

Article

Lean Manufacturing Soft Sensors for Automotive Industries

Ravi Sekhar , Nitin Solke  and Pritesh Shah * Symbiosis Institute of Technology, Pune, Symbiosis International, Deemed University,
Pune 412115, Maharashtra, India

* Correspondence: pritesh.shah@sitpune.edu.in

Abstract: Lean and flexible manufacturing is a matter of necessity for the automotive industries today. Rising consumer expectations, higher raw material and processing costs, and dynamic market conditions are driving the auto sector to become smarter and agile. This paper presents a machine learning-based soft sensor approach for identification and prediction of lean manufacturing (LM) levels of auto industries based on their performances over multifarious flexibilities such as volume flexibility, routing flexibility, product flexibility, labour flexibility, machine flexibility, and material handling. This study was based on a database of lean manufacturing and associated flexibilities collected from 46 auto component enterprises located in the Pune region of Maharashtra State, India. As many as 29 different machine learning models belonging to seven architectures were explored to develop lean manufacturing soft sensors. These soft sensors were trained to classify the auto firms into high, medium or low levels of lean manufacturing based on their manufacturing flexibilities. The seven machine learning architectures included Decision Trees, Discriminants, Naive Bayes, Support Vector Machine (SVM), K-nearest neighbour (KNN), Ensembles, and Neural Networks (NN). The performances of all models were compared on the basis of their respective training, validation, testing accuracies, and computation timespans. Primary results indicate that the neural network architectures provided the best lean manufacturing predictions, followed by Trees, SVM, Ensembles, KNN, Naive Bayes, and Discriminants. The trilayered neural network architecture attained the highest testing prediction accuracy of 80%. The fine, medium, and coarse trees attained the testing accuracy of 60%, as did the quadratic and cubic SVMs, the wide and narrow neural networks, and the ensemble RUSBoosted trees. Remaining models obtained inferior testing accuracies. The best performing model was further analysed by scatter plots of predicted LM classes versus flexibilities, validation and testing confusion matrices, receiver operating characteristics (ROC) curves, and the parallel coordinate plot for identifying manufacturing flexibility trends for the predicted LM levels. Thus, machine learning models can be used to create effective soft sensors that can predict the level of lean manufacturing of an enterprise based on the levels of its manufacturing flexibilities.

Keywords: lean manufacturing; manufacturing flexibilities; soft sensor; automotive; neural networks



Citation: Sekhar, R.; Solke, N.; Shah, P. Lean Manufacturing Soft Sensors for Automotive Industries. *Appl. Syst. Innov.* **2023**, *6*, 22. <https://doi.org/10.3390/asi6010022>

Academic Editor: Christos Douligeris

Received: 10 January 2023

Revised: 1 February 2023

Accepted: 2 February 2023

Published: 3 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Lean manufacturing is a philosophical framework that aims to maximize value for customer through elimination or minimization of waste, i.e., anything that does not add value for the customer. Modern day auto manufacturing is characterized by demand for shorter lead times, diversification of existing product lines, faster introduction of newer products, and time to market. Hence, auto manufacturing firms need to successfully implement lean manufacturing at various levels to survive and thrive in the face of stiff market competition. Lean manufacturing focuses on reducing different types of wastes leading to loss in value, viz. wastage in motion, waiting period, inventory, production, rework, and more [1–4]. As per the lean manufacturing philosophy, the minimization of cost, effort, time, labour, and equipment-related expenses or wastes automatically boosts the productivity of a manufacturing firm. Effective management of manufacturing flexibilities

is the key to achieving these goals. The following subsection introduces manufacturing flexibilities in lean manufacturing.

1.1. Manufacturing Flexibilities in Lean Manufacturing

Lean manufacturing is easier to achieve and sustain if there are minimum uncertainties or fluctuations in product varieties, market demand, supply of raw materials, availability of labour/skilled labour, manufacturing operations, material handling, and inventory. A lot of wastage in industrial production is attributed to seen/unforeseen fluctuations in various aspects of business, operations, and supply chain [5–8]. Hence, flexibility in manufacturing is quintessential to achieve lean manufacturing. Manufacturing flexibility has been addressed, defined, and elaborated by many researchers [3,9,10]. The definition given by Sethi [3] is the most comprehensive, and is stated as the ability of an organization to manage production resources and uncertainty to meet various customer requests while maintaining the desired level of product quality, dependability, and price (cost). Primarily, the following manufacturing flexibilities have been found to have the maximum impact on lean manufacturing: product flexibility (PF), volume flexibility (VF), routing flexibility (RF), material handling flexibility (MH), machine flexibility (MF), and labour flexibility (LF) [3,10–14]. Extensive research has proven that effective management of the above-mentioned manufacturing flexibilities leads to significant improvements in lean manufacturing levels of an enterprise [15–17]. Manufacturing flexibilities help industries to effectively respond to the dynamic nature of supply chains and markets. By incorporating flexibilities in the various aspects of production, the manufacturing units are more robust against unforeseen variations. Manufacturing flexibilities directly result in effort, time, capital, and performance savings in the face of unforeseen bottlenecks. Managing operations without manufacturing flexibilities built in the job flow often turns out to be costlier [18]. Moreover, manufacturing flexibilities enable enterprises to manufacture and deliver products as per customer demands/requirements building brand value and trust.

The effective management of manufacturing flexibilities is challenging due to their mutual interdependence in various combinations. It is often complex to analyse their individual and combined effect on lean manufacturing [17,19]. Hence, researchers have investigated various aspects of the role of manufacturing flexibilities in lean manufacturing attainment [20]. Investigators have also studied the effects of these flexibilities on productivity, supply chains, and time to market [12,21–25]. Furthermore, manufacturing flexibilities have been classified and sorted on the basis of levels, hierarchies, capabilities, and competencies [3,13,25,26]. The following subsection presents the various analytical methodologies adopted by researchers to study the interrelationships between manufacturing flexibilities and lean manufacturing.

1.2. Analysis of Manufacturing Flexibilities in Lean Manufacturing

Over the past two decades, researchers have explored many different approaches to analyse and quantify the levels of flexibilities for lean industries, along with their interrelationships. Analysis of variance (ANOVA) was significantly used to apply regression and correlation-based statistical study of the different flexibilities and their respective impacts on lean manufacturing levels [27–29]. However, the ANOVA methodology was unable to reveal the mediation effects of the different parameters of manufacturing flexibilities on lean manufacturing. Hence, researchers [7] applied interpretive modeling to identify such indirect effects of the manufacturing flexibility parameters along with their dominant/non-dominant impacts on lean manufacturing. Furthermore, structural equation modeling (SEM) was applied to identify the most important flexibilities affecting lean manufacturing attainments [8]. Researchers also combined the SEM methodology with conventional statistical approaches to determine the underlying interrelationships among various flexibilities and the effects of these co-dependencies on lean achievement [30]. Predictive modeling and control theory was also explored to determine the exact levels of manufacturing flexibilities

required to contribute towards a desired level of lean level in auto parts manufacturing firms [31].

1.3. Lean Manufacturing in Industry 4.0

More recently, with the advent of industry 4.0 practices, researchers have started exploring integration of digital technologies with manufacturing processes towards improved lean attainment [32]. Researchers have reviewed the shift from contemporary statistical modeling to machine learning-based data-centric methodologies for quantifying lean manufacturing [33]. Reviews have also been conducted to document the recent trends in lean industry 4.0 [34] and sustainable manufacturing attainment using industry 4.0 [35]. Surveys have also been conducted to identify and classify the industry 4.0 implementation levels in more than sixty manufacturing industries in the developing economies of Brazil and India [36]. Prominent industry 4.0 technologies such as big data analytics have helped firms to better manage lean six sigma [37], green manufacturing [38], and supply chain management [39]. Virtual simulation technologies such as augmented reality are finding increasing applications in mainstream industrial practices such as Kanban, Kaizen, poka yoke, value stream mapping (VSM), and just in time (JIT) [40]. Widespread enterprise information systems are being projected as the basis of manufacturing data-driven industrial internet of things (IIoT) [41]. Furthermore, research has already begun on integrating lean manufacturing goals as the prominent benefits achieved through the IIoT [42]. Technologies such as automated guided vehicles (AGVs) are providing greater routing flexibility to the modern IIoT architectures [43]. Investigations are also being conducted on the identification and elimination of the various barriers in IIoT implementation [44]. In fact, researchers have found that IIoT integration improves all lean manufacturing practices (such as VSM and JIT) except for Kaizen, which has been more challenging to be implemented in IIoT frameworks [45]. Nevertheless, investigators have tried to design and implement process data-based lean architectures for smart manufacturing [46].

1.4. Machine Learning for Lean Manufacturing and Industry 4.0

The reliable functionality of any data-driven IIoT relies on effective system performance modeling. Machine learning is one of the most prominent modeling approaches used for estimating modern systems. Based on effective system modeling, machine learning methodology has also been used to optimize manufacturing processes in industries [47]. Machine learning has been employed to reduce machine changeover time, a parameter related to machine flexibility, based on the motion and time study data [48]. Machine learning methods have been used to simulate and optimize production processes based on shop floor data [49]. Researchers have studied and presented the multi-faceted relationship between intelligent machine learning approaches and digital manufacturing [50]. Machine learning-based predictive modeling and control of manufacturing techniques has also been explored [51]. Interestingly, machine learning has also been utilised to predict the success of budding entrepreneurs based on their attitudes and entrepreneurial orientations [52]. Studies have also been conducted to analyse the impact of machine learning and artificial intelligence (AI) techniques on entrepreneurial ventures [53].

1.5. Motivation, Aims, and Scope of the Present Study

Most of the above discussed studies are either surveys or involve propositions of theoretical and/or simulation frameworks integrating industry 4.0 technologies with lean manufacturing principles. Very few studies seem to have implemented machine learning approaches on actual industrial data to manage manufacturing flexibilities for lean attainment. The present study aims to fill this research gap by employing machine learning-based soft sensors to assess and classify lean attainment of industries based on the actual levels of their respective manufacturing flexibilities. Seven machine learning architectures (decision trees, ensembles, K-nearest neighbours, discriminants, support vector machine, artificial neural networks, and Naive Bayes) have been explored in this study to classify 46 auto

manufacturing firms into high, medium or low lean attainment categories based on the actual levels of their product, labour, machine, volume, routing, and material handling flexibilities. The following section provides details of the data collection and modeling methodologies followed in this study.

2. Methodology

2.1. Data Collection from Companies and Initial Analyses

The present study was based on the lean manufacturing and manufacturing flexibilities' data collected and analysed by Solke and Singh [30]. The authors surveyed forty-six automotive parts manufacturing firms spread across the Pune region of Maharashtra State, India. The authors floated expert-curated questionnaires (provided in the supplementary material) to directly and indirectly gauge the levels of lean manufacturing and manufacturing flexibilities in these participating firms. The alpha test was employed to ensure the statistical validity of the designed questionnaires. The exhaustive survey questionnaires sought feedback on up to fifty different parameters related to various flexibilities mentioned in the previous section, viz. MF, VF, PF, RF, LF, and MH. Another section of the questionnaire probed nine parameters related to lean manufacturing attainments. In terms of ownerships, 4 of the surveyed companies were owned publicly, whereas 42 companies were privately owned. The product-wise categories of these companies comprised 23 auto component manufacturers, 19 assembling units, and 4 sub-assembly production units. A total of 14 companies produced goods at large scale, followed by 14, 12, and 6 medium, small, and micro-scale industries, respectively. As far as the respective labour strengths are concerned, 16 companies employed more than 250 staff, 13 firms had employee strengths between 50 and 250 followed by less than 50 workers in the remaining 17 units.

