. From the observation of Figure 6, it can be said that the baseline RF model performed better on average across allmetrics except for recall, for which XGBoost was considerably
superior. The XGBoost and SVM models performedslightly worse, with KNN performing considerably worse overall. One particularity to take into account in this MMP is the possibility of the feature distributions to change over time.
This can happen for several reasons and is further discussedin Section V, but to tackle this challenge, an approach could be to monitor the accuracy of the deployed model and retrain it if it drops below a certain threshold. This means that morecomputationally expensive models that take longer to trainand perform cross validation on, like SVM, might not be
adequate for such a scenario.Based on this, hyperparameter tuning through randomized
search was performed on the three best models, which were then compared on the test set based on the same evaluation metrics used for cross validation. The tuned parameters
can be found in Table, where any omitted parameters

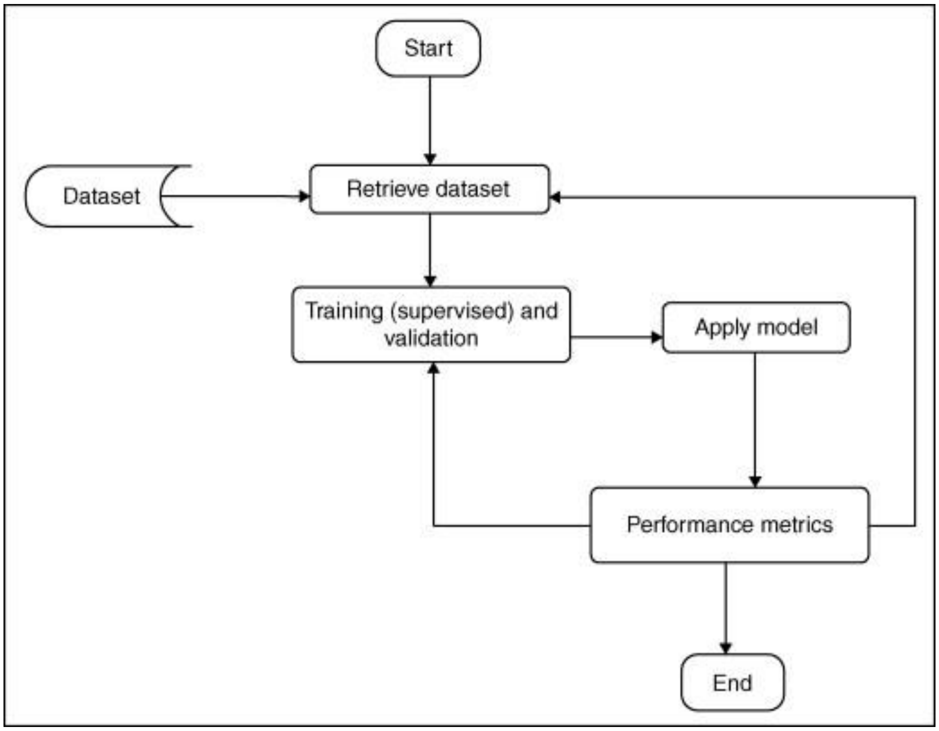


Figure 5.1 Baseline Model

5.3 RF Model

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark (as of 2019, owned by Minitab, Inc.). The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho and later independently by Amit and Geman in order to construct a collection of decision trees with controlled variance.

5.4 IDARTS Framework

The manufacturing industry represents a data rich environment, in which larger and larger volumes of data are constantly being generated by its processes. However, only a relatively small portion of it is actually taken advantage of by manufacturers. As such, the proposed Intelligent Data Analysis and Real-Time Supervision (IDARTS) framework presents the guidelines for the implementation of scalable, flexible and pluggable data analysis and real-time supervision systems for manufacturing environments. IDARTS is aligned with the current Industry 4.0 trend, being aimed at allowing manufacturers to translate their data into a business advantage through the integration of a Cyber-Physical System at the edge with cloud computing. It combines distributed data acquisition, machine learning and run-time reasoning to assist in fields such as predictive maintenance and quality control, reducing the impact of disruptive events in production.

