This is a peer-reviewed, accepted author manuscript of the following research article: Dalalah, D, Ojiako, U, Alkhaledi, KA & Marshall, A 2023, 'A support vector machine model for due date assignment in manufacturing operations', *Journal of Industrial and Production Engineering*, vol. 40, no. 1, pp. 68-85. https://doi.org/10.1080/21681015.2022.2059791

A Support Vector Machine model for due date assignment in manufacturing operations

Doraid Dalalah
College of Engineering, University of Sharjah, United Arab Emirates
Industrial Engineering department, Jordan University of Science and Technology
Email: ddalalah@sharjah.ac.ae

Udechukwu Ojiako
College of Engineering, University of Sharjah, United Arab Emirates
Hull University Business School, University of Hull, United Kingdom
UNIZIK Business School, Nnamdi Azikiwe University, Nigeria
Email: udechukwu.ojiako@outlook.com

Khaled A. Alkhaledi
College of Engineering and Petroleum, Kuwait University, Kuwait
Email: kalkhaledi@gmail.com

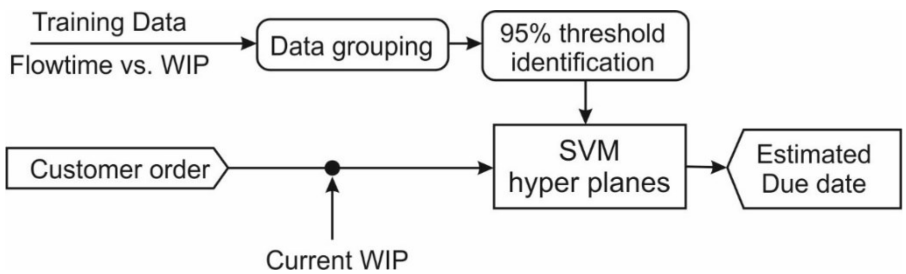
Alasdair Marshall
Southampton Business School, University of Southampton, United Kingdom
Email: A.Marshall@soton.ac.uk

Abstract

The relationship between product flow times and manufacturing system status is complex. This limits use of simple analytical functions for job shop manufacturing due date assigning, especially when dealing with orders involving multiple resource manufacturing systems in receipt of random orders of different process plans. Our approach involves developing a Support Vector Machine classifier to articulate job shop manufacturing due date assigning in heterogeneous manufacturing environments. The emergent model allows not only for the complex relationships between flowtimes and manufacturing system status, but also for the prediction of random order flowtime of manufacturing systems with multiple resources. Our findings also suggest that service levels play a major role in negotiated due dates and eventual customer propensity to place manufacturing orders. In emphasizing negotiated due dates as against exogenous assigned due dates, the study focuses scholarly attention towards the need for participative, open and inclusive due date assignments.

Keywords: Due date; Job shop; Support vector machine; Optimization; Kernel function; Flow time.

Graphical abstract



1. Introduction

In today's manufacturing operations environment, many organizations are transitioning from single silo-based environments to ‘heterogeneous production environments’; in order words, to diverse manufacturing operations and process environments (Saha et al., 2016; Yang and Han, 2021) characterized by very high operational efficiency (Tseng et al. 2021). In these environments, customers are increasingly demanding highly personalized products and services. This tendency has given rise to ‘mass personalization’, where firms ever more routinely seek to exploit technology to configure and deliver products and services that meet the peculiar needs of each *individual* customer. Most importantly, the literature suggests that meeting the needs of each individual customer should not mean compromising on design, functionality and quality (Wang et al., 2017). However, manufacturers of such personalized products and services do not only have to cater only for highly bespoke customer design and functionality requirements; rather, part of such bespoke offerings should also be the ability to fulfil lead and minimum delivery times in a reliable manner (Gershwin, 2018). The reality, however, is that fulfilling customer needs and delivery times can be highly problematic. Manufacturing firms often have to contend with multiple and conflicting product and service functionality criteria (Yan et al., 2015), and with the stochastic nature of the job shop manufacturing process. Thus, setting reliable due dates can be challenging (Kaminsky and Kaya, 2009). Furthermore, personalization entails focusing on individual customers and taking into consideration both explicit and implicit needs during product and service design (Tseng et al., 2010).

Due date assignment in job shop manufacturing is an essential element of production planning and control. Due to the (i) heterogeneity (stochastic) nature of the job shop manufacturing process (Saha et al., 2016), (ii) very diverse product and service mixes (Yan et al., 2015), (iii) fluid order arrival and processing times and (iv) alternative processing routes and also reliability consequences, assigning actual due dates for manufacturing orders becomes challenging. Due date assignment (Gordon et al., 2002; Kianpour et al., 2021) and scheduling problems therefore continue to interest scholars, as shown in various review articles on the subject (Lohmer and Lasch, 2020). Some of this research has addressed the machine scheduling of jobs in order to determine due dates. However, much of this research focuses on single machine scheduling (Zhao et al., 2014). Notably, when multiple machines in complex manufacturing systems do exist, scheduling, time allocations and due date assignments become more difficult. Generally, as flow times in manufacturing systems exhibit high positive correlations with work-in-process (*WIP*), during the forecasting of due dates, it is nonetheless feasible to employ time series analysis as a means of predicting the flowtimes of newly arriving orders in real time (Wang and Guo, 2010). Yet the relationship between flowtimes and the manufacturing system status is complex. Generally, the system status will be described by the current work-in-process (*WIP*). The complexity of the system status limits the use of simple analytical functions to articulate this relationship. To counter for the limitation of such simple description, more robust models that employ appropriate level of analytical functions are desirable.

We opine that one such model, which may be best suited for use in the prediction of flowtime of random orders of different process plans emanating from manufacturing systems with multiple resources, is the Support Vector Machine (SVM) classification model. The SVM is a mathematical optimization approach that classifies data into separate groups by analytically describing the separation hyper-plane (Guenther and Schonlau, 2016). Herein, our study aims to develop a SVM classifier to aid in establishing the flow times, and hence the due dates, at a desirable service level. An optimization model will be developed as a solution for the classifier. Intensive simulation experiments will be conducted for the purpose of training, testing and validating the SVM model.

In production planning and scheduling, conventional approaches concentrate on allocating demand to a factory, then obtaining sequences of operations on machines in this factory, which aids in finding the orders' completion dates. However, due to heterogeneity in customer demands and the uncertainty that exists in job shops due to randomness in order arrivals, processing times and breakdowns, quoting due dates for order completion becomes more challenging. To co-improve service levels, due date compliance and customer satisfaction, more precise dates should be delivered. The proposed model here contributes to the literature by hybridizing both optimization and statistical analysis to overcome the uncertainty in the due dates, by providing 95% service levels. Under the settings of our model, the quoted due dates are optimized and statistically drawn from a 95% confidence level which should present more reliable and precise delivery dates.

To address the study objective, the rest of the paper proceeds as follows. In the next section (section 2), we present a brief overview of the literature on due date assignment services. After articulating our study approach in section 3, in section 4 we present the voluntary SVM model. To demonstrate the value of the presented SVM modelling approach, in Section 5 we undertake a number of different experiments to test the operation and the results of the model. Data validation is undertaken in section 6 while the paper concludes in Section 7.

2. The literature

2.1 Due date assignment

With origins in '*just-in-time*' production and manufacturing philosophy, one approach which manufacturing organizations have widely adopted for mass personalized production planning and control is *workload control* (Sagawa and Land, 2018). The *just-in-time* production and manufacturing philosophy focuses on ensuring that exact amounts of materials required for production activities are available if and when needed, while at the same time maintaining the lowest required levels of inventories (McLachlin, 1997). Conversely, the workload control concept works on the basis that the rate of job inputs will equal that of outputs (Thürer et al., 2015). The extant literature (specifically, Bertrand and Wortmann, 1981) had suggested that in manufacturing job shops, the control functions consisted of three dimensions. These were (i) input control, (ii) output control and (iii) due date setting or assignment.

The essence of the due date assignment philosophy is that designated manufacturing activities (jobs) are to be completed on a single designated date ('*due date*') (Gordon et al., 2002) or within a specific time window interval ('*due window*') (Janiak et al., 2015). A major underlying assumption of due assignment literature is that at the beginning of the planning horizon, jobs to be assigned *due dates* or *due windows* are known. Due date assignment then commences in either an endogenous or exogenous form. An endogenous due assignment is one usually set *in*ternally by the manufacturing job shop. Such due date assignment will usually take into account information on (i) jobs that are still to be taken in, (ii) those that are already within the manufacturing system (Work-In-Process), (iii) other forthcoming jobs and (iv) other considerations such as congestion and workflows. An exogenous due date assignment is one usually set *ex*ternally to the manufacturing department. In effect, these are imposed from outside the manufacturing job shop by non-manufacturing teams within the organization. Exogenous due assignments will *not* take into account information on jobs that are still to be taken in, those that are already within the manufacturing system, other forthcoming jobs and other considerations such as congestion and workflows.

The literature appears to suggest that manufacturing firms and customers have different preferences with respect to assigned *due dates* or *due windows* (Thürer et al., 2019). Manufacturing firms, for example, find it highly desirable to be able to complete job assignments within a '*due window'* that has either emerged from (i) projected (forecasted) estimates of job flow times (Sridharan and Li, 2008), (ii) scheduling (Prabagaran and Joseph, 2019) or (iii) contractual negotiations and agreements (Shabtay, 2016). This is because a '*due window'* allows manufacturers to deliver their products within a time span; thus, with some tolerance of slack and associated imprecision in delivery dates. However, customers appear to consistently prefer working to a '*due date'*. This is because customers tend to place orders in several batches. Thus, instead of placing several different orders and requesting several different delivery dates, customers will place several orders and negotiate a specific date by which all the orders will be dispatched (Rasti-Barzoki and Hejazi, 2013).

Despite the preference for *due windows for the perspective of manufacturers,* there are a number of clearly articulated advantages in specifying '*due dates'* for both manufacturing firms and customers (even though it is not an instinctive manufacturer preference). The extant literature suggests, for example, that specifying a '*due date'* associates to discernable advantages in production scheduling and predictability, and ultimately, to manufacturer costs management (-Barzoki and Hejazi, 2013). Quoting *'due dates'* is essential for manufacturing firms seeking to demonstrate their sense-making to customers in an era of highly personalized product and service development. Indeed, emphasizing lead times and due dates can be vital in allowing manufacturers to negotiate with customers.

There are two perspectives of due date assigning (Shabtay, 2016). One perspective emphasizes speed of delivery. More specifically, this focuses on the shortest lead-time of product and service delivery to the customer. The other emphasizes reliability of delivery (Baykasoğlu et al., 2008). Here, the focus is on consistency in achieving designated due dates.

2.2 Predicting due date assignment

Due date assignment problems continue to attract scholarly interest. Prior studies in this area have mainly focused on the interface between various schedule rules and various methods of due-date. Early studies include that of Reiter (1966) who developed a system for managing job-shop production based on equipment availability. Other studies, focused on different aspects of the due date assignment problem, include that of Blackstone et al. (1982), who undertook a comprehensive review of the literature on dispatching rules in job shop operations. They not only compared various dispatching rules, but also evaluated analytical methods and simulation techniques available for addressing optimal due date determinations. In Sen and Gupta (1984), a review of the literature on static scheduling involving due dates was undertaken, with classifications based on scheduling objectives. Cheng and Gupta (1989), on the other hand, conducted an overview of date assignment in static and dynamic job shop situations. Their findings suggest two major directions of research: one dealing with desirable due date assignment and the other dealing with optimal due date determination. Their study also found that the most common approach to addressing optimal due date determination was through analytical methods and computer simulation. In Baker and Scudder (1990), an overview of the extant literature on sequencing with earliness and tardiness penalties was provided. Their main concern lay with studies focused on uses of scheduling to minimize total earliness and tardiness penalties. The framework that emerged from their study was extended by Koulamas (1994). More recent studies related to due date assignment problems include Gordon et al. (2002) and Lohmer and Lasch (2020). Gordon et al. (2002), for example, provided a comprehensive review of literature on due date assignment and scheduling problems, which led them to articulate a unified deterministic framework. Lohmer and Lasch (2020), on the other hand, focused their review on scheduling in heterogeneous manufacturing environments.