The exhaustive survey data were checked for sampling adequacy and significance using the KMO and the Bartlett's sphericity tests. The resultant KMO value of 0.675 confirmed the sampling adequacy of the collected data. On the other hand, the sphericity test at a confidence level of 95% resulted in a p value of 0.000, confirming the significance of the statistical relationships between the various flexibilities and lean manufacturing levels of the surveyed firms. The levels of flexibilities and lean manufacturing for all firms were quantified by the authors following an analytical hierarchical process that involved experts from industry as well as from academia to assign suitable weights to different performance parameters. The resultant levels of the six manufacturing flexibilities and the corresponding lean manufacturing attainments of all 46 companies are depicted in Table 1. Finally, the authors used structural equation modeling to derive direct and indirect effects of flexibilities on lean manufacturing levels. In a subsequent study [31], the authors carried forward this study and implemented machine learning-assisted control theory to determine the exact levels of flexibilities needed by the surveyed firms to attain the desired lean manufacturing levels in their operations.

The present work aimed to take this study to its next level: implementing a plethora of machine learning-based models to explore the best performing lean manufacturing soft sensor. This soft sensor would be able to classify the lean manufacturing level of an auto component manufacturing enterprise as high, medium or low based on its existing levels of manufacturing flexibilities (Figure 1). Experimental/survey data-based predictive modeling, machine learning-based soft sensors, and system identification/control architectures have been proved to be quite effective in estimating critical system performance outcomes based on the operating conditions/parameters [54–60]. The next subsection gives details of the machine learning-based soft sensor modeling of lean manufacturing followed in the present work.

Table 1. Existing levels of lean manufacturing and manufacturing flexibilities in the surveyed companies [30].

Company No.	LF	MF	VF	PF	RF	MH	LM
1	0.6797	0.6715	0.7163	0.4855	0.6111	0.6817	0.7035
2	0.6144	0.6684	0.7410	0.6389	0.7222	0.5625	0.6896
3	0.7467	0.7946	0.7467	0.6618	0.6111	0.4021	0.8656
4	0.7143	0.7184	0.7002	0.9178	0.7778	0.8022	0.8630
5	0.7319	0.7491	0.7868	0.6523	0.7222	0.7500	0.7694
6	0.6841	0.5908	0.8294	0.5915	0.6111	0.6462	0.7379
7	0.6881	0.6240	0.7037	0.5727	0.6389	0.6192	0.8652
8	0.8636	0.8521	0.8911	0.7858	0.7500	0.8692	0.8852
9	0.7235	0.8043	0.4272	0.6631	0.7222	0.7146	0.7775
10	0.7139	0.8270	0.8592	0.6523	0.5000	0.6841	0.9635
11	0.7979	0.7735	0.6980	0.5247	0.5278	0.6536	0.8384
12	0.7901	0.8050	0.9559	0.7069	0.7778	0.8785	0.9076
13	0.6806	0.6601	0.7502	0.4559	0.7778	0.4376	0.7296
14	0.6532	0.6900	0.7937	0.6573	0.6944	0.6308	0.7751
15	0.7723	0.7821	0.7345	0.6573	0.6944	0.5505	0.8808
16	0.9096	0.8260	0.8131	0.6045	0.7222	0.6968	0.8080
17	0.9073	0.8444	0.5354	0.5727	0.6111	0.7654	0.8155
18	0.6585	0.6833	0.7796	0.7287	0.6944	0.6546	0.6826
19	0.7484	0.7334	0.7212	0.6440	0.6111	0.7500	0.7963
20	0.7473	0.8834	0.8038	0.6523	0.7500	0.7854	0.7879
21	0.6683	0.7262	0.6942	0.8977	0.8056	0.6158	0.7340
22	0.6908	0.6400	0.6207	0.5977	0.3611	0.3159	0.5057
23	0.5658	0.6555	0.6746	0.7105	0.6111	0.9729	0.6340
24	0.7463	0.7515	0.4737	0.7500	0.7222	0.7500	0.7869
25	0.8374	0.7700	0.7506	0.6809	0.6389	0.6308	0.7837
26	0.6557	0.6321	0.5776	0.5691	0.6667	0.6462	0.7369
27	0.7334	0.5750	0.4796	0.5500	0.6667	0.6817	0.5125
28	0.8380	0.8442	0.7113	0.6451	0.6111	0.5271	0.7124
29	0.9193	0.8253	0.9290	0.5616	0.7500	1.0000	0.7217
30	0.5857	0.8488	0.8680	0.5559	0.6111	0.5000	0.9065
31	0.5902	0.6631	0.6472	0.5879	0.6667	0.7500	0.8048
32	0.8341	0.6336	0.8143	0.6773	0.6111	0.5752	0.7340
33	0.8341	0.6336	0.8143	0.6190	0.6389	0.5752	0.7066
34	0.7323	0.7500	0.6521	0.7963	0.6389	0.7500	0.7208
35	0.7014	0.7084	0.5980	0.5832	0.6944	0.7500	0.7081
36	0.5556	0.6540	0.8226	0.5105	0.5833	0.5354	0.6476
37	0.6423	0.7445	0.8670	0.5871	0.6667	0.7500	0.6962
38	0.6697	0.7029	0.7716	0.5858	0.6111	0.6486	0.7743
39	0.7846	0.7228	0.7015	0.6914	0.7222	0.8715	0.7931
40	0.6098	0.6448	0.6839	0.6309	0.6944	0.6817	0.8530
41	0.7777	0.6671	0.7539	0.6856	0.6944	0.7229	0.7078
42	0.7304	0.7077	0.7112	0.7204	0.6389	0.7500	0.5203
43	0.8896	0.8735	0.8761	0.6407	0.8333	1.0000	0.9275
44	0.7335	0.7635	0.6556	0.7573	0.7222	0.7500	0.7714
45	0.6715	0.5939	0.6772	0.6477	0.6944	0.3751	0.7389
46	0.6011	0.6567	0.6773	0.5691	0.6667	0.7146	0.7611
Minimum	0.5556	0.5750	0.4272	0.4559	0.3611	0.3159	0.5057
Maximum	0.9193	0.8834	0.9559	0.9178	0.8333	1.0000	0.9635
Average	0.7266	0.7254	0.7280	0.6442	0.6685	0.6821	0.7618



Figure 1. Block Diagram of LM Soft Sensor.

2.2. Soft Sensor Modeling

A data-based modeling process typically starts with preprocessing such as data normalization, data cleaning, and more. Supervised machine learning algorithms learn, generate, and optimize a structure of relationships between the various features of the input/output datasets to arrive at the most optimal description of the system behaviour. In the present study, all 46 surveyed manufacturing firms were labeled to have either high, medium or low lean manufacturing levels as per the ranges shown in Table 2. Based on these ranges, 5 of the 46 surveyed firms were labeled low, whereas 27 and 14 companies were labeled medium and high LM class, respectively. Hence, the surveyed dataset included 10.87%, 58.70%, and 30.43% low, medium, and high LM class auto component manufacturers, respectively.

Table 2. LM levels for classification.

LM	Level Labels
0.50 to 0.65	low
0.65 to 0.80	medium
0.80 to 0.96	high

The manufacturing flexibility-LM dataset of the 46 companies shown in Table 1 was partitioned into training, validation, and testing datasets having 31 (67.39%), 10 (21.74%), and 5 (10.86%) companies, respectively. Table 3 shows the general LM level misclassification cost settings specified for all soft sensor architectures. These settings impel the machine learning-based soft sensors to maximize the LM classification accuracy based on minimization of misclassification costs.

Table 3. Misclassification cost default settings for all models.

		Predicted Class		
		High	Low	Medium
True Class	High	0	1	1
	Low	1	0	1
	Medium	1	1	0

The following subsections give details of each of the seven machine learning architectures explored in the current work for LM soft sensor modeling. All machine learning models were executed using Matlab software.

2.2.1. Artificial Neural Networks (ANN)

ANN is a bio-inspired computational network model composed of input, output, and hidden layers of “neurons”. ANN networks imitate the structure of interconnected neural connections in the human brain to learn relationships between various training dataset features. The input data features are fed to the input layer nodes of the ANN and assigned initial weights. Hidden layers are composed of nodes that hold the sums of products of each input and its corresponding weight. Thereafter, an activation function

is applied to the outputs of all hidden layer nodes and multiplied with respective output layer weights to finally give the model estimation. Figure 2 shows a general representation of the ANN architectures used in this study. In this figure, '*' indicates the presence of multiple instances of the same feature.

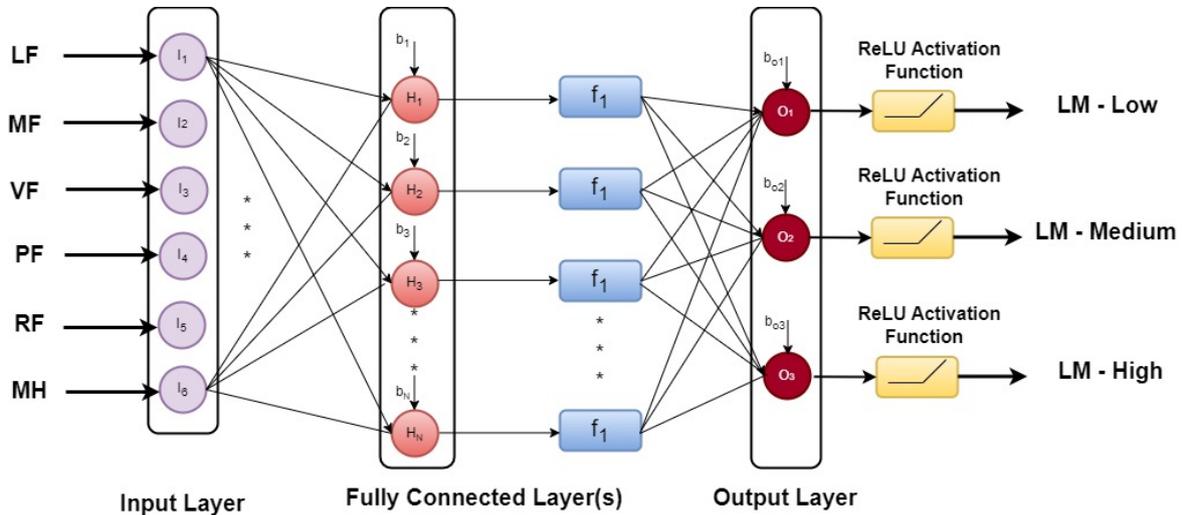


Figure 2. ANN Classification. '*' indicates the presence of multiple instances of the same feature.

In the present study, a rectified linear unit (ReLU) activation function was employed for improved computation performance. The ReLU activation is given as follows:

$$f(x) = \max(0, x) \tag{1}$$

This function returns 0 if it receives any negative input. For any positive value of x , it returns the same value. Hence, it gives an output ranging from 0 to infinity and prevents the neurons from assuming very low values. Figure 3 depicts the ReLU activation function response for different values of x .