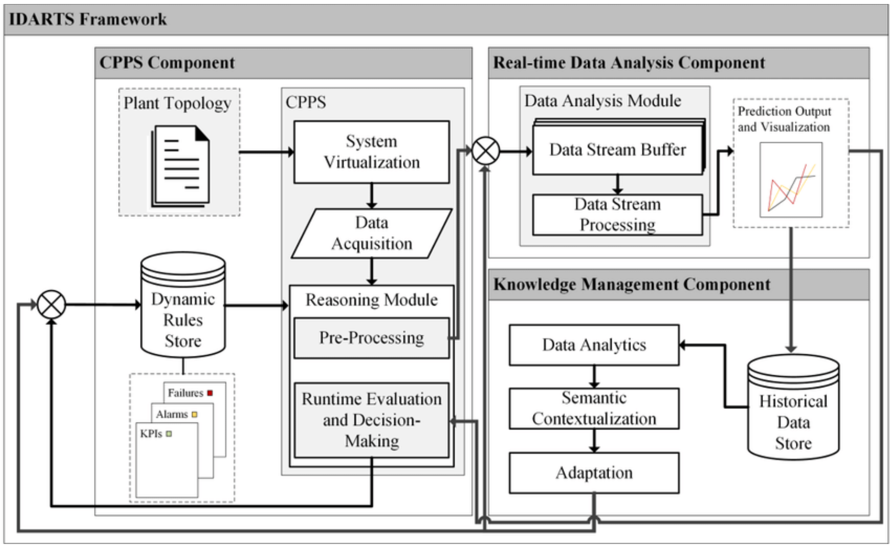


Figure 5.2 IDARTS Framework

5.5 Summary

The model fitting is under progress. The hypothesized structural equation model, is being fitted using generalized least square (GLS) method. Depending on the result of model fitting, a model modification might be conducted with some further investigation. Once we obtain a fitted model with a reasonable level of statistical significance, we can decide which hypotheses are supported in this particular loyalty program case. As a result, the hypotheses mentioned in Section 3.1 will be validated. At this moment, it can be decided which services in the loyalty program have significant effects on customer satisfaction, brand preference, and switching cost as well as on customer loyalty. Then, we will interpret the meaning of the obtained path coefficients from a practical point of view. Finally, we will seek to extract some new insights as to the current status and future strategy of the loyalty program based on all the analysis results. The effect of motor vehicle manufacturing on other industries is very great. Almost one-fifth of American steel production and nearly three-fifths of its rubber output go to the automotive industry, which is also the largest single consumer of machine tools. Moreover, the special requirements of automotive mass production have had a profound influence on the design and development of highly specialized machine tools and have stimulated technological advances in petroleum refining, steelmaking, paint and plate-glass manufacturing, and other industrial processes.
The indirect effects are also considerable through the many auto-related businesses, such as motor freight operators and highway construction firms. In addition, truck transportation has grown steadily throughout the world.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Introduction

Data set information

This data set consists of three types of entities: (a) the specification of an auto in terms of various characteristics, (b) its assigned insurance risk rating, (c) its normalized losses in use as compared to other cars. The second rating corresponds to the degree to which the auto is more risky than its price indicates. Cars are initially assigned a risk factor symbol associated with its price. Then, if it is more risky (or less), this symbol is adjusted by moving it up (or down) the scale. Actuarians call this process "symboling". A value of +3 indicates that the auto is risky, -3 that it is probably pretty safe.

The third factor is the relative average loss payment per insured vehicle year. This value is normalized for all autos within a particular size classification (two-door small, station wagons, sports/speciality, etc...), and represents the average loss per car per year.

Note: Several of the attributes in the database could be used as a "class" attribute.

Attribute Information

Attribute : Attribute Range

1. symboling: -3, -2, -1, 0, 1, 2, 3.
2. normalized-losses: continuous from 65 to 256.
3. make: alfa-romero, audi, bmw, chevrolet, dodge, honda, isuzu, jaguar, mazda, mercedes-benz, mercury, mitsubishi, nissan, peugot, plymouth, porsche, renault, saab, subaru, toyota, volkswagen, volvo
4. fuel-type: diesel, gas.
5. aspiration: std, turbo.
6. num-of-doors: four, two.
7. body-style: hardtop, wagon, sedan, hatchback, convertible.
8. drive-wheels: 4wd, fwd, rwd.
9. engine-location: front, rear.
10. wheel-base: continuous from 86.6 120.9.
11. length: continuous from 141.1 to 208.1.
12. width: continuous from 60.3 to 72.3.
13. height: continuous from 47.8 to 59.8.
14. curb-weight: continuous from 1488 to 4066.
15. engine-type: dohc, dohcv, l, ohc, ohcf, ohcv, rotor.
16. num-of-cylinders: eight, five, four, six, three, twelve, two.
17. engine-size: continuous from 61 to 326.
18. fuel-system: 1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi.
19. bore: continuous from 2.54 to 3.94.

20. stroke: continuous from 2.07 to 4.17.
21. compression-ratio: continuous from 7 to 23.
22. horsepower: continuous from 48 to 288.
23. peak-rpm: continuous from 4150 to 6600.
24. city-mpg: continuous from 13 to 49.
25. highway-mpg: continuous from 16 to 54.
26. price: continuous from 5118 to 45400.