Due to uncertainty, the due date assigned in job shop manufacturing may be either *over* or *under*-estimated. Forecasting is therefore important in due date assignment (Gansterer, 2015). For the manufacturer, completion of contracted orders prior to the due date will lead to unnecessary inventory costs, as it will (or may) not be possible to dispatch ordered products and services to the customer earlier than the date agreed with the customer (Schonberger, 2019). Conversely, late completion following delays will result in orders becoming 'tardy' (Baykasoğlu et al. 2008). Cheng (1988) articulates this dilemma very eloquently by stating that "…*completing a job early means to bear the costs of holding unnecessary inventories, while finishing a job late results in contractual penalty and loss of customer goodwill*". In sum, accurate due dates are important in that they serve to facilitate the avoidance of both earliness (completion of job shop manufacturing before due date), tardiness (delay in the commencement of job shop manufacturing) and resulting unintended consequences.

The reality is that both early completion and late completion can lead to declining customer trust. Thus, in the absence of optimized forecasting, manufacturers will inevitably seek to retain substantial product stocks in anticipation of demand. This approach allows for immediate shipment of customer orders once received (Kim, 2018). However, this may strain inventory management, as considered especially from the perspective of lean management where ever-increasing inventories held by organizations in anticipation of future customer orders is taken as indicating failure to apply lean principles (Schonberger, 2019).

2.3 Service and due dates

'Service' is a concept that appears regularly in manufacturing operations literature (Ojiako et al., 2013; Dalalah and Araidah, 2010). Construed as multi-dimensional in nature (Voss, 2003), it implies receiving a set or series of processes (Frei and Harker 1999) and actions (Johns 1999), which the recipient would have expended considerable effort to obtain. In the context of due dates, service levels will refer to the percentage of orders that the manufacturer is confident will be satisfied on the pre-agreed due dates (Hopp and Sturgis, 2000). In effect, service levels represent the guarantees made by the job shop manufacturer that they can meet their due date commitments to customers with certainty (Graves and Willems, 2003). While a 'perfect' 100% service level, which implies being able to meet -without failure- *all* customer orders on the designated due dates is desirable, due to the inherent (aleatory) uncertainty associated with supply chains, manufacturing firms are generally unable to meet such service level aspirations (Yin et al., 2012). Thus, setting a 'perfect' 100% service level is not realistic, and so different scholars have examined what service levels are feasible within the due date context. Hopp and Sturgis (2000) for example, designate service levels between 90% and 95% as being '*higher service levels'*. Radasanu (2016), on the other hand, identified three grades of service levels as (i) '*high service level'* – between 96 and 98%, (ii) '*medium service level'* – between 91 and 95% and (iii) '*lower service level'* – between 85 and 90%. In our study, we assume the 95% service level as our desired service target. There are two grounds for this assumption. First, the 95% service level appears to be the standard generally desired in most job shop manufacturing literature (Hopp and Sturgis, 2000). Second, this level of service appears more attainable (Graves and Willems, 2003).

3. Proposed approach

Consider a job shop manufacturing system consisting of a set of R resources. The system admits orders that randomly arrive. Each order is to be performed on a subset of resources with specific routing. To replicate a real manufacturing system, orders may share the resources according to their process plans. Upon order arrival, due dates are drawn from estimated production flowtimes which are the core of this article, and then agreed with the customer (preferably, endogenously) after a period of negotiation. Once the order is completed, the manufacturer compares the achieved due date with the due date that was initially agreed with the customer. This is undertaken with due consideration to a desired level of service. For this reason, the service level agreement plays a considerable part in estimating flowtimes (FT). Figure 1 demonstrates the concept, where orders with specific manufacturing route requirement arrive to a set of resources. Upon the arrival of an order, the arrival time is recorded and an estimated delivery date is negotiated with the customers. When an order is complete, the goodness of the estimated delivery date can be compared with the actual delivery date.

As earlier alluded to, the job shop manufacturing process is stochastic in nature. This makes it possible that two or more orders arrive at the same time but exhibit different flowtimes. Thus, there is a need for descriptive statistics to articulate flowtime value.

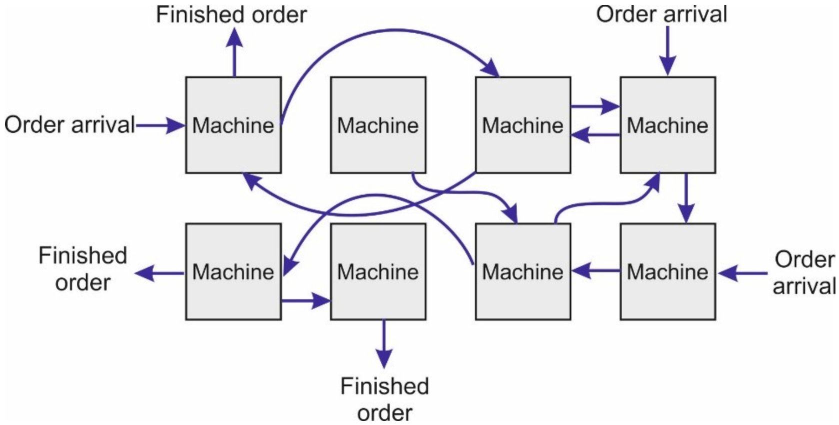


Figure 1: Orders arriving at a shop at different times and rates.

Manufacturer specifications of certain flowtimes (which implies a guaranteed zero error in estimating flow times) can have an adverse effect on manufacturer-customer relationships. This can, for example, negatively impact customer trust and goodwill. If, for example, customer due dates are set to the median value, this will entail that 50% of orders are expected to exceed such value, yielding dissatisfied customers. The same conclusion will likely hold if an average value is used.

Consider for instance, a shop floor consisting of three machines, the system status at the moment an order i arrives will be the size of total work-in-process at each machine which will be represented as WIP_i=[wip_i^1, wip_i^2, wip_i^3]. If we divide the WIP at each machine into intervals, say 3, what emerges will be a finite number of system states (i.e., a form of higher dimensional matrix of 27 elements in this instance). As our study proceeds on the basis of a desired service level target of 95% (to be discussed in the next section), by having the appropriate number of data points collected for each state, we are able to compute the necessary descriptive statistics that are able to establish the 95th percentile of each state's flowtime. Under such conditions, orders exhibiting a flowtime longer than (≥) 95% will be deemed to have not met the 95% service level. On the other hand, flowtimes shorter than (≤) 95% would have met the 95% service level. This classification yields two clusters of flowtime data points. The best hyper plane that splits the two clusters can be used to predict the 95% service level of flowtime. Upon the classification of the data points, we will employ SVM to establish the separating hyper plane function. The separation hyper plane can then be subsequently utilized to predict the flowtime of the arriving orders. To tailor SVM for the purpose of estimating flowtime, we will record the observed due dates for each identified system status and then statistically establish the 95% service threshold. This enables us to classify the system points against those (i) which are able to fulfill the service level and (ii) those which are unable to do so. By clustering, we must emphasize that there will be a 95% flowtime level for each data group.

Our approach to experimentation will involve dividing the *FT* data into two levels: (i) those above the 95% service level and (ii) those below the 95% service level. In addition, since the 95th percentile of the flowtimes can hardly be estimated for discrete *WIP* values, our designed experiment will also split the *WIP* into intervals.

As a means of demonstration, consider a simple manufacturing system comprising 3 resources. It is most likely that the flowtime will vary according to the current *WIP*. However, this variation will encompass a positive correlation. The reality, however, is that identifying the 95% percentile is a complicated task because threshold values change according to *WIP* levels. For these reasons, it is prudent to divide the *WIP* into different intervals, such that with each interval, the 95% flowtime limit is statistically computed. In Figure 2, we show the 3 resources and 3 different *WIP* intervals resulting in 9 different data groups.

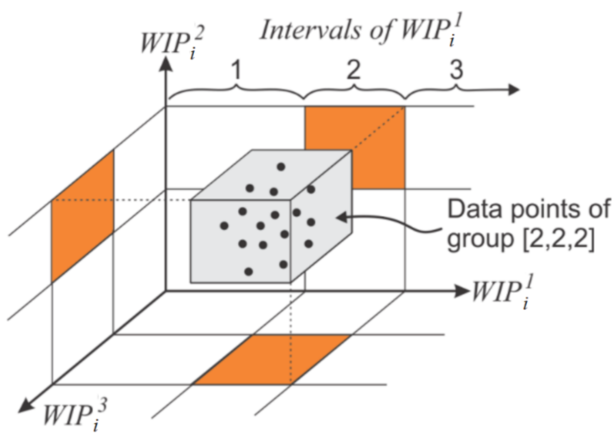


Figure 2: Three resources along with their *WIP* intervals resulting in separate groups (clusters) for order *i*.

It is observed that the number of data groups will increase exponentially by increasing the number of resources or the number of intervals of the *WIP* for each resource.

In Figure 3, we show a demonstration of the flowtime classes. Given the training data set, the WIP of each resource is divided into bins (intervals). The data points within each bin are statistically divided into two classes; those are above the 95% level of the flowtime are classified as "+1" and those below the 95% as "-1". We note that the 95% hyper plane is not necessarily a straight line as the relationship between the *WIP* and flowtime is not linear. For this reason, the *WIP* intervals are recommended to be smaller at higher *WIP* values. In fact, a higher number of intervals will result in a higher number of data clusters, which leads to enhanced predictability and precise estimates. Figure 3-*b* shows smaller bin size, and Figure 3-*c* demonstrates how smaller bins result in precise separating line. Due to less variability in flowtimes at small *WIP* values, larger bins sizes may not significantly affect the resulting separating line.

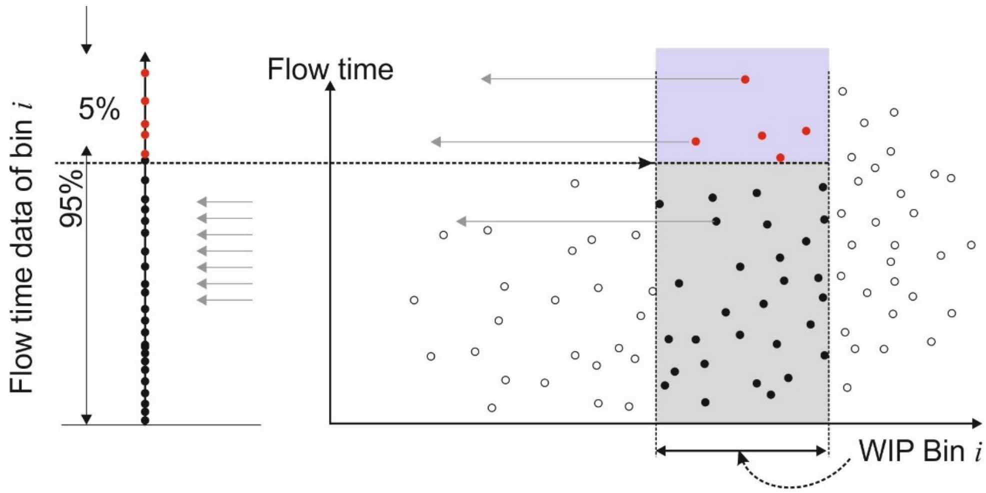


Figure 3-*a*: Classifying data points of one *WIP* interval of one resource.