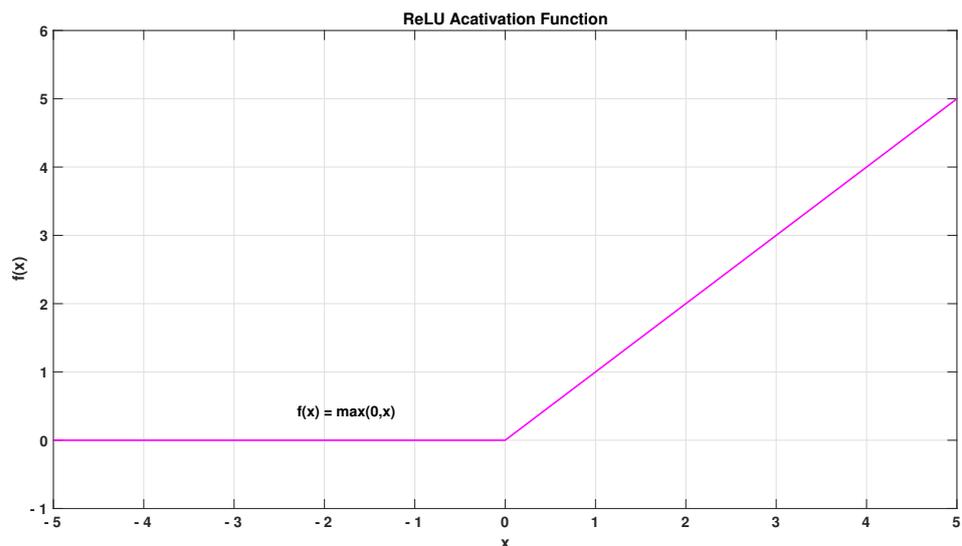


Figure 3. ReLU Activation Function.

The hidden and output layer nodal weights in an ANN structure represent the underlying relationships among the different features of the training dataset. In a feedback ANN, estimation errors are propagated back through output to input layers to re-adjust nodal weights to improve prediction in the next iteration. On the other hand, this re-adjustment of

nodal weights takes place from the input towards output layer in a feedforward ANN. Variations of ANN structures include different numbers and sizes (number of neurons/nodes in a particular layer) of fully connected layers.

Table 4 depicts the hyperparameters of the various ANN architectures explored for LM soft sensing—narrow, medium, wide, bilayered, and trilayered networks. Narrow, medium, and wide ANN have just one fully connected layer, whereas bilayered and trilayered ANN have two and three fully connected layers, respectively. All layers have 10 neurons each (layer size), except for 25 and 100 neurons in the medium and wide ANN, respectively. Regularization was kept at zero values to prevent underfitting of the ANN soft sensors. All data were standardized, i.e., scaled to fit a standard normal distribution. Maximum 1000 iterations were fixed for all networks.

Table 4. ANN classification hyperparameters.

Model Hyperparameters	Narrow ANN	Medium ANN	Wide ANN	Bilayered ANN	Trilayered ANN
Preset Neural Network	Narrow	Medium	Wide	Bilayered	Trilayered
Number of fully connected layers	1	1	1	2	3
Layer Size	10	25	100	10, 10	10, 10, 10
Activation Function	ReLU	ReLU	ReLU	ReLU	ReLU
Iteration Limit	1000	1000	1000	1000	1000
Regularization strength (Lambda)	0	0	0	0	0
Standardized data	Yes	Yes	Yes	Yes	Yes

2.2.2. K-Nearest Neighbours (KNN)

The KNN algorithm works on the premise “birds of a feather flock together”. Translated in terms of a dataset, this premise suggests that data points having similar attributes are located closer to each other as compared to data points having dissimilar characteristics. KNN classifies a query based on its relative distances from a specified number of neighbouring data points (K) closest to the query. KNN sorts the K-nearest neighbouring data points in the ascending order of their distances from the query point and classifies the query point as per the mode of the K-nearest data point labels. Figure 4 shows a pictorial representation of KNN classification.

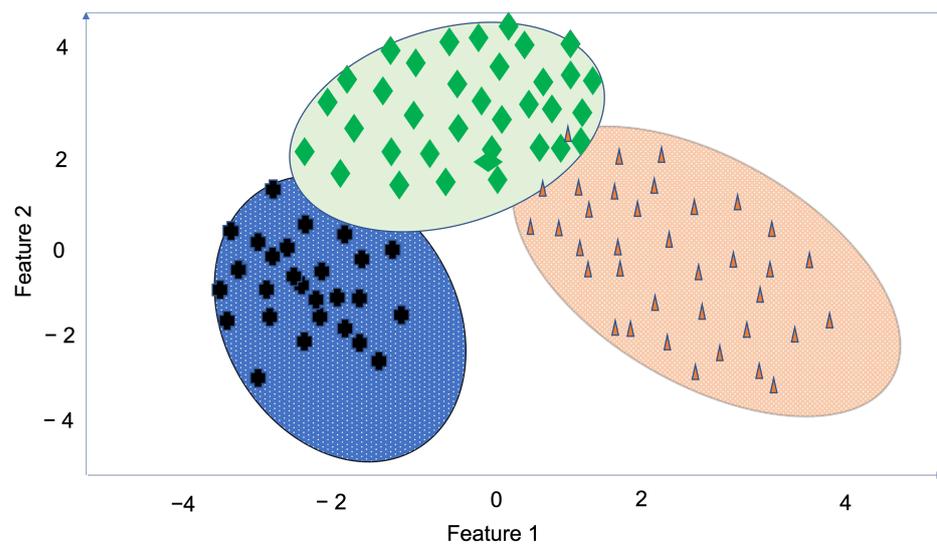


Figure 4. KNN Classification.

The classification accuracy of the KNN algorithm depends on the appropriate selection of the K value as per the dataset in question. Increasing the K value improves prediction accuracy to a certain extent due to averaging/majority voting. Beyond a certain limit, the

prediction errors start rising because the algorithm starts considering data points from the neighbouring clusters of dissimilar characteristics/labels. In a nutshell, the KNN algorithm methodology is simple, versatile, and easy to implement. It can be used for regression, search, and classification tasks. The main drawback of this algorithm is its inability to quickly solve problems involving large datasets. Table 5 shows the classification hyperparameters of the six KNN architectures explored in the present work. The number of K neighbours is ten in all cases except for the fine and cosine KNNs, which have 1 and 100 neighbours, respectively. The distance weights were kept equal and flexibilities-LM data were standardized for all architectures. The fine, medium, and coarse KNNs employed the Euclidean distance metric given by:

$$E_d(x, y) = \sqrt{\sum_{j=1}^n (y_j - x_j)^2} \tag{2}$$

Table 5. KNN classification hyperparameters.

Model Hyperparameters	Fine KNN	Medium KNN	Coarse KNN	Cosine KNN	Cubic KNN	Weighted KNN
Preset	Fine	Medium	Coarse	Cosine	Cubic	Weighted
Number of Neighbors	1	10	100	10	10	10
Distance Metric	Euclidean	Euclidean	Euclidean	Cosine	Minkowski (cubic)	Squared Inverse
Distance Weight	Equal	Equal	Equal	Equal	Equal	Equal
Standardized data	True	True	True	True	True	True

The cosine distance metric used in the cosine KNN is given by:

$$\cos(\theta) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|} = \frac{\sum_{i=1}^n (x_i y_i)}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \tag{3}$$

The Minkowski distance metric was used in the cubic KNN, expressed as follows:

$$M_d(x, y) = \left(\sum_{j=1}^n |y_j - x_j|^p\right)^{1/p} \tag{4}$$

The inverse squared distance used in the weighted KNN is calculated as follows:

$$I_d(x, y) = \frac{1}{\sum_{j=1}^n |y_j - x_j|^2} \tag{5}$$

where x and y represent inputs and corresponding outputs, respectively. \vec{x} and \vec{y} are the respective input and output vectors. θ is the angle between the input and output vectors. n is the total number of samples in the dataset and p is a real number typically set between 1 and 2.

2.2.3. Trees

Decision tree is a tree-shaped model of a set of hierarchical decisions that eventually lead to a regression or classification result. Trees have their “root” nodes at the top from where it splits into branches based on certain feature-based conditions (Figure 5).

Each of these branches may further split into sub-branches based on more specific sub-feature conditions. The splitting of branches may continue till they reach the leaf node, that is, the final classification decision is reached and no further splitting is needed. Thus, trees represent classification models whose decisions are built in their hierarchical structure which are simple to visualize, understand, and interpret. Feature selection is an implicit property of the Trees, and practically no data preprocessing is required. Trees are not affected by any nonlinear parametric relations, and can handle categorical as well

numerical data. However, too big and detailed tree models often tend to be overfitting. Over-complex trees are unable to bring out general classifications of the data. Moreover, unbalanced datasets lead to biased tree models that give prominence to the dominant data classes. To avoid complexity of too many split branches, a “greedy” algorithm compares different split options using a cost function and selects the feature-based split with the lowest cost to proceed. Splitting can also be limited by specifying a minimum number of training inputs required on each node or by limiting the maximum depth (path from root to leaf)/splits of the overall tree model. Cost function-based “pruning” may also be carried out to eliminate branches that are based on less important data features. Methods such as boosting and bagging are also applied to minimize instabilities/variances in tree models.

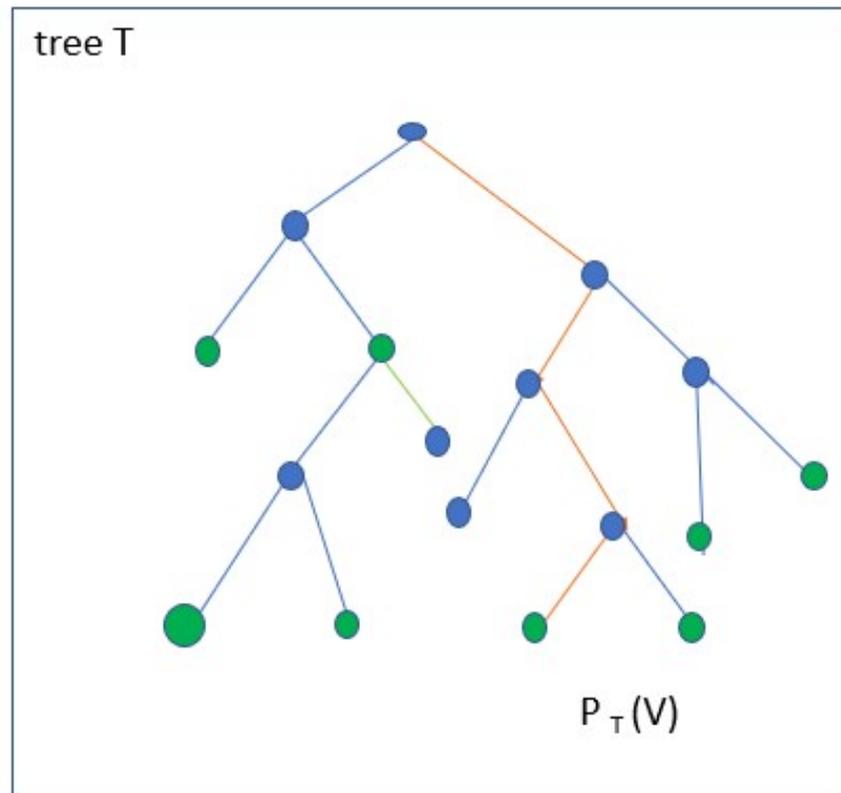


Figure 5. Tree Classification.

Table 6 shows the classification hyperparameters of the decision trees used for LM soft sensor modeling. The maximum number of branch splits was limited to 100, 20, and 4 for the fine, medium, and coarse trees, respectively. Surrogate splits were not needed since there were no missing samples in the training dataset. The Gini’s diversity index split criterion is given as follows:

$$Gini = 1 - \sum_{i=0}^C (P_i)^2 \tag{6}$$

where C is the number of classes and P_i is the probability of the i th class.

Table 6. Tree classification hyperparameters.