6.2 System Implementation

Code

```
import pandas as pd
import numpy as np
import seaborn as sns
sns.set( palette="muted", color_codes=True,style="whitegrid")

import matplotlib.pyplot as plt
df_am_losses = pd.read_csv('DATASET/Automobile/automobile-losses.csv')
df_am_risk = pd.read_csv('DATASET/Automobile/automobile-risk.csv')
df_am_spec = pd.read_csv('DATASET/Automobile/automobile-spec.csv')
```

```
df_am_losses.head()
```

ID	normalized-losses	
0	1	NaN
1	2	NaN
2	3	NaN
3	4	164.0
4	5	164.0

Figure 6.1 df_am_losses table

```
df_am_risk.head()
```

ID	symboling	
0	1	3
1	2	3
2	3	1
3	4	2
4	5	2

Figure 6.2 df_am_risk table

```
df3.dtypes
```

```
ID                    int64
make                  object
fuel-type             object
aspiration            object
num-of-doors          object
body-style            object
drive-wheel           object
engine-loc            object
wheel-base            float64
length                float64
width                 float64
height                float64
curb-weight           int64
engine-type           object
cylinder              object
engine-size           int64
fuel-system           object
bore                  float64
stroke                float64
compression-ratio     float64
```

```
horsepower          float64
peak-rpm            float64
city-mpg              int64
highway-mpg           int64
price               float64
symboling             int64
normalized-losses   float64
dtype: object
```

Check Shape

```
df3.shape
(205, 27)
```

Check duplicate row(s)

```
duplicate_rows_df = df3[df3.duplicated()]
print('number of duplicate rows: ', duplicate_rows_df.shape)
#df3 = df3.drop_duplicates()
number of duplicate rows:  (0, 27)
```

3. Identify missing value. If any, how will you handle it?

```
df3['num-of-doors'].fillna(value=df3['num-of-doors'].value_counts().index[0],inplace =True)
df3['bore'].fillna(value=df3['bore'].median(),inplace =True)
df3['stroke'].fillna(value=df3['stroke'].median(),inplace =True)
df3['horsepower'].fillna(value=df3['horsepower'].median(),inplace =True)
df3['peak-rpm'].fillna(value=df3['peak-rpm'].median(),inplace =True)
df3['price'].fillna(value=df3['price'].median(),inplace =True)
df3['normalized-losses'].fillna(value=df3['normalized-losses'].median(),inplace =True)
```

Visualize

Perform visualization using at least 5 difference visualization technique (barplot, scatter plot, area, boxplot, pie chart, line chart, etc)
Countplot

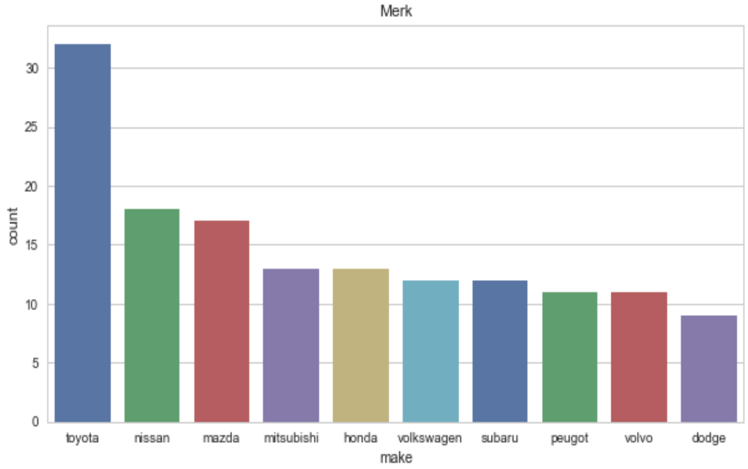


Figure 6.3 visualization

```
plt.figure(figsize=(10,5))
fig = sns.countplot(x= df3['make'],palette='deep', label = 'Merk',order = df3['make'].value_counts(ascending = False).index[:10]).set_title('Merk')
```