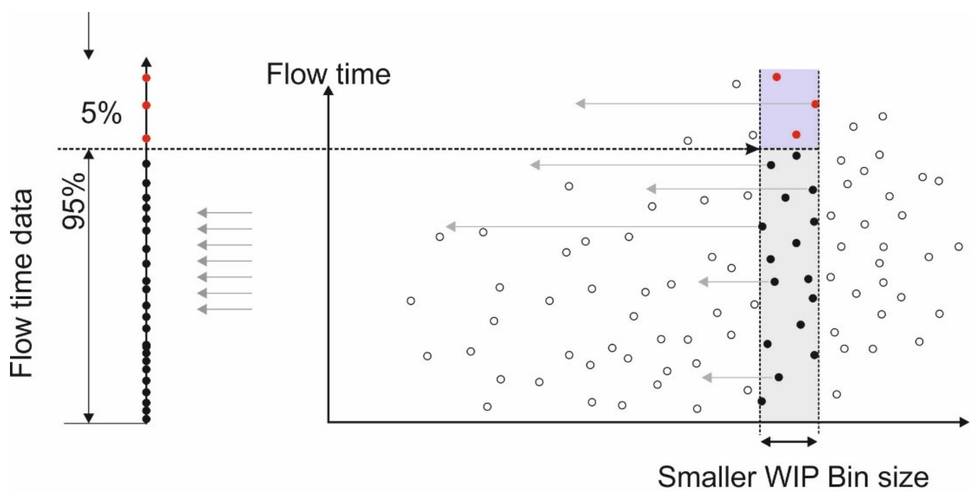


Figure 3-*b*: Classification the data points of smaller *WIP* interval of one resource.

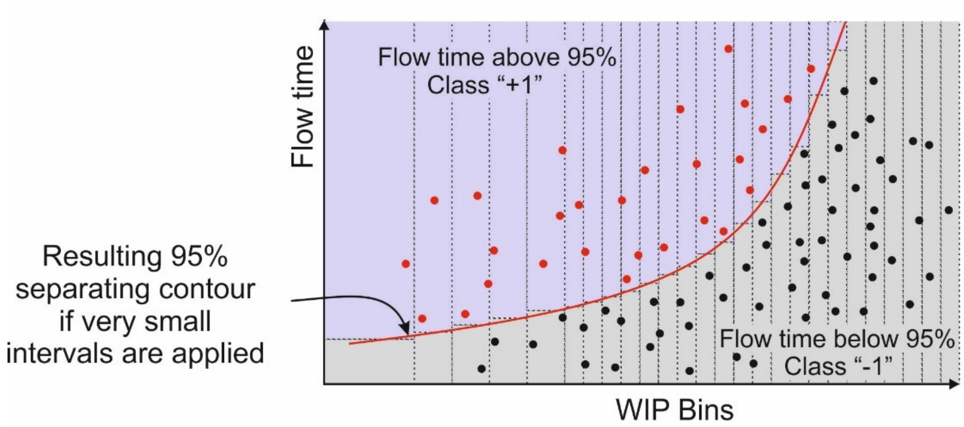


Figure 3-*c*: Flowtime classification using small bin sizes of *WIP* of a single resource.

Figure 3 also shows that the 95% percentile varies according to the *WIP*. Thus to establish the 95%, it is necessary to divide the *WIP* into intervals. In effect, the regions have to be as small as possible. However, it must be noted that by having smaller regions more data points are required. The demand for more data points serves as a means of ensuring that each region includes the necessary number of data points. When the due date is assigned based on a 95% threshold, it provides assurance that up to 95 % of *FT* values are within the threshold limit. In fact, the line that splits the top 5% from the 95% is a hyper plane that can be obtained by employing SVM. Once the hyper plane that separates the *WIP* data points is established, we can employ this plane to estimate the minimum *FT* that provides a 95% service level. We then set up SVM by capturing the training data points and their classes. One major advantage of SVM is the ability to filter, and hence, reduce, the set of data points for faster classification (Meyer et al., 2003). This process is explained in more detail in the next section, where the separating hyper plane depends on the closest set of points to the separating hyper plane between the classes (which we term, the relevant points). Hence, only a proportion of the data below the 95% will be relevant, while the remaining points do not affect the separating hyper plane. In Figure 4, we show the relevant data points.

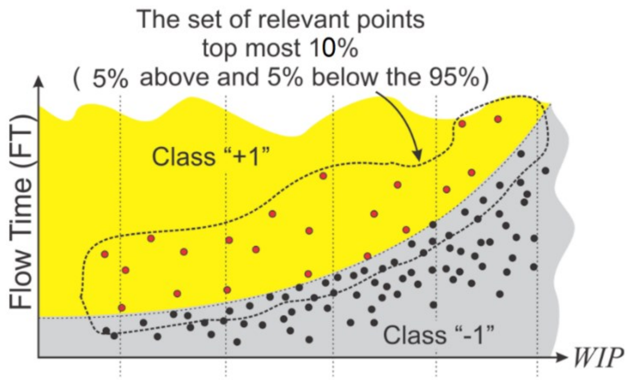


Figure 4: The set of relevant data points

4. Modelling

In this section, we walk through the main steps for the application of SVM to achieve the research objectives as follows: (i) Articulating SVM model, (ii) Data points transformation through the Kernel Trick, (iii) Work-in-Process and Data Reduction and (iv) Predicting the *FT* at a 95% Service Level.

4.1 Flowtime SVM Model

Support Vector Machine (SVM) is a clustering technique employed to establish the best separating hyper planes via dual optimization (Meyer et al., 2003). More specifically, it is a classification approach that can be set up utilizing training data points allowing for historical data points to be collected and used for training and tuning of model parameters. As a means of illustration, consider two classes of historical data points as shown in Figure 5. It will be possible to establish many separating lines to divide the data points into two classes. However, from the perspective of SVM, the best separating line (the hyper plane for higher dimensions) is the mid-line of the widest strip that separates the two classes. This strip can be found by maximizing the distance *D* between the two edges resembling the strip borders. This strip represents the maximum gap between data points of the two classes.

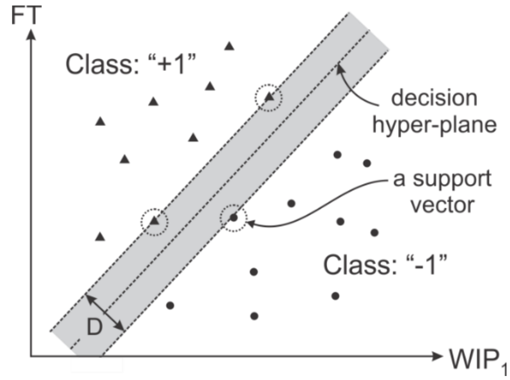


Figure 5: *WIP-FT* sketch with the support vectors circled

The distance *D* is termed the margin of the classifier. The decision surface is merely determined by a few set of points, termed Support Vectors. The remaining data points have no effect on determining the decision hyper-plane. Similar separate strip can be applied for nonlinear case as shown in Figure 6. The wider the gap, the better the classifier. In fact, the entire SVM concept is underpinned by a need to maximize such gaps in order to establish the analytical classifier function.

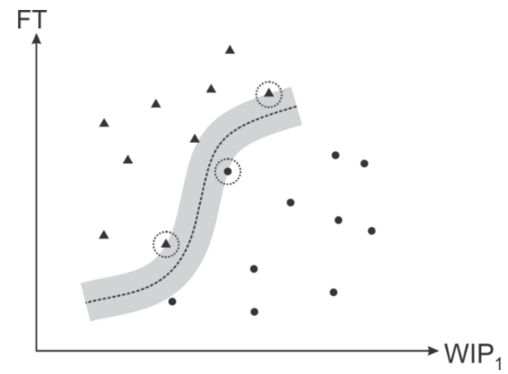


Figure 6: Support vectors circled on a *FT-WIP* sketch of nonlinear separator

Let the manufacturing system status when order *i* arrives be characterized by the vector WIP_i where WIP_i=[wip_i^1 wip_i^2 ... wip_i^R] which encompasses the *WIP* of all resources. The actual flowtime of order *i* upon completion of the job is denoted by FT_i where such an actual flowtime will not be available until the job is completed, i.e., the data collection phase. To find the actual flowtimes in the data collection phase, the arrival times of the arriving orders are recorded. Upon completion of the orders, the departure times are also recorded. The difference between the departure and arrival times is the actual flowtime which is used for model training. Accordingly, the training data set denoted by ***WF*** (work and flowtime matrix) can be given by:

$$\boldsymbol{WF} = \begin{bmatrix} \boldsymbol{WF_1} \\ \vdots \\ \boldsymbol{WF_N} \end{bmatrix} = \begin{bmatrix} wip_1^1 & \dots & wip_1^R & FT_1 \\ \vdots & \ddots & \vdots & \vdots \\ wip_N^1 & \dots & wip_N^R & FT_N \end{bmatrix}$$

where $\boldsymbol{WF_i} = [wip_i^1 \cdots wip_i^R \quad FT_i]$ is the work-flowtime vector of of i and N is the number of orders.

Suppose that each data point is classified by either "1" and "-1" as illustrated earlier, we can construct and train a classifier function to classify the entire set of data points into those of a flowtime above the 95% (+1) and those below the 95% (-1). Let the class of each point in $\boldsymbol{WF}$ be denoted by y_i, $\forall i = 1,\ldots, N$, where y_i = "-1" or "1", our model will implement *SVM* to establish the best hyper-plane that separates the two classes. The classifier function *f* which is in essence a function resulting by training the SVM using the $\boldsymbol{WF}$ matrix will classify the data points. Hence given a certain *WIP* value, the 95% *FT* can be estimated for such *WIP* level.

Note that our data points are hardly separable, which necessitates the need for higher degree classifying function. For simplicity, let us consider linearly separable data first. Figure 7 shows data which is separable by a unique linear separator.