Model Hyperparameters	Fine Tree	Medium Tree	Coarse Tree
Preset	Fine	Medium	Coarse
Maximum number of splits	100	20	4
Split criterion	Gini’s diversity index	Gini’s diversity index	Gini’s diversity index
Surrogate decision splits	OFF	OFF	OFF

2.2.4. Discriminants

Linear discriminant is a supervised dimensionality reduction algorithm that maximizes inter-class discrimination/separability. This algorithm transforms the dataset features into a linear discriminant axis to reduce feature dimensions. It measures mean and variance of each class and maximizes the difference between the mean of each class to ensure maximum separation between distinct data classes. On the other hand, it minimizes the variance of data within classes to prevent data points of different classes overlapping over one another for better classification accuracy. Figure 6 shows a schematic depiction of discriminant classification.

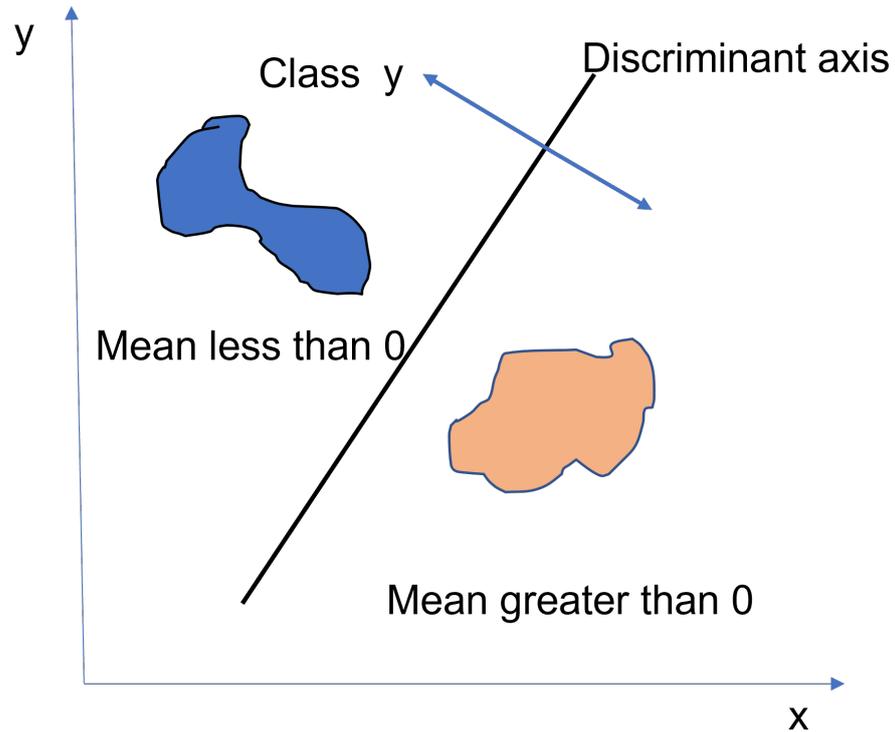


Figure 6. Discriminant Classification.

The primary difference between the linear and quadratic discriminant analyses is that the former assumes the covariance and mean of all classes to be equal, whereas the latter computes it separately for each class. The linear score function of the discriminant is given by:

$$\delta_k(x) = x^T \sum_k^{-1} m_k - \frac{1}{2} m_k^T \sum_k^{-1} m_k + \log p_{i_k} \tag{7}$$

The quadratic discriminant function is given as follows:

$$\delta_k(x) = -\frac{1}{2} \log \left| \sum_k \right| - \frac{1}{2} (x - \mu_k)^T \sum_k^{-1} (x - \mu_k) + \log \pi_k \tag{8}$$

where m_k is the class average, x represents the inputs, p_{i_k} is the prior probability of a class, and μ_k represents the sample mean of class k .

Table 7 shows the classification hyperparameters of the discriminant models for LM soft sensing. The covariance structure was selected to be full, in order to account for the interrelationships and correlations of one flexibility parameter with all others (including LM).

Table 7. Discriminant classification hyperparameters.

Model Hyperparameters	Linear Discriminant	Quadratic Discriminant
Preset Discriminant	Linear	Quadratic
Covariance Structure	Full	Full

2.2.5. Naive Bayes

Naive Bayes algorithms are probabilistic classifiers based on the application of Bayes’ theorem for supervised learning. These algorithms simply assume conditional independence between individual dataset features. Simply put, the presence or absence of any particular feature is assumed to be unrelated to that of another feature. This assumption decouples the conditional class-based feature distributions, making it possible to estimate each class feature as a stand-alone single-dimensional distribution (Figure 7).

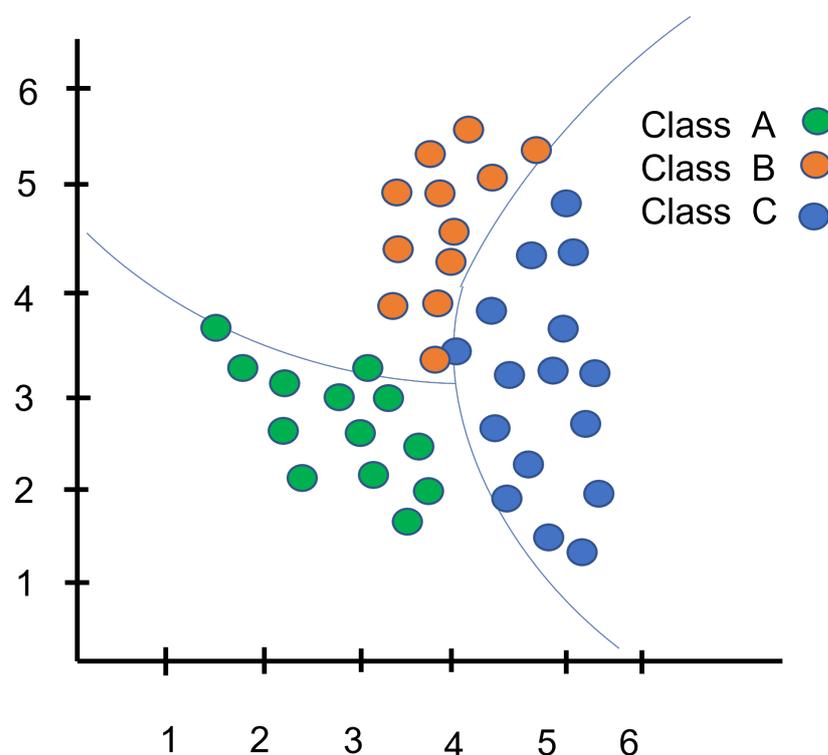


Figure 7. Naive Bayes Classification.

However, these algorithms are called “naive” due to their overly simplified assumptions. Naive Bayes classifiers can be extremely fast as compared to other more complex methods, since they need not determine the entire covariance matrix but only the variances of each label of dataset variables. Moreover, these algorithms do not require large datasets for learning, and have proved to work well in spam filtering and document classification real-world applications. The Bayes’s probabilistic classification theory can be expressed as follows:

$$P(y/X) = \frac{P(X/y)P(y)}{P(X)} \tag{9}$$

where y is the class variable and X represents the respective class parameters/features.

The major drawback of these methods is their prediction accuracy, which is generally not up to the mark. The Gaussian Naive Bayes algorithm is based on the Gaussian distributions whereas Kernel Naive Bayes algorithms rely on the Kernel weighting functions. These functions are feature density estimators and are non-parametric in their constitution. Hence, Kernel functions do not depend on a fixed structure as is the case with parametric

estimators wherein the functional parameters are optimized and stored. Kernel functions directly utilize data points for arriving at predictions/estimations without following any pre-fixed functional structures. Table 8 shows the classification hyperparameters for Gaussian and Kernel Naive Bayes models. Categorical predictors were not needed for the dataset used in the present work, since the input variables (manufacturing flexibilities) did not have discrete categorical values (labels). Gaussian kernels were employed for minimal estimation errors and optimal classifications in Kernel Naive Bayes. Unbounded kernel smoothing support was set for the same. Separate kernels or kernel-smoothing density supports were not applicable in case of the Gaussian Naive Bayes architecture.

Table 8. Naive Bayes classification hyperparameters.

Model Hyperparameters	Gaussian Naive Bayes	Kernel Naive Bayes
Preset Naive Bayes	Gaussian	Kernel
Distribution name for numeric predictors	Gaussian	Kernel
Distribution name for categorical predictors	NA	NA
Kernel type	NA	Gaussian
Support	NA	Unbounded

2.2.6. Support Vector Machine (SVM)

SVM is a popular regression and classification supervised learning algorithm that segregates multi-dimensional data classes using a hyperplane. SVM creates and uses a hyperplane as a decision boundary to be able to classify a future data point into its correct category. The extreme cases of each data class (known as support vectors) help in defining the class boundary, or the hyperplane. The support vectors of respective data classes lie closest to the hyperplane. The SVM algorithm explores a number of hyperplane solutions and selects the best class boundary that maximizes the separation margin between classes. SVM achieves this by attaching a penalty to every point that lies on the other side of its class across a particular hyperplane. The hyperplane dimensions are dependent on the number of data class features. For instance, for a dataset having only two features, the hyperplane takes the form of a line (Figure 8), whereas for a dataset with three features, the hyperplane takes the form of a plane to segregate the classes.

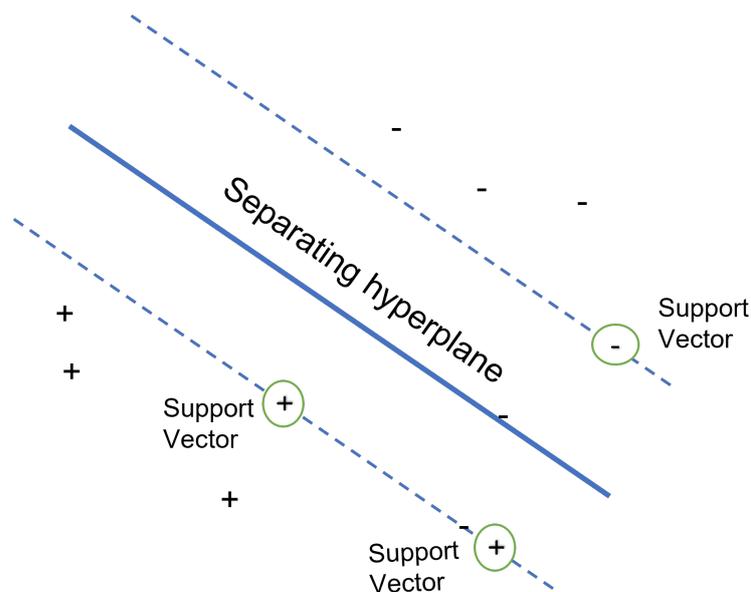


Figure 8. SVM Classification.

Hence, linear SVM is useful for a dataset having linearly separable entities, whereas a nonlinear SVM classifier is required for datasets wherein linear separation is not possible.

SVM is more effective in solving higher-dimensional feature spaces, as they are more separable into distinct class label regions across the hyperplane. Similarly, SVM works better in datasets that have wider separation margins between data classes. This implies that SVM is more effective in datasets wherein the number of samples is lesser than the number of dimensions. Moreover, since it uses only a small part of the entire dataset, namely the support vectors, SVM is highly memory-efficient. Conversely, SVM proves relatively ineffective in larger datasets, or when multiple data classes are overlapping, and/or when the total number of data samples is lesser than the number of features per data point. The SVM loss function is defined as follows:

$$\min_w \lambda \|w\|^2 + \sum_{i=1}^n (1 - y_i \langle x_i, w \rangle)_+ \tag{10}$$

where x , y , λ , and w represent the input parameters, class output, Kernel scale, and the regularisation parameters, respectively.