Distplot

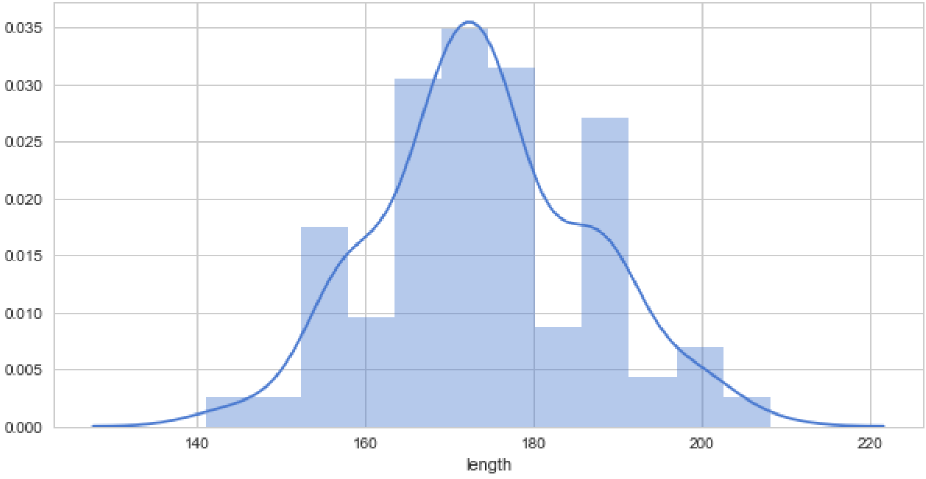


Figure 6.4 Distplot

```
plt.figure(figsize=(10,5))
fig = sns.distplot(df3['length'])
```

```
# size = df3[['length','width','height']]
# size.head()
# top3_car = df3['make'].value_counts(ascending = False).index[:3]
# top3_car
# plt.figure(figsize=(10,5))
# fig = sns.distplot(x='size',hue='top3_car')
```

Violinplot

```
plt.figure(figsize=(10,5))
ax = sns.violinplot(x="fuel-type", y="price", hue="aspiration",
                    data=df3, palette="Set2", split=True,
                    scale="count", inner="quartile")
```