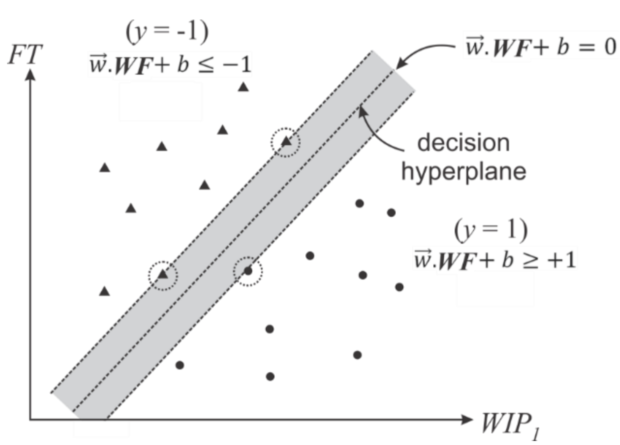


Figure 7: The hyper-plane maximizing the gap between the two classes of a single resource production system

The training data set is represented by the form: $\{\boldsymbol{WF}_i, y_i\}$, $\boldsymbol{WF}_i \in \Re^N$, $y_i \in \{-1,1\}$, $\forall\ i = 1,\ldots, N$. For a linearly separable data, a hyper-plane can be described by $\vec{w}.\boldsymbol{WF} + b = 0$, where $\vec{w}$ is a coefficient vector of n elements (n = 2 for binary classification) and b is an intercept. The boundaries of the separating strip are given by:

$$\left.\begin{array}{l}\vec{w}.\boldsymbol{WF}+b=-1\\ \vec{w}.\boldsymbol{WF}+b=+1\end{array}\right\} \tag{1}$$

and the distance between the above parallel border lines is given by $D=\frac{|b_1-b_2|}{\|\vec{w}\|}$, where the b_1 and b_2 are two parallel plane constants, hence,

$$D=\frac{2}{\|\vec{w}\|} \tag{2}$$

As we aim to maximize the gap between the two classes of (0-to-95%) and (95-to-100%), we will minimize $\|\vec{w}\|$, which is also equivalent to minimizing the quantity $\frac{1}{2}\|\vec{w}\|^2$, where the fraction is added to simplify subsequent analysis later in the dual formulation. This objective function is subject to correctly predicting the data points, that is:

$$\left.\begin{array}{lll}\vec{w}.\boldsymbol{WF}_i+b\leq -1 & if & y_i=-1\\ \vec{w}.\boldsymbol{WF}_i+b\geq \ \ 1 & if & y_i=+1\end{array}\right\} \tag{3}$$

Casting the above into a single constraint, results in:

$$y_i(\vec{w}.\boldsymbol{WF}_i+b)\geq \ \ 1,\ \forall i=1,\dots,N \tag{4}$$

For a linearly separable data set, our model can be summarized in a primal form as follows:

$$\left.\begin{array}{c}\min\frac{1}{2}\|\vec{w}\|^2\\ s.t.\\ y_i(\vec{w}.\boldsymbol{WF}_i+b)\geq \ \ 1 \quad \forall\, i=1,\dots,N\end{array}\right\} \tag{5}$$

The objective function is equivalent to $\frac{1}{2}\sum_{i=1}^{n}w_i^2$. Now, given a new instance $\boldsymbol{z}$, the classifier function will predict the class of this data point as "-1" if $\vec{w}.\boldsymbol{z}+b\leq 0$ and as "1" if $\vec{w}.\boldsymbol{z}+b\geq 0$, alternatively, this can be represented as $f(\boldsymbol{z})=sign(\vec{w}.\boldsymbol{z}+b)$. The primal optimization model shown in *Equation 5* has two decision variables, particularly, the components of $\vec{w}$ and the scalar b. The model works well when the data is linearly separable. Although the primal model can be solved by any standard optimization solver, we would like to present the dual problem to prepare the model for the case of inseparable data. The model can be recast in a dual form yielding a quadratic problem of N dual variables, i.e., $\boldsymbol{\alpha}=\{\alpha_1,\dots,\alpha_N\}$.

$$\left.\begin{array}{c} \max \quad \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{WF}_i \cdot \boldsymbol{WF}_j \\ s.t. \\ \alpha_i \geq 0 \\ \sum_{i=1}^{N}\alpha_i\, y_i = 0 \end{array}\right\} \tag{6}$$

The data points are presented inside a dot product and the coefficient vector $\vec{w}$ is defined in terms of α_i as:

$$\vec{w} = \sum_{i=1}^{N}\alpha_i\, y_i \boldsymbol{WF}_i \tag{7}$$

The classifier function of any point, that is $\boldsymbol{z}$, then becomes:

$$f(\boldsymbol{z}) = sign\ (\sum_{i=1}^{N}\alpha_i y_i \boldsymbol{WF}_i \cdot \boldsymbol{z} + b) \tag{8}$$

By implementing Lagrange relaxation, we can define the Lagrangian function:

$$\mathcal{L}(\vec{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{n} w_i^2 - \sum_{i=1}^{N}\alpha_i\,(y_i(\vec{w} \cdot \boldsymbol{WF}_i + b) - 1) \tag{9}$$

The Karush-Kuhn-Tucker (KKT) conditions for solving the above Lagrangain can be derived in a straightforward manner. If we set the derivatives with respect to $\vec{w}$ and b to 0, we get:

$$\left.\begin{array}{lcr} \frac{\partial L(\vec{w},b,\alpha)}{\partial b} = 0 & \rightarrow & \sum_{i=1}^{N}\alpha_i\, y_i = 0 \\ \frac{\partial L(\vec{w},b,\alpha)}{\partial \vec{w}} = 0 & \rightarrow & \vec{w} = \sum_{i=1}^{N}\alpha_i\, y_i \boldsymbol{WF}_i = 0 \end{array}\right\} \tag{10}$$

Hence, solving *Equation 10* will yield the analytical optimal solution. However, for large number of data points, optimization solvers will easily find the solution of *Equation 6*.

If the points are inseparable in the case of noisy data or misclassification, we can assign a slack variable ξ_i to each instance in the data. The parameter ξ_i can be thought of as the distance from the data points to the separating hyper-plane as depicted in Figure 8.

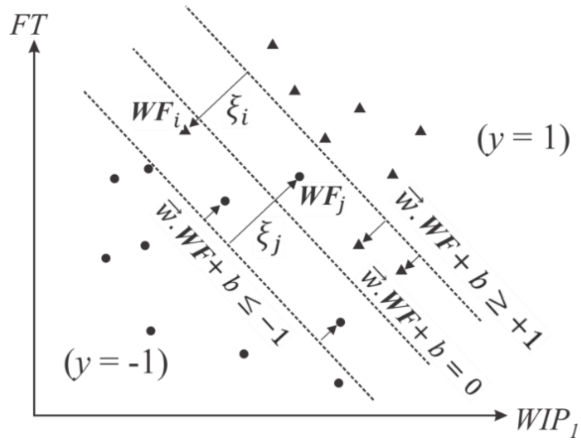


Figure 8: *WIP-FT* diagram with SVM classifier of soft margins

Mathematically, this can be represented in a primal form as:

$$\left.\begin{array}{c} \min \quad \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ s.t. \\ y_i(\vec{w}\cdot \boldsymbol{WF}_i + b) \geq 1 - \xi_i, \forall i = 1, \dots, N \end{array}\right\} \quad (11)$$

where C is a positive tradeoff parameter. The classifier is still represented as $f(\boldsymbol{z}) = sign\ (\vec{\boldsymbol{w}}\cdot \boldsymbol{z} + b)$, with the dual formulations being represented as:

$$\left.\begin{array}{c} \max \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{WF}_i \cdot \boldsymbol{WF}_j \\ s.t. \\ C \geq \alpha_i \geq 0, \forall i = 1, \dots, N \\ \sum_{i=1}^{N}\alpha_i y_i = 0 \end{array}\right\} \quad (12)$$

Large C values call for sharp threshold and hard margins (no misclassified data exists) which is the same as for the optimization problem in *Equation 6*. Smaller values of C mean softer margin is selected, allowing for misclassifications.

4.2 Inseparable Data and the Kernel Trick

In inseparable data, the data points can be transformed into a higher dimensional space named *feature space.* This operation is performed using a transformation function $\varphi(\mathbf{x}_i)$ for a data set $\mathbf{x}$. By so doing, the nonlinear operation in the input space will be equivalent to linear operation in the feature space. Figure 9 shows a transformation instance from nonlinear to linear.

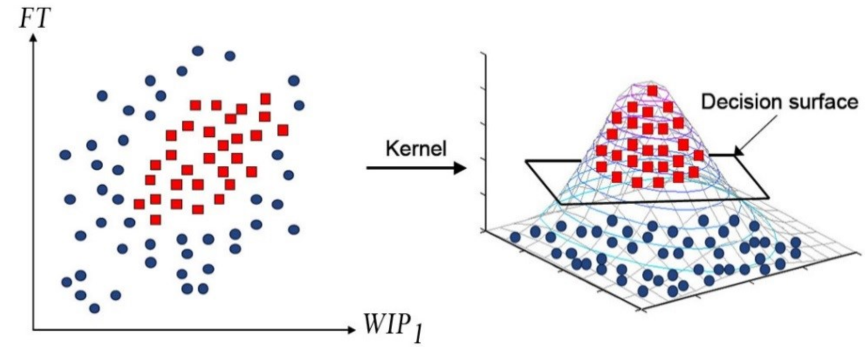


Figure 9: Two linearly inseparable classes of data points marked by squares and circles

Noticeably, the data points appear only in the inner product operation as shown in *Equation 12*; accordingly, as long as the inner product can be computed in the feature space, there is no need for any awareness of the explicit mapping (i.e., we may not be concerned about the function φ, instead we will be interested in the dot product of such function). For data points denoted by $\mathbf{x}_i$, $\forall i = 1, \ldots, N$, a kernel function is defined as:

$$\boldsymbol{K}\,(\mathbf{x}_i, \mathbf{x}_j) = \varphi\,(\mathbf{x}_i)^T \varphi\,(\mathbf{x}_j)$$

This is referred to as the kernel trick. The transformation function $\varphi\,(.)$ is represented by the Eigen functions of the kernel $\boldsymbol{K}\,(\mathbf{x}_i, \mathbf{x}_j)$ (a concept in functional analysis). In general, the exact transformation does not attract attention, instead, the kernel function is the one that is usually specified. The kernel function is simply an inner product representative of a similarity measure between the data objects. It is important to emphasize that the selected kernel has to satisfy Mercer function. Such condition implies that the kernel squared matrix of the (i, j) entries is always positive definite. Many kernel functions do exist, the simplest is the polynomial of degree d which is given by $\boldsymbol{K}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d$. When $d = 1$, this reduces to a straight line separator. Radial basis kernel is given by $\boldsymbol{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/(2\sigma^2))$ where σ is a kernel parameter. Power kernels are given by:

$$\boldsymbol{K}(\mathbf{x}_i, \mathbf{x}_j) = -||\mathbf{x}_i - \mathbf{x}_j||^{\beta} \tag{13}$$

In our model, we decided to implement the power kernel as this was the best choice to present a single separating contour. Other kernels usually fail to do so as they create multiple peaks/valleys and hence, poor clustering for our application. Our preferred option also provides a wide variety of separation hyper-planes which, compared to the radial basis, tends to produce clusters bounded by closed loops. Consequently, by implementing the kernel trick, our model can be represented as:

$$\left.\begin{array}{c}\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1,j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{K}(\boldsymbol{WF}_i, \boldsymbol{WF}_j) \\ C \geq \alpha_i \geq 0 \\ \sum_{i=1}^{N} \alpha_i y_i = 0\end{array}\right\} \quad (14)$$

For a test point *z*, the discriminant function is essentially a weighted sum of the similarity between **z** and the support vectors. The classification function is represented mathematically as:

$$\left.\begin{array}{c}\boldsymbol{f} = \big(w, \varphi(\boldsymbol{z})\big) + b = \sum_{m=1}^{s} \hat{\alpha}_m \hat{y}_m \, \boldsymbol{K}\big(\widehat{\boldsymbol{WF}}_m, \boldsymbol{z}\big) + b \\ \boldsymbol{w} = \sum_{m=1}^{s} \hat{\alpha}_m \hat{y}_m \, \varphi(\widehat{\boldsymbol{WF}}_m)\end{array}\right\} \quad (15)$$

where $\widehat{\boldsymbol{WF}}_m$, $\hat{y}_m$, $\hat{\alpha}_m$ and *s* are the data, the classification, the dual variables associated with the support vectors and the number of support vectors listed in *SV*, respectively. If a point is not a support vector, it is associated α value will be equal to 0. The set *SV* includes only the support vectors of the matrix $\boldsymbol{WF}$.