Table 9 shows the classification hyperparameters for the various SVM models explored in the current study. Each SVM model employed a kernel function suitable to its configuration. Input data were scaled automatically (before input to the kernel function) by the Matlab software for linear, quadratic, and cubic SVMs. Progressively higher kernel input scales were employed for fine, medium, and coarse Gaussian SVMs. A box constraint level of one was defined for all SVMs, ensuring a high missclassification cost and a stricter data classification. The one-versus-one methodology enables SVM, which is a binary classifier, to execute multi-class classification. One-versus-one splits the multi class dataset into multiple binary classification problems. All input data were standardized and scaled to a standard normal distribution.

Table 9. Support Vector Machine (SVM) classification hyperparameters.

Model Hyperparameters	Linear SVM	Quadratic SVM	Cubic SVM	Fine Gaussian SVM	Medium Gaussian SVM	Coarse Gaussian SVM
Preset SVM	Linear	Quadratic	Cubic	Fine Gaussian	Medium Gaussian	Coarse Gaussian
Kernel Function	Linear	Quadratic	Cubic	Gaussian	Gaussian	Gaussian
Kernel Scale	Automatic	Automatic	Automatic	0.61	2.4	9.8
Box Constraint Level	1	1	1	1	1	1
Multiclass Method	One-vs-One	One-vs-One	One-vs-One	One-vs-One	One-vs-One	One-vs-One
Standardized data	True	True	True	True	True	True

2.2.7. Ensemble Models

Ensemble machine learning models aim to achieve better prediction accuracy by strategically combining the estimations of multiple models on the same dataset. The bootstrap aggregation method (bagging) is one of the prominent ensemble techniques wherein multiple decision trees are fitted on different parts of the same dataset and the final prediction is computed as an average of all tree estimations.

Boosting is another significant ensemble method that involves sequential application of multiple models that build upon the estimations of previous models. The successive ensemble models attempt to further boost the estimation accuracy achieved by the previous model, thus ensuring continuous correction of prediction errors. In other words, boosting ensemble iteratively moves models to focus on more and more data samples that are hard to predict. The final estimation of the boosted ensemble learning is computed as the weighted average of all model predictions. Random undersampling boosting (RUSBoost) is a boosting ensemble method that is especially adept at handling datasets that are imbalanced, having significantly asymmetric/non-uniform number of data samples across classes. Adaptive Boosting (AdaBoost), on the other hand, assigns weights to each data point to improve overall estimation, continuously adapting higher weights to the data instances that are difficult to classify.

Subspace ensembles are yet another category of ensemble models in which a model is fitted on different subsets of features of the same dataset. In other words, the same model

architecture is fitted on different groups of input data features from the same training dataset. The final prediction of the subspace ensemble is the mode (for classification) or mean (for regression) of the class labels estimated by the model applied on different data feature subsets. Table 10 shows the classification hyperparameters for the various ensemble models explored in this work. The type of learning methodology of each model varied as per its own unique ensemble method and preset. NA indicates the hyperparameters that were not applicable to the specific models. For instance, subspace dimensions are only applicable to subspace ensembles, whereas the number of predictors, maximum number of splits, and learning rates are only applicable to ensemble trees. The next section presents and discusses the LM soft sensing results obtained from all of the above-mentioned models.

Table 10. Ensemble classification hyperparameters.

Model Hyperparameters	Ensemble Boosted Tree	Ensemble Bagged Tree	Ensemble Subspace Discriminant	Ensemble Subspace KNN	Ensemble RUSBoosted Trees
Preset	Boosted Tree	Bagged Trees	Subspace Discriminant	Subspace KNN	RUSBoosted Trees
Ensemble Method	AdaBoost	Bag	Subspace Discriminant	Subspace KNN	RUSBoost
Learner type	Decision Tree	Decision Tree	Subspace Discriminant	Nearest Neighbors	Decision Tree
Maximum number of splits	20	40	NA	NA	20
Number of learners	30	30	30	30	30
Learning rate	0.1	NA	NA	NA	0.1
Number of predictors to sample	Select All	Select All	NA	NA	Select All
Subspace dimension	NA	NA	3	3	NA

3. Results and Discussion

The present study implemented seven machine learning architectures (decision trees, ensembles, K-nearest neighbours, discriminants, support vector machine, artificial neural networks, and Naive Bayes) to classify 46 auto manufacturing firms into high, medium or low lean attainment categories based on the actual levels of their respective product, labour, machine, volume, routing, and material handling flexibilities. The following subsections firstly discuss the soft sensing performances of 29 models belonging to the above-mentioned machine learning architectures. Thereafter, the best model from each machine learning method/group is identified. From this list of group-wise best models, the topmost performing model is selected for a subsequent detailed analysis of its LM classification results.

3.1. ANN

Table 11 depicts the results of the neural network group of models. The medium network achieved the highest validation accuracy of 70%. However, this model attained the lowest testing accuracy of 40%, along with the bilayered network. The highest testing accuracy of 80% was achieved by the trilayered network. The wide and narrow networks attained the second highest testing accuracy of 60%, consuming a training time of 0.71 and 1.01 s, respectively. The trilayered and medium networks consumed the maximum and minimum training times of 1.02 and 0.54 s, respectively. These results show that a higher number of neurons does not necessarily guarantee better testing accuracies, but a higher number of layers may improve estimations. Hence, among ANN models, the trilayered neural network provides the best lean level estimation (based on testing accuracy) of an automobile parts manufacturing organization based on the levels of its manufacturing flexibilities.

Table 11. ANN Model Results.

Sr. No.	Model Type	Number of Fully Connected Layers	First Layer Size	Validation	Testing	Training Time (s)
1	Narrow Neural Network	1	10	30	60	1.01
2	Medium Neural Network	1	25	70	40	0.54
3	Wide Neural Network	1	100	40	60	0.71
4	Bilayered Neural Network	2	10	60	40	0.75
5	Trilayered Neural Network	3	10	30	80	1.02

3.2. KNN

Table 12 shows estimation results of the KNN family of models. In this group, the Fine KNN achieved the highest validation accuracy of 80%, whereas all other KNN models attained 60% validation accuracies. However, Fine KNN obtained a very low testing accuracy of only 20%, whereas all other KNN models could attain slightly better testing accuracies of 40%. Fine KNN performed relatively the worst in terms of training time consumption as well, taking up to 0.95 s, whereas coarse KNN consumed the training time of 0.51 s, the least among all KNN models. The testing accuracy results indicate that the KNN family of models were not suitable to reliably predict lean manufacturing levels of auto firms based on their flexibilities as per the dataset used in the current study.

Table 12. KNN Model Results.

Sr. No.	Model Type	Number of Neighbors (Default)	Distance Metric	Distance Weight	Validation	Testing	Training Time (s)
1	Fine KNN	1	Euclidean	Equal	80	20	0.95
2	Medium KNN	10	Euclidean	Equal	60	40	0.83
3	Coarse KNN	100	Euclidean	Equal	60	40	0.51
4	Cosine KNN	10	Cosine	Equal	60	40	0.65
5	Cubic KNN	10	Minkowski	Equal	60	40	0.63
6	Weighted KNN	10	Euclidean	Squared Inverse	60	40	0.58

3.3. Tree

Table 13 displays the tree model family results. This group is composed on fine, medium, and coarse trees with 100, 20, and 4 splits, respectively. The validation and training accuracies of all these models were attained at 70% and 60%, respectively. The least training time was consumed by the coarse tree (0.76 s) whereas the most training time was taken by the fine tree (2.62 s), indicating that more training time is required by trees having higher number of branch splits. The testing accuracy results indicate that the tree models attained only moderate accuracies while estimating lean levels based on the respective firms' flexibilities.

Table 13. Tree Model Results.

Sr. No.	Model Type	Maximum Number of Splits	Validation	Testing	Training Time (s)
1	Fine Tree	100	70	60	2.62
2	Medium Tree	20	70	60	0.83
3	Coarse Tree	4	70	60	0.76

3.4. Discriminants

Table 14 shows results of the discriminant models. This group consisted of only two types—the linear and the quadratic discriminants having linear and quadratic presets. Herein, the linear model attained validation and testing accuracies of 60% and 40%, respectively, consuming a training time of 0.93 s. The testing accuracy results show that the quadratic model failed to model the lean-flexibility dataset considered in the present work.

Table 14. Discriminant Model Results

Sr. No.	Model Type	Preset	Validation	Testing	Training Time (s)
1	Linear Discriminant	Linear Discriminant	60	40	0.93
2	Quadratic Discriminant	Quadratic Discriminant	Fail	Fail	NA

3.5. Naive Bayes

Table 15 shows the performance results of the Naive models. This model group consisted of the Gaussian and Kernel Naive Bayes configurations having Gaussian and

Kernel distributions for their respective numeric predictors. Both configurations attained only 40% testing accuracies. As regards validation, the Gaussian and Kernel Naive Bayes models achieved 70% and 60% accuracies, consuming 0.86 and 1.04 s of training times, respectively. The testing accuracy results indicate that the Naive Bayes models were not suitable for estimating lean levels of firms based on their flexibility parameters.

Table 15. Naive Bayes Model Results.

Sr. No.	Model Type	Distribution Name for Numeric Predictors	Validation	Testing	Training Time (s)
1	Gaussian Naive Bayes	Gaussian	70	40	0.86
2	Kernel Naive Bayes	Kernel	60	40	1.04

3.6. SVM

Table 16 displays the performance results of the SVM group of models explored in the current study. All SVM models attained 60% validation accuracies, except for quadratic and cubic SVMs which achieved relatively inferior validation accuracies of 50% and 40%, respectively. Similarly, all SVM configurations achieved relatively inferior testing accuracies of 40%, whereas the quadratic and cubic SVMs performed better with 60% testing accuracies each. However, quadratic SVM consumed the maximum and very high training time of 4.47 s! The cubic SVM also required a relatively higher training time of 1.58 s. In general, the linear, quadratic, and cubic SVMs took more time to train as compared to their Gaussian counterparts. The coarse Gaussian SVM took the least time to train (0.51 s). The testing accuracy results show that quadratic and cubic SVM models performed moderately well in terms of lean prediction based on manufacturing flexibilities of auto parts producing firms.

Table 16. SVM Model Results.

Sr. No.	Model Type	Kernel Function	Kernel Scale	Validation	Testing	Training Time (s)
1	Linear SVM	Linear	Automatic	60	40	1.62
2	Quadratic SVM	Quadratic	Automatic	50	60	4.47
3	Cubic SVM	Cubic	Automatic	40	60	1.58
4	Fine Gaussian SVM	Gaussian	0.61	60	40	0.61
5	Medium Gaussian SVM	Gaussian	2.4	60	40	0.53
6	Coarse Gaussian SVM	Gaussian	9,8	60	40	0.51

3.7. Ensemble

Table 17 shows the results of the ensemble group of model trees. This group consisted of ensemble boosted, bagged, subspace discriminant, subspace KNN, and the RUSboosted trees. The ensemble methods adopted in these trees were AdaBoost, bag, subspace, subspace, and RUSbooster, respectively. Decision trees were selected as the learner types in the ensemble boosted, bagged, and RUSboosted trees. On the other hand, discriminant and nearest neighbour-based learning were adopted in the ensemble subspace discriminant and KNN trees, respectively. The RUSboosted ensemble tree attained the best validation accuracy of 80%, followed by boosted and subspace (discriminant and KNN) trees with 60% validation accuracy each. The ensemble-bagged tree achieved relatively lowest validation accuracy of 50%. In general, the ensemble models performance was quite inferior in terms of the testing accuracies. Only the RUSboosted tree attained a moderately better testing accuracy of 60%, whereas all others achieved 40% or lesser. The ensemble subspace KNN model achieved the lowest testing accuracy of only 20%. All ensemble models consumed more than 1 s of training time, which is quite higher than that generally taken by the models of other families discussed in the previous sections. The bagged ensemble tree needed the maximum time to train (1.64 s), whereas the ensemble subspace KNN tree trained in the relatively shortest time span of 1.08 seconds. Hence, the testing accuracy results indicate that among the ensemble model family, the RUSboosted tree can be selected as the best performing model to predict the lean level of an organization based the flexibility factors.