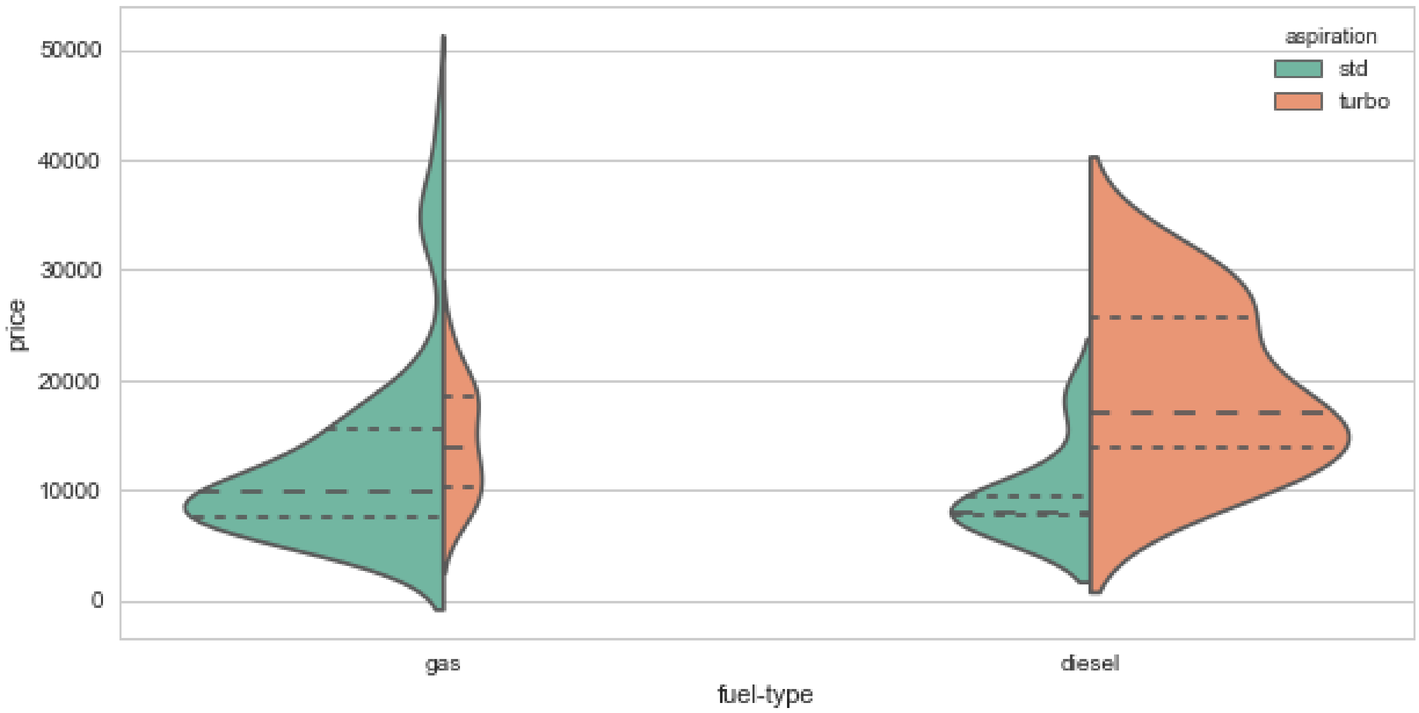
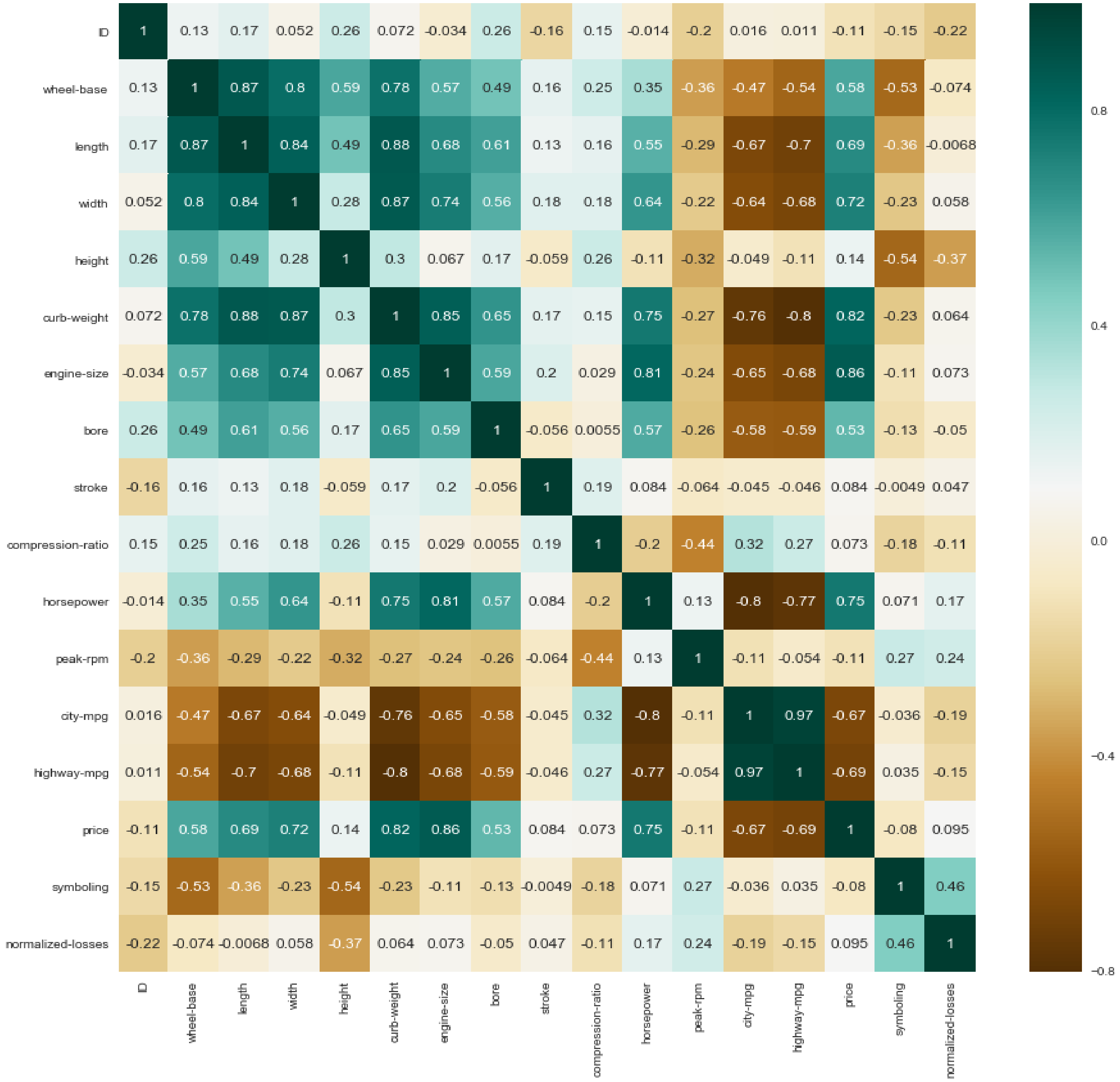


Figure 6.5 Violinplot

Heatmap

```
plt.figure(figsize=(15,15))
c= df3.corr()
sns.heatmap(c,cmap='BrBG',annot=True)
<matplotlib.axes._subplots.AxesSubplot at 0x1423849c128>
```