4.3 Work-in-Process and Data Reduction

We had earlier opined that the 95 percentile differs according to the *WIP* intervals of all resources. The narrower the width of the *WIP* intervals, the more accurate the percentiles. However, this is at the expense of higher number of training data points. Higher number of data points involuntarily entails longer simulation periods to confirm that each group encompasses the sufficient number of data points. For simplicity, in our model, we will divide all the *WIP* into equal number of intervals, say, *m*. Accordingly, for a production system with *R* resources and *m* *WIP* intervals, the outcome will be m^R distinct *WIP* groups. For each group, the necessary statistical analysis will be conducted in order to establish the threshold value of the *FT* at the 95 percentile. The statistical package SPSS will be employed for this. The data points within each group will then be classified into two separate classes: those with a flowtime above the 95% denoted "+1" and those with a flowtime less than 95% which will be denoted "-1". Inevitably, each distinct group will maintain its own specific 95% threshold.

As the separating hyper-plane in *SVM* depends solely on the support vectors which will be within the 90% to 100% margin, there will be no need to consider the entire set of data points in the model. Hence, only the top 10% of the data points will be included in the analysis. Such subset of the data points will constitute the relevant points. Thus, any data point with *FT* value below the 90% threshold of its group will be excluded from the dataset. This data reduction will significantly speed up the *SVM* solution.

4.4 Prediction the FT at a 95% Service Level

Once the SVM model is trained via a set of data points, the classifier function will be ready for use, which is given by *Equation 15*. The classifier function represents the separation hyper-plane at the contour of a "0" value. Given a data point, say $\boldsymbol{WF_i}$, we can insert this point into the classifier in order to establish whether it belongs to the top 5% or below by watching for the classifier sign.

However, the classification is not really what gives this research value in its contribution. Instead, it is the prediction of *FT* at a given *WIP*. The *FT* point that falls on the separating hyper plane provides the 95% service level. Accordingly, given a specific *WIP* and by equating the classifier to "0", we can easily solve for the *FT*. This is accomplished by mathematically solving *FT* in:

$$\boldsymbol{f} = \left(w, \varphi(\boldsymbol{z})\right) + b = 0$$

An alternative mathematical representation will be:

$$\boldsymbol{f} = \left(w, \varphi(\boldsymbol{WF_i})\right) + b = 0 \tag{16}$$

The above equation shows that the sole unknown variable is the *FT* inside $\boldsymbol{WF}$. However, the resulting hyper plane is highly nonlinear, hence solving for *FT* can be attained numerically using a simple algorithm such as Bisection method. Note that the classifier crosses the "0" value at a specific *WIP*, hence, using the Bisection method we can easily track at what *FT* value the classifier crosses the "0" contour. Note that the above model relies on statistical data input to identify the 95% threshold of flowtimes. Hence, given the current situation of *WIP* for the existing resources, the model should demonstrate values that are 95% of the time correct. With such model settings, only 5% of estimates are expected to exceed the predicted flowtime. Finally, the entire model is illustrated in the following diagram; first, a training data has to be available to establish an initial separating hyperplane that will be used to predict the flow time. The data is statistically split into two classes according to the 95th percentile, then the SVM model is built and optimized via an iterative test of the classification error, where the log likelihood is used to indicate the error. Once the SVM model is fine-tuned, arriving orders can attain a quoted flowtime using the SVM model. The data of the new arrive orders, once completed will be added to the set of training data. Figure 10 demonstrates the mentioned steps.

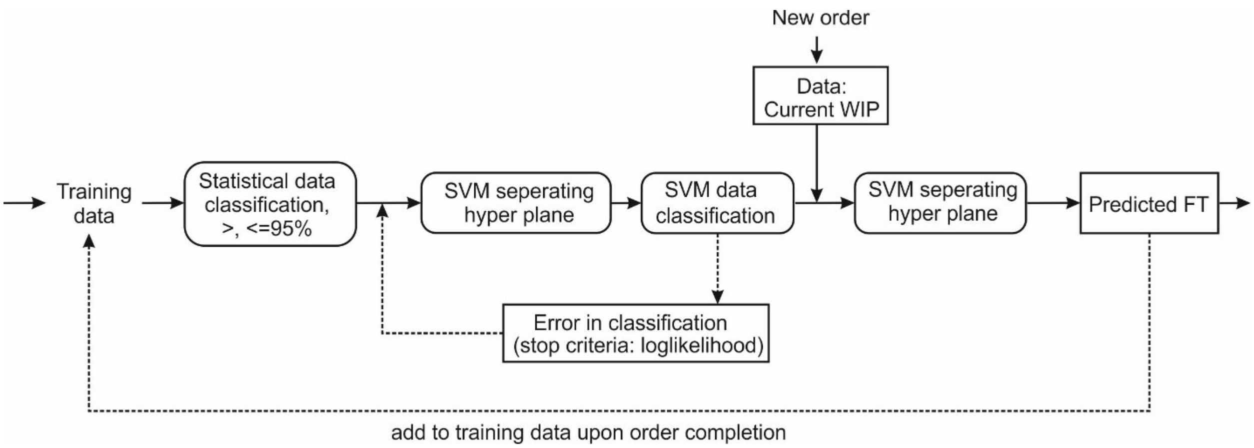


Figure 10: Flow time prediction model illustrated.

5. Experimentation

5.1 Setup

To test the operation and the results of the model, and to demonstrate the value of the presented SVM modelling approach, we employed different real and simulated experiments. To replicate the manufacturing process, Discrete Event Simulation was implemented via ARENA software. Grouping of data was undertaken utilizing a MATLAB as our built algorithm. We also employed the MATLAB's `FMINCON` optimization library in building the SVM model. Some general settings for instances used include: (1) Jobs cannot be preempted; (2) A single operation can be done on a machine at a time;(3) Workers are assigned to specific machines and cannot work elsewhere; (4) The number of workers does not change;(5) Setup times of machines are negligible and (6) Exponential distributions for arrivals and normal distributions for service times are used. Note that other distributions are also possible if such distributions mimic the reality.

The descriptive statistics for the different groups were produced using SPSS. Different manufacturing systems were configured to collect the necessary data. The targeted manufacturing systems in our proposed model consist of different resources distributed across the job shop floor. It is further recognized that on that shop floor, orders arrive at random times from different customers requiring different products. However, each manufacturing system maintains a certain product mix policy. Accordingly, the routing of orders may differ depending not only upon the type of products that are ordered, but also upon available resources (e.g. the factory machines); hence, each product type maintains its own route. Due to production heterogeneity, each product's influx creates different *WIP* at each resource. To obtain the training datasets, the actual flowtimes and *WIP* are collected from the simulated manufacturing systems for each arriving order. Upon arrival, the current level of *WIP* is recorded for the corresponding order. The Total flowtime is subsequently added to the same data records from the simulation results. The training data is then captured/inserted into the SVM model to compute the classifier function which will be used to predict the *FT* for any *WIP* level.

5.2 Experimental results

In this section, different real and simulated production examples are tested via the SVM flowtime model. The estimated flowtimes (from our results) are compared with the actual flowtime statistics attained via simulation. Although the SVM flowtime model is a classification algorithm, it was modified in this study to serve as a prediction tool. The percent of correct predictions is also tested. The percent of correctly predicted points is a practical measure of the model prediction power; however, the estimate of the *FT* is what we are seeking in this proposed model.

5.2.1 Experimental results – First example

To demonstrate the descriptive outcomes of the model, we consider an existing simple manufacturing system consisting of one resource that can accommodate up to 3 different product types; each type has its own processing time. Orders for each product type arrive at random. The anticipated due date for each order is estimated by the arrival time, adding in the predicted flowtimes. The collected data for this scenario is shown in Figure 11 for a set of 200 orders. The diagram allows us to visualize the correlation between the *WIP* and the flowtime. For instance, the marked dot shows the flowtime (5.5hrs) of a completed order that arrived when *WIP* of the single resource was 5. In effect, 5 orders had to be processed ahead of the marked order.

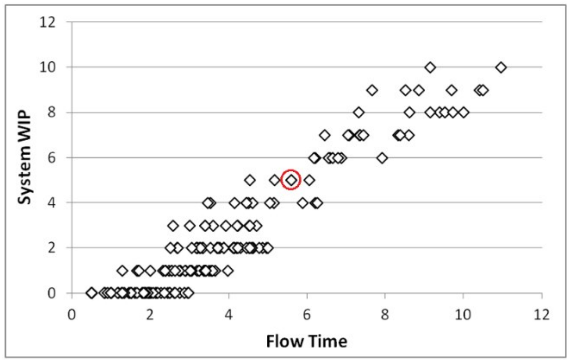


Figure 11: Flowtime vs. *WIP*

To use the given data in training, we are only interested in only the top 10% of flowtimes of each *WIP* group. This is because the top 10% are the relevant points that should be sufficient to identify the support vectors and the classifying function.

We emphasize, however, that by having more *WIP* intervals, smoother classifier functions will result. To start with, suppose the *WIP* is divided into two intervals. Figure 11 shows that the minimum observed *WIP* is "0" and the maximum is "10". The data resulted in the following descriptive statistics of the two regions as shown in Table 1.

Table 1: Descriptive statistics of example 1.

	Flowtime			
WIP interval	*Min*	*Max*	*Average*	*FT @ 95%*
[0-5]	0.5	6.26	2.7752	5.1068
(5-10]	6.17	10.95	8.1608	10.659

By considering the top 10% of each group, the data is reduced to 33 points. The classification of the new set of points is shown in Table 2.

Table 2: The classification of the relevant points.

FT	*WIP*	Class
5.164✦	5.00	1
6.212	4.00	1
4.968✦	2.00	-1
4.872	2.00	-1
4.628	2.00	-1
4.703	3.00	-1

5.893	4.00	1
6.055	5.00	1
5.604	5.00	1
5.146	4.00	1
6.259	4.00	1
4.592	2.00	-1
4.784	2.00	-1
4.603	4.00	-1
5.604	5.00	1
5.058	4.00	1
4.585	2.00	-1
5.000✦	2.00	1
9.997✦	8.00	-1
10.407✦	9.00	1
9.717	8.00	-1
10.500	9.00	1
10.955	10.00	1

✦: Means the point is a support vector

By optimizing the SVM model using the above set of points, we obtain the following Support Vector:

$$SV = \begin{pmatrix} 5.1643 & 5.000 \\ 4.9676 & 2.000 \\ 5.0000 & 2.000 \\ 9.9973 & 8.000 \\ 10.4072 & 9.000 \end{pmatrix}$$

The corresponding classification of the above support vector can be found from the training data set which is given by: y_i=[1 -1 1 -1 1]. The related dual vector of the support vector is: $\hat{\alpha}$ = [3.1293 480.6270 480.0021 6.4177 3.9133] and the intercept *b* = -35.2791. Any testing point $\boldsymbol{z}$ can be classified by the resulting function:

$$\begin{aligned} f(\boldsymbol{z}) &= \textstyle\sum_{i=1}^{s} \alpha_m y_m \boldsymbol{WF}_m \cdot \boldsymbol{z} + b \\ &= 3.1293\,(1)(\langle 5.1643 \quad 5.000\rangle \cdot \langle \boldsymbol{z}\rangle) + \\ &\quad 480.627(-1)(\langle 4.9676 \quad 2.000\rangle \cdot \langle \boldsymbol{z}\rangle) + \\ &\quad 480.0021(1)\,(\langle 5 \quad 2\rangle \cdot \langle \boldsymbol{z}\rangle) + \\ &\quad 6.4177(-1)\,(\langle 9.9973 \quad 8\rangle \cdot \langle \boldsymbol{z}\rangle) + \\ &\quad 3.9133(1)\,(\langle 10.4072 \quad 9\rangle \cdot \langle \boldsymbol{z}\rangle) \\ &\quad -35.2791, \end{aligned}$$

where the "·" represents the kernel operation. If $f > 0$, then the point belongs to the class denoted by "1" (i.e., the *FT* is above the 95%) otherwise it will be in the "-1" class. The resulting separating line and the separating strip which is bounded by the contours (-1 and 1) are shown in Figure 12.