Table 17. Ensemble Model Results.

Sr. No.	Model Type	Ensemble Method	Learner Type	Validation	Testing	Training Time (s)
1	Ensemble Boosted Tree	AdaBoost	Decision Tree	60	40	1.23
2	Ensemble Bagged Tree	Bag	Decision Tree	50	40	1.64
3	Ensemble Subspace Discriminant	Subspace	Discriminant	60	40	1.5
4	Ensemble Subspace KNN	Subspace	Nearest Neighbors	60	20	1.08
5	Ensemble RUSBoosted Tree	RUSBoost	Decision Tree	80	60	1.21

3.8. Top Seven Models

Table 18 displays the list of best performing architectures from all model families. While selecting the best performing model from a group, preference was given to testing accuracy followed by the validation accuracy, and training time. Hence, in case of the KNN model group, five models had the same validation and testing accuracies. Among them, Table 18 lists coarse KNN as the best performer owing to its least training time. Similarly, the coarse tree was selected as the best performing tree model owing to its lowest training time, other performance parameters being equal. In case of the SVM model family, both quadratic and cubic architectures were tied at maximum testing accuracy of 60%. Among them, quadratic SVM was chosen as the better performer based on its superior validation accuracy. In cases wherein training times are more important, cubic SVMs may be selected as they took much lesser time to train as compared with their quadratic counterpart. Overall, it is evident from Table 18 that the trilayered neural network attained the best testing accuracy of 80% among all others. The trilayered neural network consumed relatively higher training time (1.02 s), with only the ensemble RUSBoosted trees taking higher time than this model (1.21 s). A detailed result analysis of the best performing model is presented in the next subsection.

Table 18. Top Seven Models: Result.

Sr. No.	Model Type	Testing	Validation	Training Time (s)
1	Trilayered Neural Network	80	30	1.02
2	Ensemble RUSBoosted Trees	60	80	1.21
3	Coarse Tree	60	70	0.76
4	Quadratic SVM	60	50	4.47
5	Gaussian Naive Bayes	40	70	0.86
6	Coarse KNN	40	60	0.51
7	Linear Discriminant	40	60	0.93

3.9. Result Analyses of the Best Performing Model

In the previous subsection, it was presented that the trilayered neural network obtained the best testing accuracy of 80% for the LM soft sensor application in the current study. This section presents analyses of scatter plots, confusion matrices, receiver operating characteristic (ROC) curves, and the parallel coordinate plot for the trilayered neural network LM soft sensor.

3.9.1. Scatter Plots

Scatter plots are used to plot the correct and/or incorrect predictions of a classifier model with respect to a pair of dataset features expressed along the X and Y axes of the plot. The dataset used in current study included six data features (flexibilities), hence at least five scatter plots were required to depict all six features (flexibilities) of the companies. Figures 9–13 show the validation scatter plots of the high, medium, and low LM classifications predicted for different companies by the trilayered neural network soft sensor. In this figure, dots and crosses indicate the accurate and inaccurate validation predictions made by the trilayered neural network LM soft sensor, respectively. The X and Y positions of each dot/cross indicate the corresponding flexibility levels of the respective enterprise,

whereas the colours of the dots/crosses indicate the respective LM levels predicted by the trilayered network—high (blue), medium (yellow), and low (red).

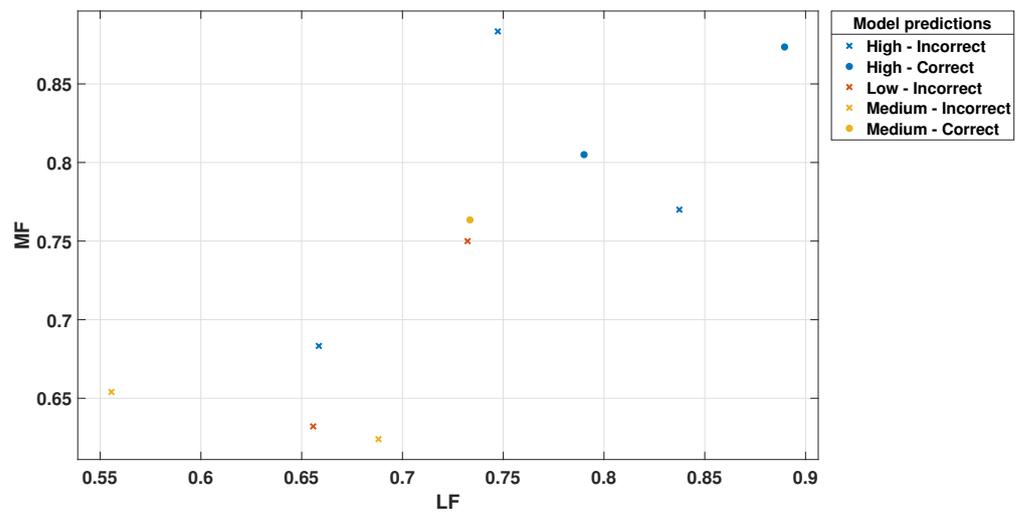


Figure 9. LF Scatter Plot for Trilayered Neural Network.

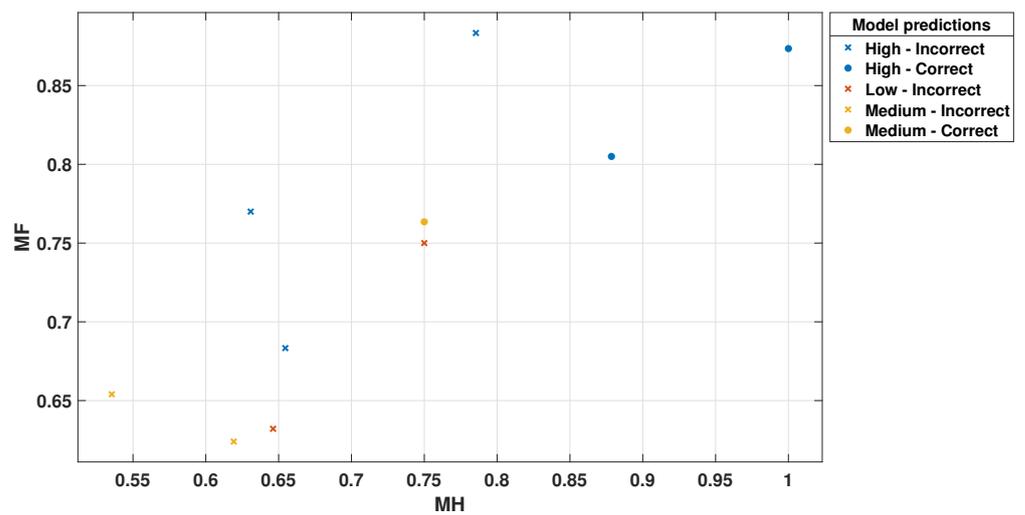


Figure 10. MH Scatter Plot for Trilayered Neural Network.

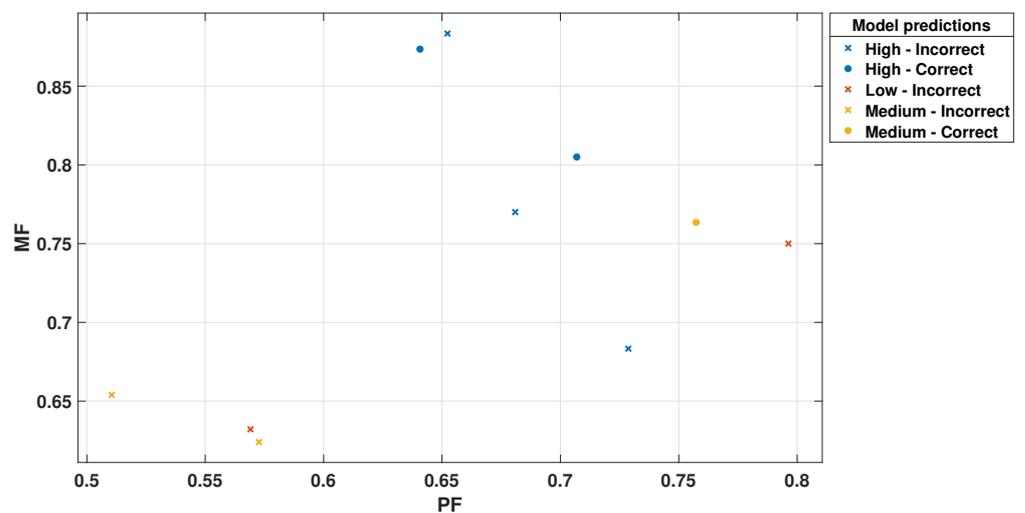


Figure 11. PF Scatter Plot for Trilayered Neural Network.

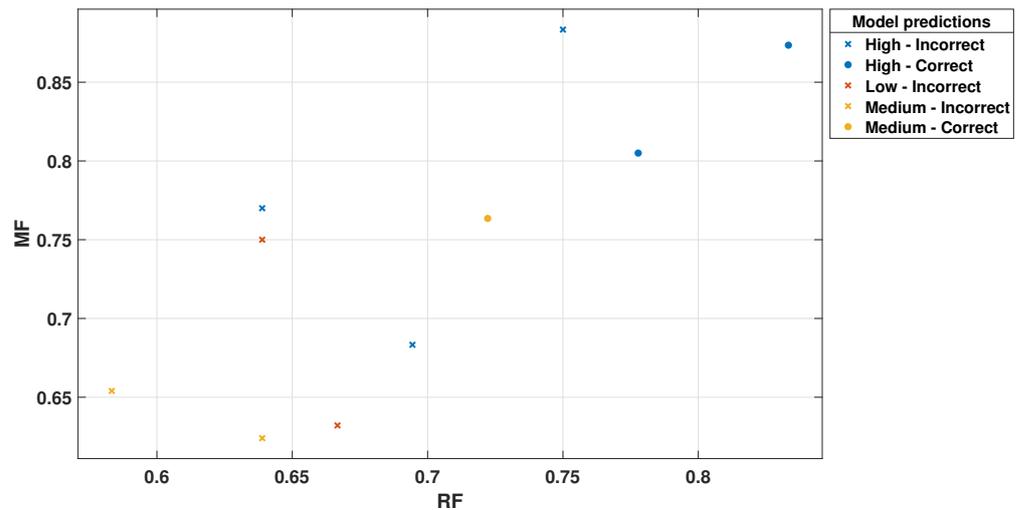


Figure 12. RF Scatter Plot for Trilayered Neural Network.