	ID	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price	symboling	normalized-losses
ID	1	0.13	0.17	0.052	0.26	0.072	-0.034	0.26	-0.16	0.15	-0.014	-0.2	0.016	0.011	-0.11	-0.15	-0.22
wheel-base	0.13	1	0.87	0.8	0.59	0.78	0.57	0.49	0.16	0.25	0.35	-0.36	-0.47	-0.54	0.58	-0.53	-0.074
length	0.17	0.87	1	0.84	0.49	0.88	0.68	0.61	0.13	0.16	0.55	-0.29	-0.67	-0.7	0.69	-0.36	-0.0068
width	0.052	0.8	0.84	1	0.28	0.87	0.74	0.56	0.18	0.18	0.64	-0.22	-0.64	-0.68	0.72	-0.23	0.058
height	0.26	0.59	0.49	0.28	1	0.3	0.067	0.17	-0.059	0.26	-0.11	-0.32	-0.049	-0.11	0.14	-0.54	-0.37
curb-weight	0.072	0.78	0.88	0.87	0.3	1	0.85	0.65	0.17	0.15	0.75	-0.27	-0.76	-0.8	0.82	-0.23	0.064
engine-size	-0.034	0.57	0.68	0.74	0.067	0.85	1	0.59	0.2	0.029	0.81	-0.24	-0.65	-0.68	0.86	-0.11	0.073
bore	0.26	0.49	0.61	0.56	0.17	0.65	0.59	1	-0.056	0.0055	0.57	-0.26	-0.58	-0.59	0.53	-0.13	-0.05
stroke	-0.16	0.16	0.13	0.18	-0.059	0.17	0.2	-0.056	1	0.19	0.084	-0.064	-0.045	-0.046	0.084	-0.0049	0.047
compression-ratio	0.15	0.25	0.16	0.18	0.26	0.15	0.029	0.0055	0.19	1	-0.2	-0.44	0.32	0.27	0.073	-0.18	-0.11
horsepower	-0.014	0.35	0.55	0.64	-0.11	0.75	0.81	0.57	0.084	-0.2	1	0.13	-0.8	-0.77	0.75	0.071	0.17
peak-rpm	-0.2	-0.36	-0.29	-0.22	-0.32	-0.27	-0.24	-0.26	-0.064	-0.44	0.13	1	-0.11	-0.054	-0.11	0.27	0.24
city-mpg	0.016	-0.47	-0.67	-0.64	-0.049	-0.76	-0.65	-0.58	-0.045	0.32	-0.8	-0.11	1	0.97	-0.67	-0.036	-0.19
highway-mpg	0.011	-0.54	-0.7	-0.68	-0.11	-0.8	-0.68	-0.59	-0.046	0.27	-0.77	-0.054	0.97	1	-0.69	0.035	-0.15
price	-0.11	0.58	0.69	0.72	0.14	0.82	0.86	0.53	0.084	0.073	0.75	-0.11	-0.67	-0.69	1	-0.08	0.095
symboling	-0.15	-0.53	-0.36	-0.23	-0.54	-0.23	-0.11	-0.13	-0.0049	-0.18	0.071	0.27	-0.036	0.035	-0.08	1	0.46
normalized-losses	-0.22	-0.074	-0.0068	0.058	-0.37	0.064	0.073	-0.05	0.047	-0.11	0.17	0.24	-0.19	-0.15	0.095	0.46	1

Figure 6.6 Heatmap

6.3 Algorithms

Gaussian Naive Bayes

The Gaussian Naive Bayes (GNB) method is a supervised learning algorithm based on the Bayes' theorem with the naive assumption of conditional independence between the various pairs of features given the value of the target variable. The *GaussianNB* class from scikit-learn implements GNB for classification, with the likelihood of the features assumed to be Gaussian:

P(*xi* j *y*) D
1
q
2__2
y
exp
_
□
(*xi* □ _*y*)2
2_2
Y

where the parameters _*y* and _*y* are estimated using maximum likelihood. Some practical applications of GNB include text prediction, document classification and spam altering. It requires a relatively small amount of training data to estimate the necessary parameters, can be quite fast in comparison to more complex methods and is easy to implement, being often used as a baseline [21]. However, while its naive assumptions can make such efficiency possible, they can also adversely affect the quality of the results in several real world applications, such as the use case at hand, in which the feature pairs are unlikely to be independent.

K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a type of instance-basedlearning algorithm, meaning it does not construct a general internal model, but instead stores instances of the training data with computation being deferred until classifcation. Over the years it has seen several applications in both statistical estimation and pattern recognition including for instance the classifcation of heart disease to provide a decision support system for clinicians [22]. Conceptually, such an approach can be carried over to the use case at hand, as we are effectively attempting to identify a condition in the cars, and furthermore, being one of the simplest ML algorithms for classi_cation it is at least a good candidate to serve as a baseline.