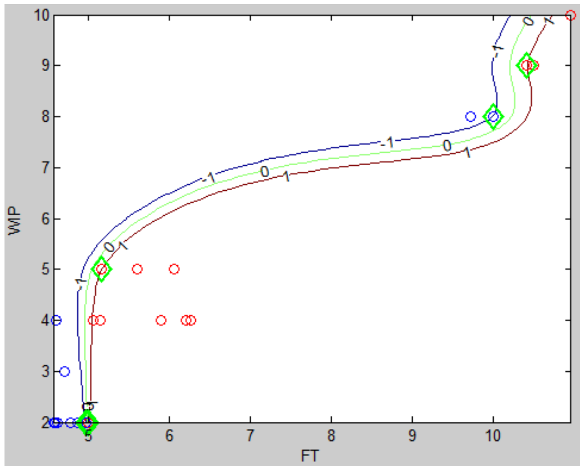
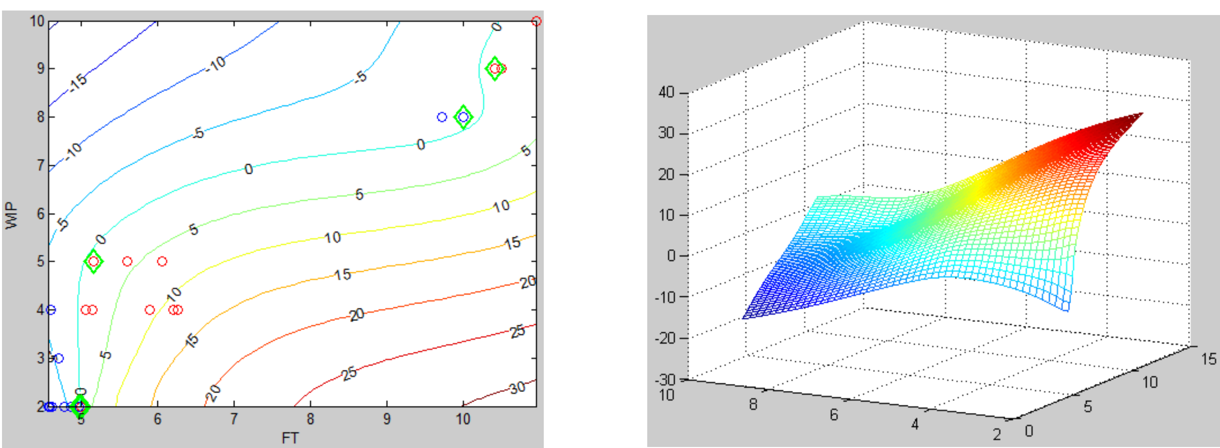


Figure 12: The separating hyper plane, and strip borders

The entire set of contours for the classifier are shown in Figure 13-*a*. The contours represent the classifier elevation. Note the contour "0" is the separator line, which represents the 95% border of the flowtimes. Figure 13-*b* shows a 3D representation of the classifier along with the "0" contour.



(*a*) (*b*)

Figure 13: a) the classifier contours, b) three dimensional sketch of the classifier

Once the classifier function and the decision hyper plane are established, we can easily test the classification of the training data as well as any other data set. It is however observed that the classifier function can serve two purposes. The first purpose is to check if an arriving order will take more than 95% of the flowtime required for similar *WIP* levels. The second purpose is to estimate the flowtimes at this satisfaction level using the bisection method via solving equation (16).

To test the goodness of our proposed model, consider an order arriving at a *WIP* level of 7. By searching for the flowtime that satisfies *Equation 16*, we find that *FT*=7.335 (Figure 11). This flowtime value satisfies 95% of orders arriving at similar *WIP*. Accordingly, only 5% of the orders can be expected to exceed the estimated value. Similarly, suppose that an order arrives when the current *WIP* is 8 and the assigned flowtime is 5 time units. By testing such point where $\boldsymbol{z} = [5, 8]$ we find $f(\boldsymbol{z}) = -12.1608$ which means this due date is way less than the 95% and should not have been agreed to.

5.2.2 Experimental results – Second example

The same example above is tested again here. However, this experiment is undertaken with different number of intervals to examine the effect of the *WIP* regions on the estimated *FT*. Higher number of *WIP* regions results in smoother classifiers. The *FT* tends to asymptotically approach the correct value as shown in Figure 14 for estimating the *FT* at a *WIP* of 18.

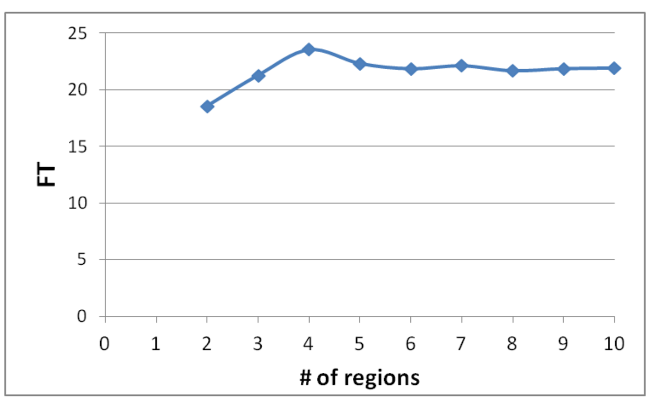


Figure 14: Flowtime asymptotically approaching the accurate value.

The classifier of a production system resulting from 500 orders and 5 *WIP* intervals is shown in Figure 15.

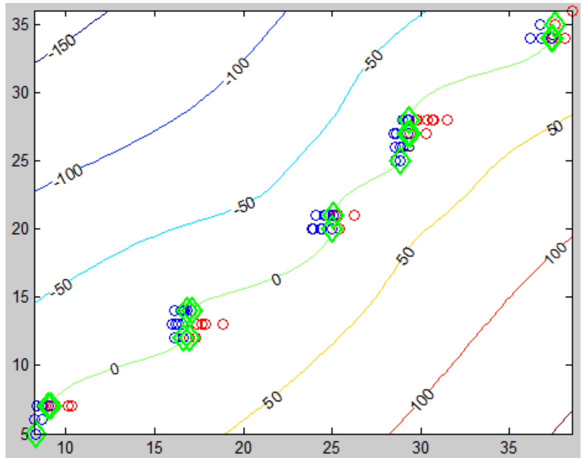


Figure 15: The separation hyper surface and the classifier contours of example 2.

6. Testing and validation

To test the resulting flowtimes, simulated replications are required. Once the same training sets are employed during testing, the percent of correctly classified points are found to be around 100%. This will be the case as the classifying function would already have emerged from the training data. This is particularly the case when the constant C is set to infinity, which dictates a "0" misclassifying tolerance.

Consequently, at least two sets of data points are required; one for training and the other for testing and validation. Discrete event simulation is undertaken with various streams of random numbers following different distributions (exponential and normal) to produce both the training and testing data. For the test data, the points will be statistically classified according to the resulting flowtimes; that is, the 95% level will be found from the data, and then the points are classified accordingly. This procedure will then be followed by: (i) the extraction of the relevant points (i.e. five percent above and below the 95%), (ii) the capturing/insertion of the resulting relevant data into the SVM model to build the classifier and (iii) the testing of the classifier with new data sets to compare the predicted and observed flowtimes.

6.1 Testing and validation: First Experiment - Testing the 95% Service Level

In this experiment, the 95% separating hyper plane is tested. More specifically, a comparison between the estimated 95% levels using the SVM model and the statistically estimated level is undertaken. The statistically calculated values are derived directly from the observed data by identifying the 95% quantile level using SPSS. In fact, since the *WIP* values are always integers in this experiment, the observed flowtimes can be processed via factorial analysis according to the *WIP* using SPSS software. More specifically, for each *WIP* value, a group of *FT* is analyzed and then compared with the results of the SVM modelling. By employing the previous example against 1500 jobs, the following 95% service levels were found.

Table 3: Results of the statistical analysis and SVM model.

WIP	95% (statistical analysis)	95% (SVM model)	%error
1	3.809095	4.24	10.1628632
2	4.91597	4.82	1.99107884
3	5.783689	5.65	2.36616991
4	6.737134	6.68	0.85529865
5	8.32407	8.52	2.29964319
6	9.965443	9.86	1.06939757
7	11.28187	11.07	1.91388482
8	11.87967	12.44	4.50422749
9	13.41986	13.53	0.81400961
10	14.4505	13.91	3.88569662
		Average =	2.98622699

Table 3 shows that the error margin is less than 5%. Since we are comparing the statistical value with that found using the SVM model, the percent error is found by dividing the difference between both values by the statistical value. This indicates a robust prediction power. In other words, the SVM estimated 95% levels are sufficiently accurate. Hence, once an order arrives, the *WIP* status can be plugged into the classifier search algorithm to find the 95% threshold which can be quoted to customer.

Let us consider the last case in Table 3. The resulting separator is shown in Figure 16. The figure shows that the smothering of the 95% line suggesting more robust predictions.

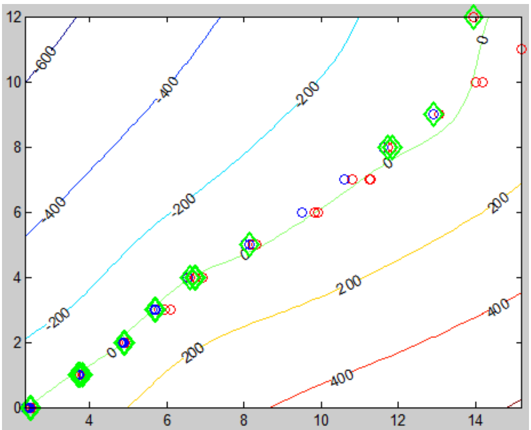


Figure 16: The 95% flowtime and the classifier contours.

6.2 Testing and validation: Second Experiment

In this experiment, a manufacturing system consisting of 3 resources and a product mix of 3 types is simulated. The resulting data points are filtered by 5 *WIP* regions at each resource. The 95% *FT* value was found for those groups maintaining enough data points (Table 4). Although the statistically computed 95% level is also an estimate, the SVM predicted values conform to the statistically found estimates with low error values.

Table 4: Comparison between statistical SVM predicted results.

	WIP vector	95% (statistical analysis)	95% (SVM model)	%error
1	[0 0 0]	6.9863	8.9	21.50216
2	[0 0 2]	8.4104	8.64	2.657142
3	[0 0 3]	8.8152	8.79	0.287002
4	[0 0 4]	9.4583	11.75	19.50387
5	[0 0 5]	11.2256	12.4	9.470831
6	[0 0 6]	14.7094	12.76	15.27732
7	[0 0 7]	13.3655	11.67	14.52877
8	[0 0 8]	8.4850	9.27	8.468289
9	[0 1 0]	10.6868	8.9	20.07626
10	[0 1 3]	11.4235	12.16	6.056743
11	[0 1 4]	8.8099	9.42	6.476645
12	[1 0 0]	9.4405	8.88	6.311937
13	[1 0 1]	11.4912	8.49	35.34982
14	[1 0 2]	10.2473	9.31	10.06767
15	[1 1 0]	10.9544	9.48	15.55294
			Average	12.77249

For better demonstration, the same quantiles shown above for both the statistically derived and predicted values are shown in a tornado chart. The values of the flow times are sorted from the highest to the lowest. Note that a symmetrical tornado means less deviations between both the actual and predicted values. Fig. 17 shows a symmetry which reveals less deviation from the real flowtime values.