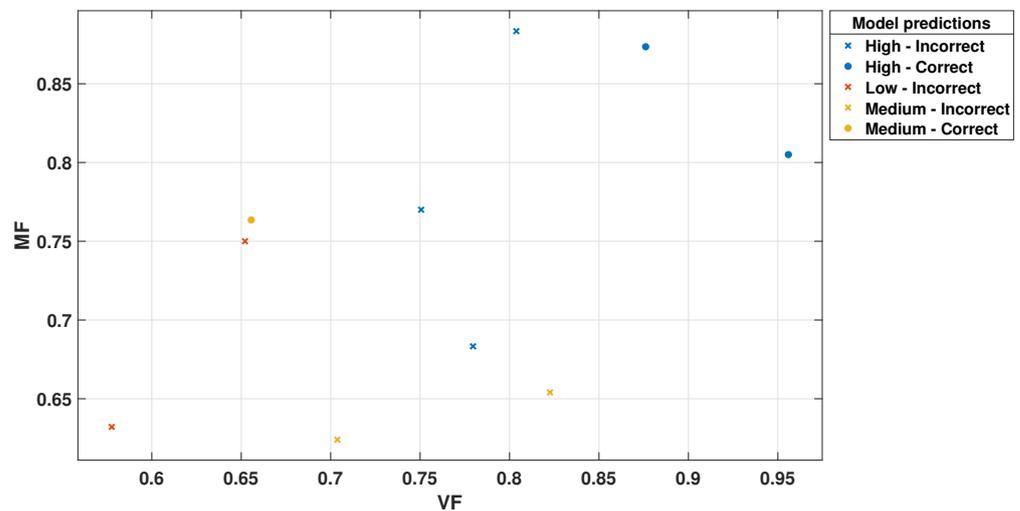


Figure 13. VF Scatter Plot for Trilayered Neural Network.

These figures show that out of the ten companies included in the validation dataset partition, only three were correctly classified as per their true LM ranges. Two of these correct classifications belonged to the high LM class companies (blue dots), whereas the third correct prediction belonged to a medium LM class firm (yellow dot). The remaining seven firms were misclassified by the LM soft sensor during validation. This result substantiates the low validation accuracy of the trilayered neural network LM soft sensor. Considering the correctly classified high LM class companies (blue dots), the machine (MF) and labour (LF) flexibilities of both these companies are above 0.80 and 0.75, respectively (Figure 9). The material handling (MH), routing (RF), and volume (VF) flexibilities of these companies are above 0.85 (Figure 10), 0.75 (Figure 12), and 0.85 (Figure 13), respectively. However, the product flexibility (PF) of one these enterprises is between 0.7 to 0.75 and for the other is between 0.6 and 0.65 (Figure 11). The company having lower PF is having very high MF (>0.85), LF (>0.85), MH (1), RF (>0.8), and VF (>0.95). This analysis shows that a company may be able to compensate for its lower product flexibility by having exceptionally high performances in the remaining flexibilities and attain an overall superior LM level. On the other hand, the correctly classified medium LM class firm (yellow dot) obtained MF, LF, MH, PF, and RF in the ranges 0.75 to 0.8, 0.7 to 0.75, 0.75, 0.75 to 0.8, and 0.7 to 0.75, respectively. This implies that the medium LM class firm scored better than its high LM counterparts only in case of PF.

3.9.2. Confusion Matrices

Figure 14 shows the confusion matrix for the validation results of the trilayered neural network. In this figure, true classes are represented in rows whereas the predicted classifications are shown in columns. The number of observations indicates the number of companies in each cell. Blue color-filled cells indicate the cases wherein the model predictions match the true classes. For instance, there are 2 companies in the first row and first column, indicating that the true class as well as the predicted class of these companies is high LM. Similarly, there is one company shown in the third row and third column, indicating that the true class as well as the predicted class of this company is medium LM. All other predicted classifications are incorrect, as they do not match with the corresponding true classes. For instance, 5 of the 6 companies that are medium LM in reality have been misclassified by the model—3 as high and 2 as low class. Moreover, two companies having true classes low and high were misclassified by the model as medium. Overall, the validation confusion matrix of the trilayered neural network depicts a poor classification performance, as corroborated by the associated accuracy of only 30% (Table 18).

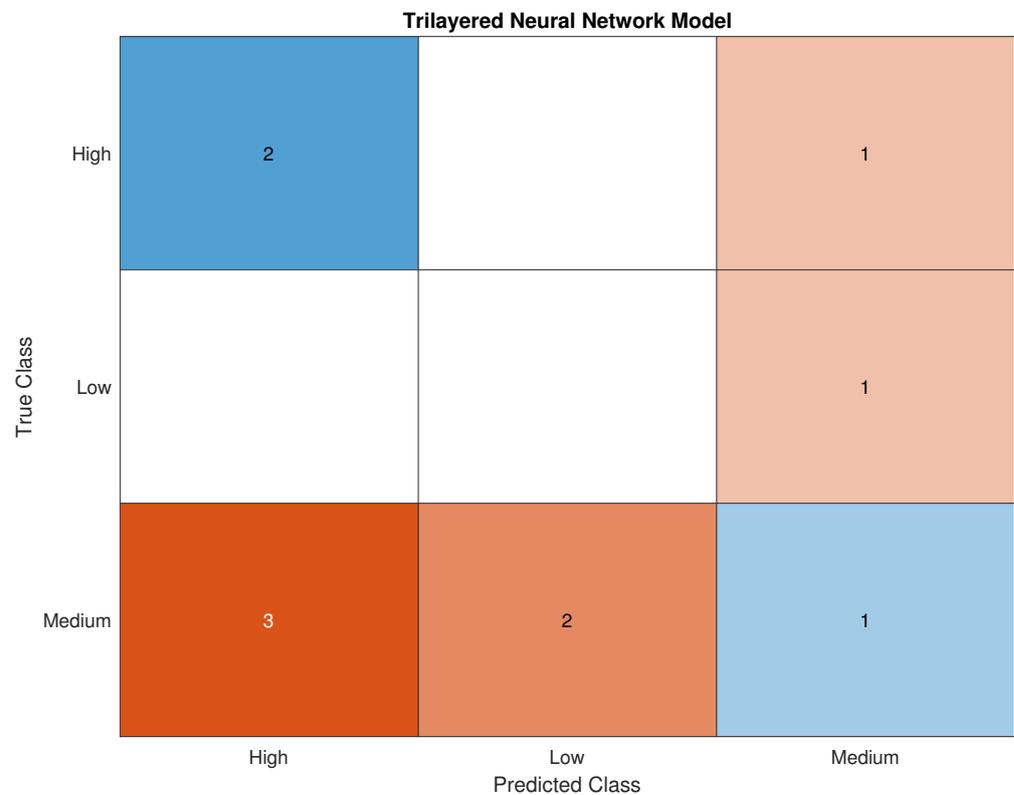


Figure 14. Validation Confusion Matrix for Trilayered Neural Network.

Figure 15 shows the confusion matrix for the testing results. In this figure, four out of five observations (companies) lie in blue colored cells, indicating high testing accuracy of the trilayered network LM soft sensor. Only one observation having high LM true class was misidentified by the model to belong to the medium class. Hence, the testing confusion matrix of the trilayered neural network depicts a superior classification performance, as corroborated by the associated accuracy of 80% (Table 18).

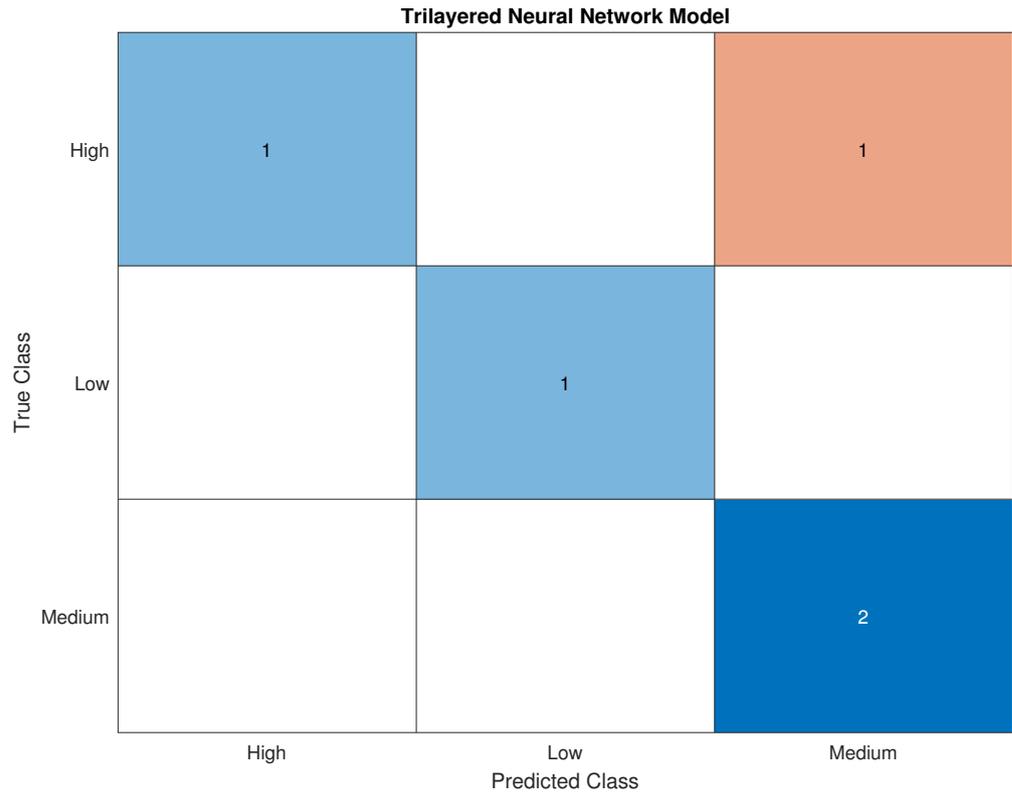


Figure 15. Testing Confusion Matrix for Trilayered Neural Network.

The testing confusion matrix (Figure 15) was further analysed to yield the precision and recall metrics of all classes. For instance, the true positives (TP) for the high, medium, and low classes are 1, 2, 1 and false positives (FP) are 0, 1, and 0, respectively. The false negatives (FN) for these classes (in the same order) are 1, 0, and 0, respectively. Model precision is defined as a measure of prediction accuracy based on the total predicted samples, determined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

Whereas, recall indicates the prediction accuracy based on the actual number of samples, given as:

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

F1 scores, on the other hand, are harmonic means of recall and precision values. F1 scores give a balanced measure of both metrics by considering both false negatives as well as false positives:

$$F1\ Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \tag{13}$$

Table 19 depicts the precision, recall, and F1 scores of each of the three LM classes based on the respective TP, FP, and FN values obtained during testing of the trilayered neural network.

Table 19. Precision, recall, and F1 scores of the trilayered neural network soft sensor (testing results).

		Prediction Metrics		
		Precision	Recall	F1 Score
LM Classes	High	1	0.5	0.67
	Low	1	1	1
	Medium	0.67	1	0.80

This result confirms that the trilayered neural network soft sensor is insensitive/robust to the relatively unbalanced dataset used in the present study. As per Sun et al. [61], imbalanced class distribution deteriorates classification performance. The authors state that even the small class to dominant class sample size ratio of 1:10 may prove difficult to be successfully classified. Secondly, the authors state that class separability is one of the primary contributors to unsuccessful classification of small classes due to the presence of overlapping patterns across classes. Thirdly, the authors discuss that the networks trained on unbalanced datasets are likely to misclassify small classes due to the inadequate representation of the small class dependency patterns in the network weights. In the present work, 5 of the 46 surveyed firms belonged to the low LM class, whereas 27 and 14 companies belonged to the medium and high LM classes, respectively. Hence, the training dataset in the present study did involve unbalanced class sizes. However, the ratio of the lowest (low LM) to the highest (medium LM) sized classes was less than 1:6, indicating that the level of imbalance among the class sizes was not severe. Secondly, the distinct ranges of LM classes shown in Table 2 ensured class separability and ruled out the possibility of any overlapping class patterns. Thirdly, the high precision and recall values of the low LM class (Table 19) confirm that the trilayered neural network soft sensor correctly classified the lowest-sized LM class present in the testing dataset.