For KNN, the input consists in the *k* closest training examples in the feature space, with the output being a class membership attributed by a simple majority vote of the nearest neighbours based on some distance metric such as the Euclidean distance.

XGBoost

XGBoost [17] stands for eXtreme Gradient Boosting andis an optimized implementation of gradient boosted trees, designed to be highly efficient and fexible. It is a nonlinear algorithm which typically works well with numerical features and requires relatively less feature engineering and hyperparameter tuning to yield good results. Generally, such methods can be prone to overwriting, as they constantly involve fitting a model on the gradient. To mitigate this, one can optimize for the number of trees until the out of sample error starts increasing once more. XGBoost models are frequently used to solve Kaggle challenges across several domains, with real world applications including for instance the identification of complex relationships between variables for rare failure prediction in manufacturing processes.

Random Forest

In the context of classifcation problems, RF is an ensemble learning method that operates by constructing several decision trees at training time and outputting the class that is the mode of the classes of the individual trees. While a single decision tree can easily run into overwriting problems, being also sensitive to small variations in the data, due to their nature RFs are more robust to such challenges.

Support vector machine

The SVM algorithm constructs hyperplanes in infinite dimensional spaces to classify data into distinct classes. One can consider a good separation to be achieved by the hyperplane with the largest distance to the nearest training-data point of any class (functional margin), as typically larger margins correspond to a lower generalization error. While this is a fairly formal approach to the classification problem, one disadvantage mentioned in the scikit-learn documentation for the *SVC* implementation is that time complexity is more than quadratic with the number of samples, making it hard to scale for data sets with more than a couple of 10000 samples. While this is not the case for this particular case study, it is something to keep in mind when comparing to other approaches.

6.4 Tools and Technology Used

Scikit-Learn

Scikit-Learn is an open-source machine learning package. It is a unified platform as it is used for multiple purposes. It assists in regression, clustering, classification, dimensionality reduction, and preprocessing. Scikit-Learn is built on top of the three main Python libraries viz. NumPy, Matplotlib, and SciPy. Along with this, it will also help you with testing as well as training your models.

TensorFlow

TensorFlow is an open-source framework that comes in handy for large-scale as well as numerical ML. It is a blender of machine learning as well as neural network models. Moreover, it is also a good friend of Python.The most prominent feature of TensorFlow is, it runs on CPU and GPU as well. Natural language processing, Image classification are the ones who implement this tool**.**

Pytorch

Pytorch is a deep learning framework. It is very fast as well as flexible to use. This is because Pytorch has a good command over the GPU. It is one of the most important tools of machine learning because it is used in the most vital aspects of ML which includes building deep neural networks and tensor calculations. Pytorch is completely based on Python. Along with this, it is the best alternative to NumPy.

RapidMiner

RapidMiner is a piece of good news for the non-programmers. It is a data science platform and has a very amazing interface. RapidMiner is platform-independent as it works on cross-platform operating systems.With the help of this tool, one can use their own data as well as test their own models. Its interface is very user-friendly. You only drag and drop. This is the major reason why it is beneficial for non-programmers as well.

Jupyter Notebook

Jupyter notebook is one of the most widely used machine learning tools among all. It is a very fast processing as well as an efficient platform. Moreover, it supports three languages viz. Julia, R, Python. Thus the name of Jupyter is formed by the combination of these three programming languages. Jupyter Notebook allows the user to store and share the live code in the form of notebooks. One can also access it through a GUI. For example, winpython navigator, anaconda navigator, etc.

Azure machine learning studio

Azure machine learning studio is launched by Microsoft. Just like, Google's Cloud AutoML, this is Microsoft's product which provides machine learning services to the users. Azure machine learning studio is a very easy way to form connections of modules and datasets. Along with this, Azure also aims to provide AI facilities to the user. Just like TensorFlow, it also works on CPU and GPU.

MLLIB

Like Mahout, MLLIB is also a product of Apache Spark. It is used for regression, feature extraction, classification, filtering, etc. It also often called Spark MLLIB. MLLIB comes with very good speed as well as efficiency.