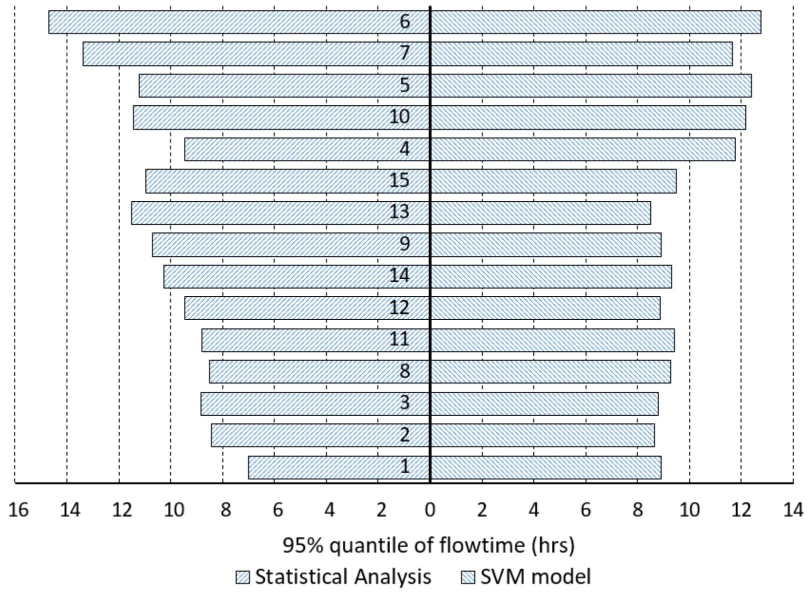


Figure 17: Tornado chart of real and predicted flowtimes.

The same configuration was tested by 10 other replications of the same manufacturing system (Table 5). All the data sets are used for training and then testing. Table 6 then shows the percentages of points predicted correctly. Note that the SVM model shows high *FL* class prediction power. The number of points which were misclassified by the model is reducible by increasing the number of regions of each *WIP* and by means of increasing the simulation length to achieve a better representing sample. However, increasing the number of *WIP* intervals results in excessive computational time as the number of resulting groups is given by m^R.

Table 5: Correct predictions of a production system of 1500 jobs. The 10 data sets used for training as well as testing the SVM

	Test data sets									
Training sets	1	2	3	4	5	6	7	8	9	10
1	1278	1060	1230	1303	1262	1111	1044	1180	1363	1264
2	1268	1311	1585	1308	1257	1201	1168	1242	1379	1308
3	1267	1102	1254	1316	1197	1099	1104	1179	1370	1270
4	1150	981	1109	1294	1133	1007	996	1097	1312	1207
5	1172	1076	1162	1242	1232	1112	1073	1200	1321	1251
6	1382	1282	1344	1365	1368	1270	1252	1296	1375	1349
7	1215	1112	1246	1322	1272	1183	1169	1223	1344	1279
8	1133	1068	1161	1270	1214	1141	1078	1244	1399	1224
9	1220	1057	1168	1277	1183	1028	990	1132	1248	1236
10	1128	989	1123	1250	1134	1016	1000	1076	1337	1247

The percentage of correct predictions is given by Table 6. The column at the extreme right hand of the table shows the average percent of each SVM model. The grand average of the data is approximately 86.3%.

Table 6: Correct prediction percent

	Test data sets										
	1	2	3	4	5	6	7	8	9	10	Ave.
1	0.91	0.76	0.88	0.93	0.90	0.79	0.75	0.84	0.97	0.90	0.86
2	0.91	0.94	1.13	0.93	0.90	0.86	0.83	0.89	0.99	0.93	0.93
3	0.91	0.79	0.90	0.94	0.86	0.79	0.79	0.84	0.98	0.91	0.87
4	0.82	0.70	0.79	0.92	0.81	0.72	0.71	0.78	0.94	0.86	0.81
5	0.84	0.77	0.83	0.89	0.88	0.79	0.77	0.86	0.94	0.89	0.85
6	0.99	0.92	0.96	0.98	0.98	0.91	0.89	0.93	0.98	0.96	0.95
7	0.87	0.79	0.89	0.94	0.91	0.85	0.84	0.87	0.96	0.91	0.88
8	0.81	0.76	0.83	0.91	0.87	0.82	0.77	0.89	1.00	0.87	0.85
9	0.87	0.76	0.83	0.91	0.85	0.73	0.71	0.81	0.89	0.88	0.82
10	0.81	0.71	0.80	0.89	0.81	0.73	0.71	0.77	0.96	0.89	0.81
									Grand ave.		0.863

6.3 Testing and validation: Third Experiment

In this experiment, a manufacturing system of 10 resources produces a mix of 5 products. Each product type has its own processing time set against each resource. The system is configured in ARENA software to generate the necessary training data. Only 2 regions are used for the *WIP* which results in $2^{10}=1024$ separate groups. A total 1500 orders were simulated. The training data were reduced by the classification procedure in order to facilitate the selection of the relevant points. These relevant points were subsequently captured by the SVM model. In this experiment, since the number of groups is significantly large, a few samples from those groups which consist of more than 20 data points were selected. Table 7, presents the *WIP* at the 10 resources, the actual time needed to complete the order, the 95% flowtime level estimated statistically and the SVM 95% flowtime estimate and the percent error, respectively.

Table 7: WIP of resources upon order arrival

	WIP of resources upon order arrival											95%		
#	1	2	3	4	5	6	7	8	9	10	Observed *FT*	SPSS	SVM	% error
1	1	2	1	0	0	4	1	0	1	0	2.67	18.085	17.85	1.316527
2	0	2	0	3	1	0	3	1	3	0	15.06	16.400	16.40	0
3	2	1	1	0	0	3	2	0	3	1	13.26	17.9105	17.81	0.56429
4	0	1	3	0	0	0	0	11	1	0	11.64	28.9897	29.91	3.077065
5	0	1	1	1	0	8	1	0	1	0	22.65	22.2125	21.56	3.026438

6	2	1	0	0	10	0	0	7	0	0	32.36	39.1705	38.71	1.189615
7	0	5	2	6	0	0	0	2	0	0	23.36	23.400	24.83	5.759162
8	4	5	7	2	1	2	0	3	1	1	29.23	26.774	27.63	3.098082
9	0	7	3	3	0	0	0	4	0	0	25.59	27.752	27.42	1.210795
10	1	10	1	7	0	1	1	1	0	0	28.58	32.629	31.25	4.4128
11	3	6	7	2	2	2	0	2	1	0	27.85	29.600	29.34	0.886162
12	7	7	2	5	0	1	1	2	0	1	34.49	36.548	35.50	2.952113
													average	2.291087

The error in estimating the 95%, as shown in Table 7 is significantly small (with a percent average of 2.29%). This result supports the proposition that the 95% of *FT* data can be easily estimated by the classifier resulting from the SVM. In fact, once the support vector is constructed, the classifier can be written in a simple closed form, although this is with the ability to accommodate highly inseparable data points. The flowtimes estimated by the classifier function achieve a 95% service level of all orders arriving at similar system status.

6.4 Managerial Implications

When orders arrive at a firm, the due-date assignment question comes to the scene during sale negotiations with customers. Usually, if the due date offered far exceeds the one expected by customers, the company has to offer a price reduction. In most of the cases, due dates are discussed by the customers rather than strictly dictated. Late due dates, although dissatisfying for customersprovide do at least improve chances of finishing and shipping the orders on schedule. To retain a good consumer profile, many businesses accept fair holding and tardiness costs in favor of dictating their due dates. The decision-maker must also weigh the losses arising from the cost of the keeping and the benefits of delivering the orders on time.

In most current job shop scheduling studies, all processing times are assumed to be deterministic and known. However, due to breakdowns, variability in manufacturing conditions, scarcity and readiness in resources, human or computer exhaustion, learning, and random arrivals, processing times of jobs may not be constant in real-life output. Multitasking scheduling and diversity in the product mix render due date assignments more complex in realistic applications.

Therefore, in this context, the objective is to determine the right value of the due date so as to reduce penalties and increase service quality. Upon the arrival of an order, it is important to quote a due date that satisfies both the customer and the manufacturer. However, with the uncertainty considered in the settings of our model, it is important to have previous data about both the actual due dates and the *WIP* status when previous orders arrived at the job shop under consideration. Without historical data, it will not be possible to estimate the 95% threshold that can help deliver estimated due dates under such satisfaction levels. The model can be implemented by building a simple software with a user-friendly interface that can read the data and estimate flow times /due dates for any *WIP* values. While the model falls in the category of static approaches, it can accommodate the new collected data by further runs of the optimization model. Compared to problem settings of deterministic service times and arrival rates, this model is applicable only when there is significant variability in such parameters.

7. Conclusions

As businesses increasingly take a global perspective, both scholars and practitioners have been particularly interested in finding ways to ensure manufacturing competitiveness. One such approach involves the concept of workload control which is rooted in the '*just-in-time*' production and manufacturing philosophy. Due date setting or assignment is one of the core workload control mechanisms. Of particular relevance is that due date assignment is central to the contractual relationship between manufacturers and customers who purchase and utilize their products and services. However, although it is clearly desirable to conduct business by adhering to contractually agreed roles and specifications, manufacturers may often be unable to deliver on negotiated and assigned due dates. Where forecasting has been employed, assigned due dates which are either early or late are associated with a range of undesirable and unintended consequences. The factors involved, as discussed in the study, are varying. One such factor pertains to the complex nature of the relationship between flowtimes and the manufacturing system status. This complexity, we argued, serves as a major reason why the use of simple analytical functions to establish due date assignment in job shop manufacturing may not be viable. For this reason, we contended that the use of a robust analytical classification model, the SVM model, was a more appropriate means of carrying out precise estimates of assigned job shop manufacturing due dates. Drawing from the literature, we assumed a 95% service level as representative of an achievable desired service target. The SVM modelling was undertaken to develop a classifier function that informs on flowtimes for different *WIP* combinations. A duality analysis along with Lagrange relaxation were utilized to establish a strict form of the SVM classifier. A special Kernel function was employed to handle highly inseparable data. Training data sets were used to tune the classifier. The training data sets were statistically split into those which satisfied the 95% service level and those which did not. The classifier was trained by the “relevant set” of data points. The bisection method was used to solve for the classifier and to find the final *FT* values.

To test the descriptive merit of the model, different production systems were simulated, resulting in a substantial amount of training and testing data sets with different number of *WIP* intervals, product mixes and also a range of resources. The estimated flowtimes showed remarkably interesting conformance to the observed flowtimes, with error margins of less than 10% in most cases.

As shown, the model we propose provides manufacturing managers with the competency to be cognizant of the complex interrelationships between flowtimes and manufacturing system status, and also to be able to predict flowtimes of random orders for different process plans emanating from manufacturing systems with multiple resources. By incorporating a 95% service level, most importantly, manufacturing managers are able to maintain confidence of the realistic predictive capabilities of the SVM model. Arguably, among other potential benefits of our modelling for manufacturing managers is that much broader workload scheduling and control will be supported, particularly in environments with multiple operating machines. Thus, this study has a forecasting role. Successful use of the SVM to achieve high service levels (and by implication, customer satisfaction) suggests that there is considerable potential for use of SVM modelling in manufacturing systems. Of particular relevance are its ability to articulate complex functions and its low prediction errors in highly heterogeneous production environments. Indeed, SVM offers considerable potential in manufacturing areas such as job clustering and flowtime assignment, due not only to its robustness, but also to its adaptability for complex data.

References

Baker, K. and Scudder, G. 1990. Sequencing with earliness and tardiness penalties: a review. Operations research, 38(1), pp.22-36.

Baykasoğlu, A., Göçken, M. and Unutmaz, Z. 2008. New approaches to due date assignment in job shops. European Journal of Operational Research, 187(1), pp.31-45.

Bertrand, J., and Wortmann, J. 1981. Production Control and Information Systems for Component-Manufacturing Shops. Amsterdam: Elsevier Scientific Publishing Company.

Blackstone, J., Phillips, D. and Hogg, G. 1982. A state-of-the-art survey of dispatching rules for manufacturing job shop operations. The International Journal of Production Research, 20(1), pp.27-45.

Breithaupt, J., Land, M. and Nyhuis, P. 2002. The workload control concept: theory and practical extensions of Load Oriented Order Release. Production Planning & Control, 13(7), pp.625-638.

Cattani, K., Dahan, E. and Schmidt, G. 2010. Lowest cost may not lower total cost: Using "spackling" to smooth mass-customized production. Production and Operations Management, 19(5), pp.531-545.

Cheng, T. and Gupta, M. 1989. Survey of scheduling research involving due date determination decisions. European Journal of Operational Research, 38(2), pp.156-166.

Cheng, T. 1988. Optimal common due-date with limited completion time deviation. Computers & Operations Research, 15(2), pp.91-96.

Dalalah, D. Al-Araidah, O. 2010. Dynamic decentralised balancing of CONWIP production systems. International Journal of Production Research 48 (13), 3925-3941.

Duan, K. and Keerthi, S. 2005. Which is the best multiclass SVM method? An empirical study. In International workshop on multiple classifier systems (pp. 278-285). Springer, Berlin, Heidelberg.

Ferris, M. and Munson, T. 2002. Interior-point methods for massive support vector machines. SIAM Journal on Optimization, 13(3), pp.783-804.

Frei, F. and Harker, P. 1999. Measuring the efficiency of service delivery processes: an application to retail banking. Journal of Service Research, 1, pp. 300–312.

Gansterer, M. 2015. Aggregate planning and forecasting in make-to-order production systems. International Journal of Production Economics, 170, pp.521-528.

Gershwin, S. 2018. The future of manufacturing systems engineering. International Journal of Production Research, 56(1-2), pp.224-237.

Gilat, A. 2008. MATLAB: An introduction with applications, New York, Wiley.

Gordon, V., Proth, J. and Chu, C. 2002. A survey of the state-of-the-art of common due date assignment and scheduling research. European Journal of Operational Research, 139(1), pp.1-25.

Graves, S. and Willems, S. 2003. Supply chain design: safety stock placement and supply chain configuration. Handbooks in Operations Research and Management Science, Vol. 11, pp.95-132.

Guenther, N. and Schonlau, M. 2016. Support vector machines. Stata Journal, 16(4), pp.917-937.

Hopp, W. and Sturgis, M. 2000. Quoting manufacturing due dates subject to a service level constraint. IIE Transactions, 32(9), pp.771-784.

Hsu, C. and Lin, C. 2002. A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks, 13(2), pp.415-425.

Husseini, S., O'Brien, C. and Hosseini, S. 2006. A method to enhance volume flexibility in JIT production control. International Journal of Production Economics, 104(2), pp.653-665.

Janiak, A., Janiak, W., Krysiak, T. and Kwiatkowski, T. 2015. A survey on scheduling problems with due windows. European Journal of Operational Research, 242(2), pp.347-357.

Johns, N. 1999. What is this thing called service? European Journal of Marketing, 33, pp. 958–973.

Kaminsky, P. and Kaya, O. 2009. Combined make-to-order/make-to-stock supply chains. IIE Transactions, 41(2), pp.103-119.

Kelton, K., Sadowski, R., Zupick, N. 2014. Simulation with Arena, 6th edition. McGraw-Hill Professional.

Kianpour, P., Gupta, D., Krishnan, K. and Gopalakrishnan, B. 2021. Automated job shop scheduling with dynamic processing times and due dates using project management and industry 4.0. Journal of Industrial and Production Engineering, DOI: https://doi.org/10.1080/21681015.2021.1937725.

Kim, T. 2018. An integrated model for production and batch shipments for ramp-up production. Computers & Industrial Engineering, 118, pp.138-152.

Koulamas, C. 1994. The total tardiness problem: review and extensions. Operations research, 42(6), pp.1025-1041.

Kramer, F. and Lee, C. 1993. Common due-window scheduling. Production and Operations Management, 2 (4), pp. 262-275

Kumar, A. 2007. From mass customization to mass personalization: a strategic transformation. International Journal of Flexible Manufacturing Systems, 19(4), pp.533-547.

Lee, Y., Lin, Y. and Wahba, G. 2004. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. Journal of the American Statistical Association, 99(465), pp.67-81.

Li, F., Chen, Z. and Tang, L. 2017. Integrated production, inventory and delivery problems: Complexity and algorithms. INFORMS Journal on Computing, 29(2), pp.232-250.

Lohmer, J. and Lasch, R. 2020. Production planning and scheduling in multi-factory production networks: a systematic literature review. International Journal of Production Research, DOI: https://doi.org/10.1080/00207543.2020.1797207, pp.1-27.

McLachlin, R. 1997. Management initiatives and just-in-time manufacturing. Journal of Operations Management, 15(4), pp.271-292.

Meyer, D., Leisch, F. and Hornik, K. 2003. The support vector machine under test. Neurocomputing, 55(1-2), pp.169-186.

Moodie, D. 1999. Demand management: The evaluation of price and due date negotiation strategies using simulation. Production and Operations Management, 8(2), pp.151-162.

Ojiako, U., Maguire, S. and Chipulu, M. 2013. Thematic elements underlying the delivery of services in high-contact public service encounters. Production Planning & Control, 24(6), pp.532-545.

Prabagaran, S. and Joseph, O. 2019. Dynamic due date assignment method: A simulation study in a job shop with sequence-dependent setups. Journal of Manufacturing Technology Management, 30(2), pp.421-437.

Quarteroni, A., Saleri, F. and Gervasio, P. 2006. Scientific computing with MATLAB and Octave (Vol. 2). Berlin: Springer.

Radasanu, A. 2016. Inventory management, service level and safety stock. Journal of Public Administration, Finance and Law, 9, pp.145-153.

Rasti-Barzoki, M. and Hejazi, S. 2013. Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries for multiple customers in supply chains. European Journal of Operational Research, 228(2), pp.345-357.

Reiter, S. 1966. A system for managing job-shop production. The Journal of Business, 39(3), pp.371-393.

Rossetti, M. 2010. Simulation Modeling with Arena. John Wiley & Sons.

Saha, C., Aqlan, F., Lam, S. and Boldrin, W. 2016. A decision support system for real-time order management in a heterogeneous production environment. Expert Systems with Applications, 60, pp. 16-26.

Sagawa, J. and Land, M. 2018. Representing workload control of manufacturing systems as a dynamic model. IFAC-PapersOnLine, 51(2), pp.825-830.

Schonberger, R. 2019. The disintegration of lean manufacturing and lean management. Business Horizons, 62(3), pp.359-371.

Seidmann, A. and Smith, M. 1981. Due date assignment for production systems. Management Science, 27(5), pp.571-581.

Sen, T. and Gupta, S. 1984. A state-of-art survey of static scheduling research involving due dates. Omega, 12(1), pp.63-76.

Shabtay, D. 2016. Optimal restricted due date assignment in scheduling. European Journal of Operational Research, 252(1), pp.79-89.

Solomon, H., Jilcha, K. and Berhan, E. 2015. Lead time prediction using simulation in leather shoe manufacturing. In Afro-European Conference for Industrial Advancement (pp. 283-292). Springer, Cham.

Sridharan, V. and Li, X. 2008. Improving delivery reliability by a new due-date setting rule. European Journal of Operational Research, 186(3), pp.1201-1211.

Stevenson, M., Hendry, L., and Kingsman, B. 2005. A Review of Production Planning and Control: The Applicability of Key Concepts to the Make to Order Industry. International Journal of Production Research 43 (5), pp. 869–898.

Thürer, M., Land, M., Stevenson, M., Fredendall, L. and Godinho Filho, M. 2015. Concerning Workload Control and Order Release: The Pre-Shop Pool Sequencing Decision. Production and Operations Management, 24(7), pp.1179-1192.

Thürer, M., Stevenson, M., Land, M. and Fredendall, L. 2019. On the combined effect of due date setting, order release, and output control: an assessment by simulation. International Journal of Production Research, 57(6), pp.1741-1755.

Tseng, M., Tran, T., Ha, H., Bui, T. and Lim, M. 2021. Sustainable industrial and operation engineering trends and challenges Toward Industry 4.0: a data driven analysis. Journal of Industrial and Production Engineering, DOI: 10.1080/21681015.2021.1950227

Tseng, M., Jiao, R.J. and Wang, C. 2010. Design for mass personalization. CIRP Annals, 59(1), pp.175-178.

Tuong, N. and Soukhal, A. 2010. Due dates assignment and JIT scheduling with equal-size jobs. European Journal of Operational Research, 205(2), pp.280-289.

Vesanen, J. 2007. What is personalization? A conceptual framework. European Journal of Marketing, 41(5/6), pp.409-418.

Voss, C. 2003. Rethinking paradigms of service: service in a virtual environment. International Journal of Operations and Production Management, 23 (1), pp. 88–104.

Wang, Y., Ma, H., Yang, J. and Wang, K. 2017. Industry 4.0: a way from mass customization to mass personalization production. Advances in Manufacturing, 5(4), pp.311-320.

Wang, J. and Guo, Q. 2010. A due-date assignment problem with learning effect and deteriorating jobs. Applied Mathematical Modelling, 34(2), pp.309-313.

Xiong, X., Wang, D., Edwin Cheng, T., Wu, C. and Yin, Y. 2018. Single-machine scheduling and common due date assignment with potential machine disruption. International Journal of Production Research, 56(3), pp.1345-1360.

Yan, H., Wan, X. and Xiong, F. 2015. Integrated production planning and scheduling for a mixed batch job-shop based on alternant iterative genetic algorithm. Journal of the Operational Research Society, 66(8), 1250-1258.

Yang, T. and Han, J. 2021. Integrated Management Strategy with Feasible Smartness over Heterogeneous IoT Environments. Electronics, 10(2), p. 10.3390/electronics10020149

Yin, Y., Cheng, T., Xu, D. and Wu, C. 2012. Common due date assignment and scheduling with a rate-modifying activity to minimize the due date, earliness, tardiness, holding, and batch delivery cost. Computers & Industrial Engineering, 63(1), 223-234.

Yin, Y., Stecke, K., Swink, M. and Kaku, I. 2017. Lessons from seru production on manufacturing competitively in a high cost environment. Journal of Operations Management, 49, 67-76.

Zhao, C., Yin, Y., Cheng, T. and Wu, C. 2014. Single-machine scheduling and due date assignment with rejection and position-dependent processing times. Journal of Industrial and Management Optimization, 10(3), 691-700.

Zipkin, P. 1991. Does manufacturing need a JIT revolution? Harvard Business Review, 1/2 (1991), 40-50.