Pylearn2

Pylearn2 is a machine learning library that is built on top of Theano. Therefore, there are many functions that are similar between them. Along with this, it can perform math calculations. Pylearn2 is also capable of running on the CPU and GPU as well. Before getting to Pylearn2, you must be familiar with Theano.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

The present work attempted to use the SD approach to model the real-world multistage assembly line. We developed the data-driven continuous flow model using continuous flow approximation for assembly work transfers. The part movement is treated as fluid flow, buffer stocks are water tanks, the conveyor belt is a water pipe, and manufacturing stations are the valves that control the rates of flow. Machine states, i.e. breakdown, starved, and blocked, are treated as discrete events that cause variations in the material flow rate. The statistics of throughput rates match closely with those from previous DES implementations and real-world historical results. It is shown that the proposed continuous flow model can capture the instantaneous (dynamic) throughput rate in a faster manner, and also closely match the DES output during the steady-state. The continuous flow model offers flexibility in choosing the initial states and the actual breakdown and repair times. The model for most cases reached a steady-state within 100 min of simulation time during different initial conditions. The mean throughput values for the presented continuous flow model were within 5% of those from the real world assembly historical data. Future work includes improving computational efficiency, performing characterizations to yield appropriate prediction model structures, implementing real-time prediction routines, and using continuous flow models for performance estimation in the real-world operation of the multistage assembly line.

The analysis of this MMP is particularly challenging due to the amount of variability introduced by the human operators in the loop, responsible for the alignment and inspection of the assembled cars. However, the results suggest that there are certain dimensional variations in the early stages (even those within specification) that can be used to predict deviations at the end of the line regardless of these interventions, indicating that some of these feature interactions are considerably hard to detect without the assistance of more complex data analytics approaches like the one being proposed. While domain expert knowledge is critical for the correct assessment of the corrective actions that need to be carried out during the assembly operations (i.e. offsetting the jig), such an approach can provide further insights to enable an earlier intervention in the framing stages to prevent the propagation of the defects downstream, as well as a quicker identificationof problematic cars for the final assembly.

We showed that non-linear algorithms like XGBoost and RFs are capable of detecting the complex relationships encompassed in this multivariate data set, providing quality estimations with a high capacity to distinguish between OK and NOK cars in an automotive multistage assembly process with high recall.We validated this results on two different test sets, one pertaining to the original data set and the other containing samples collected over the course of the 3 months following the last sample from the original data set. On both we show that the selected models are capable achieving high performance across all the evaluation metrics considered in this study, namely accuracy, recall, precision, F1 score and AUC. Limitations and possible obstacles to the long term success of the predictive approach presented in this work were also discussed, more concretely in regards to the detection of concept drift, with possible solutions and venues for future research having been proposed in Section V. Overall we consider that the approach shows real potential in contributing towards the improvement of existing quality control strategies,with results hinting that reliable predictions can be provided to assist domain knowledge experts in making informed decisions towards the mitigation of defect propagation in multistage assembly scenarios.

REFERENCES

[1] Y. Ding, D. Ceglarek, and J. Shi, ``Fault diagnosis of multistage manufacturing processes by using state space approach," *J. Manuf. Sci. Eng.*, vol. 124, no. 2, pp. 313_322, 2002.

[2] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster, *Recommendations for implementing strategic initiative INDUSTRIE 4.0: Securing future German Manuf. industry; _nal Rep. Industrie 4.0 Work. Group*. Forschungsunion, Berlin, Germany, 2013.

[3] J. Lee, E. Lapira, B. Bagheri, and H.-A. Kao, ``Recent advances and trends in predictive manufacturing systems in big data environment," *Manuf. Lett.*, vol. 1, no. 1, pp. 38_41, Oct. 2013.

[4] D. Djurdjanovic, J. Lee, and J. Ni, ``Watchdog agent-an infotronics-based prognostics approach for product performance degradation assessment and prediction," *Adv. Eng. Informat.*, vol. 17, nos. 3_4, pp. 109_125, 2003.

[5] J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao, ``Intelligent prognostics tools and e-maintenance," *Comput. Ind.*, vol. 57, no. 6, pp. 476_489, Aug. 2006.

[6] R. S. Peres, A. D. Rocha, P. Leitao, and J. Barata, ``IDARTS_Towards intelligent data analysis and real-time supervision for industry 4.0," *Com-put. Ind.*, vol. 101, pp. 138_146, Oct. 2018.

[7] 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, Jan. 2016.

[8] G. Köksal, I. Batmaz, and M. C. Testik, ``A review of data mining applications for quality improvement in manufacturing industry," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13448_13467, Sep. 2011.

[9] D.-S. Kwak and K.-J. Kim, ``A data mining approach considering missing values for the optimization of semiconductor-manufacturing processes," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2590_2596, Feb. 2012.

[10] D. Kim, P. Kang, S. Cho, H.-J. Lee, and S. Doh, ``Machine learning-based novelty detection for faulty wafer detection in semiconductor manufacturing," *Expert Syst. Appl.*, vol. 39, no. 4, pp. 4075_4083, Mar. 2012.