3.9.3. ROC Curves

ROC curve shows the true positive rate (TPR) versus false positive rate (FPR) performance of a model classifier for a particular class. TPR and FPR indicate the percentages of true and false positives (respectively) from all the positives predicted for a class by a model. Area under the curve (AUC) indicates the integral of TPR over FPR 0 to 1. Figures 16–18 show validation ROC curves for high, medium, and low LM classes, respectively. AUC for these classes is 0.71, 0.54, and 0.44, respectively, indicating highest validation accuracy for high LM class followed by the medium and low LM classes. The red dot in the ROC curve indicates the optimal operating point of the model for a particular class that aims to minimize the cost of false positives and the cost of misidentifying true positives. The validation ROC optimal operating point for the high LM class was obtained at 0.43 FPR and 0.67 TPR. The same was obtained for the medium LM class at 0.50 FPR and 0.17 TPR; whereas it was obtained for the low LM class at 0.22 FPR and 0.00 TPR. Hence, the optimal operating points for validation ROC curves were obtained at the ranges of low to moderate FPR and very low to moderately high TPR values.

Figures 19–21 show testing ROC curves for high, medium, and low LM classes, respectively. AUC for these classes is 0.58, 0.75, and 1.00, respectively, indicating highest testing accuracy for low LM class followed by the medium and high LM classes. High AUC for the low LM class also affirms the robustness of the trilayered neural network soft sensor towards the unbalanced dataset used in the present study, indicating successful classification of the low sample-sized class (low LM) during testing. The testing AUC for high LM class is inferior to the corresponding validation AUC by a margin of 0.13. However, the testing AUCs for medium and low LM classes are superior to the corresponding validation AUCs of the same classes by margins of 0.21 and 0.576, respectively. Hence, the testing accuracy of the trilayered network LM soft sensor is much higher than its validation accuracy. The testing ROC optimal operating point for the high LM class was obtained at 0.00 FPR and 0.50 TPR. The same was obtained for the medium LM class at 0.33 FPR and

1.00 TPR; whereas it was obtained for the low LM class at 0.00 FPR and 1.00 TPR. Hence, the optimal operating points for testing ROC curves were obtained ranging from very low to low FPR and from high to very high TPR values.

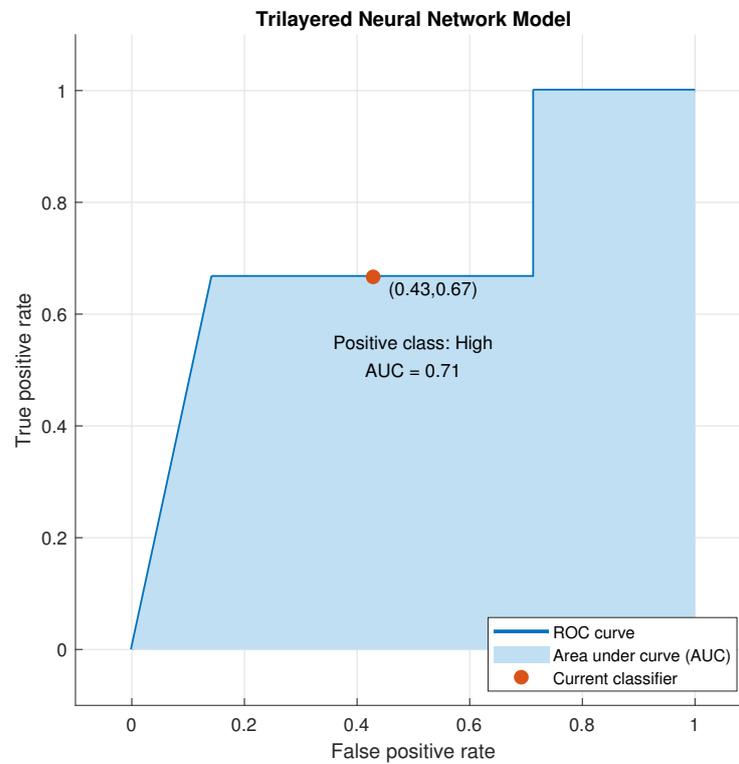


Figure 16. Validation ROC for Trilayered Neural Network for High Class.

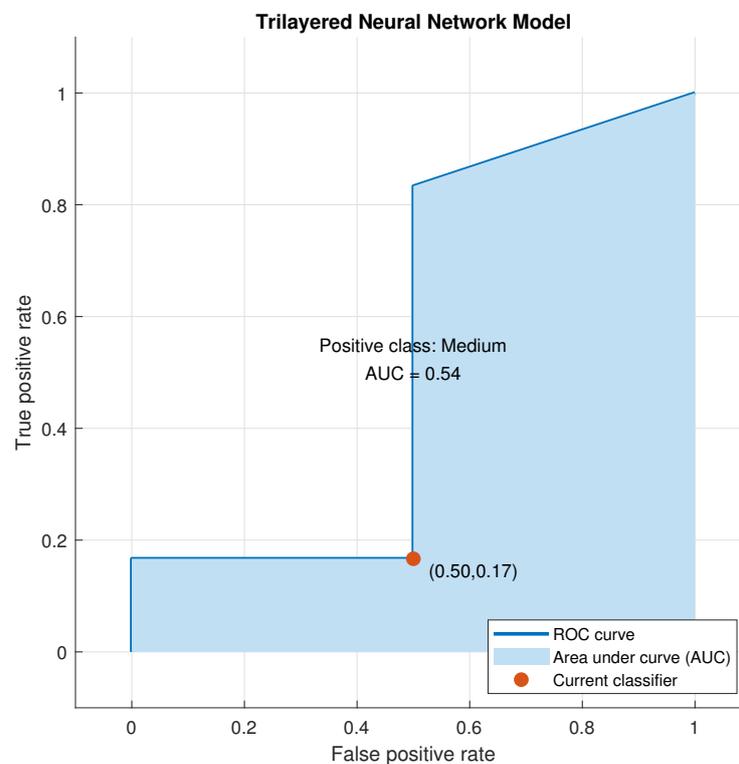


Figure 17. Validation ROC for Trilayered Neural Network for Medium Class.

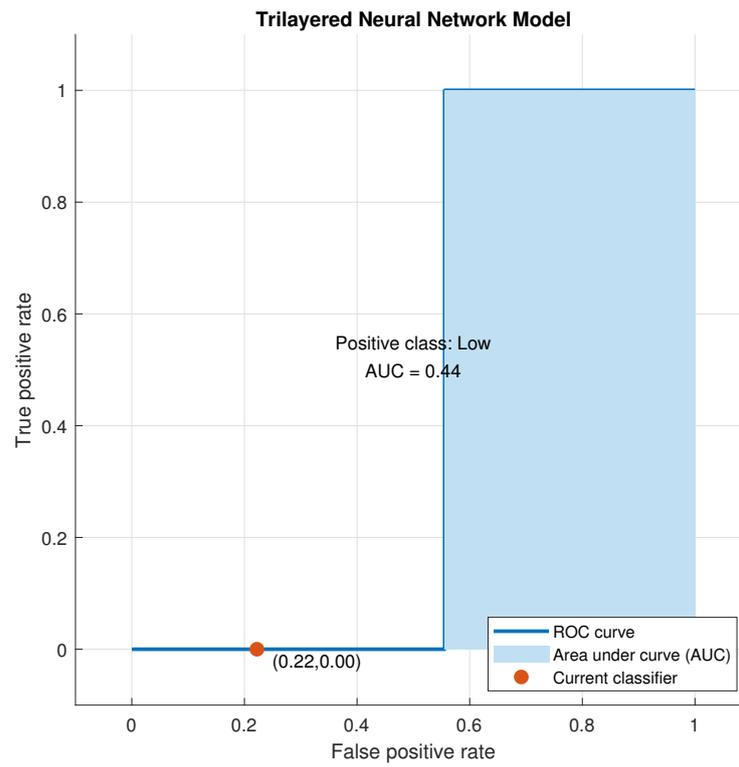


Figure 18. Validation ROC for Trilayered Neural Network for Low Class.

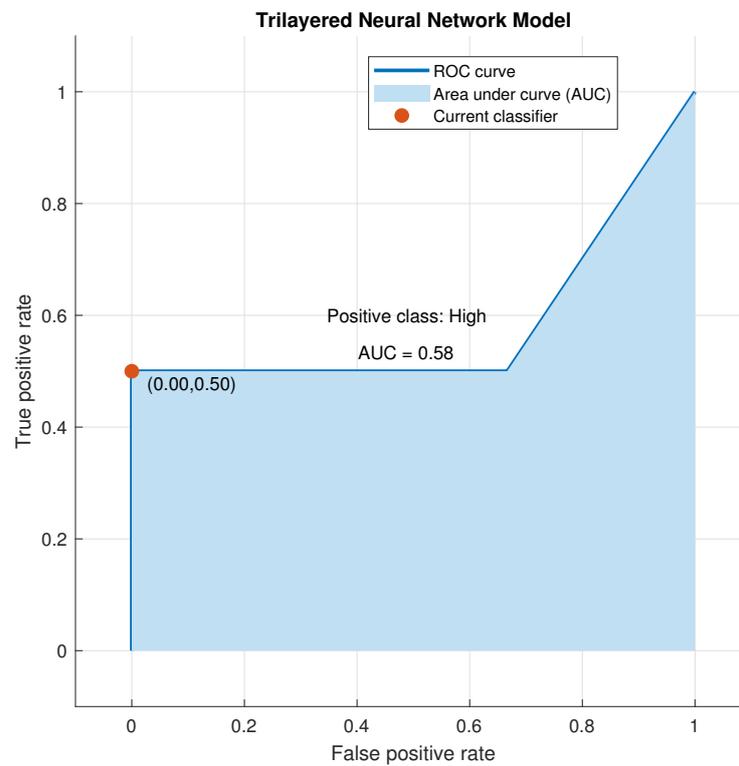


Figure 19. Testing ROC for Trilayered Neural Network for High Class.

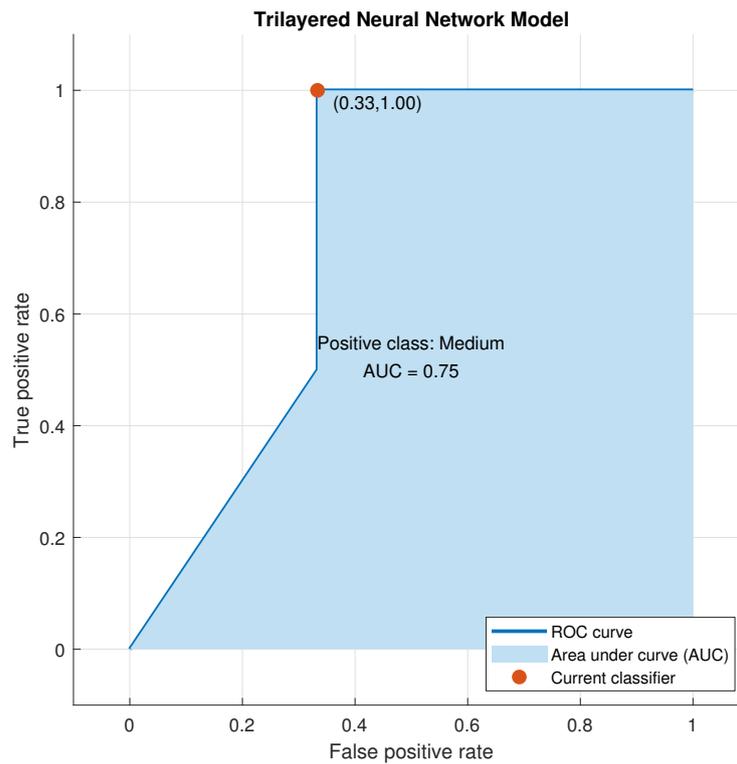


Figure 20. Testing ROC for Trilayered Neural Network for Medium Class.

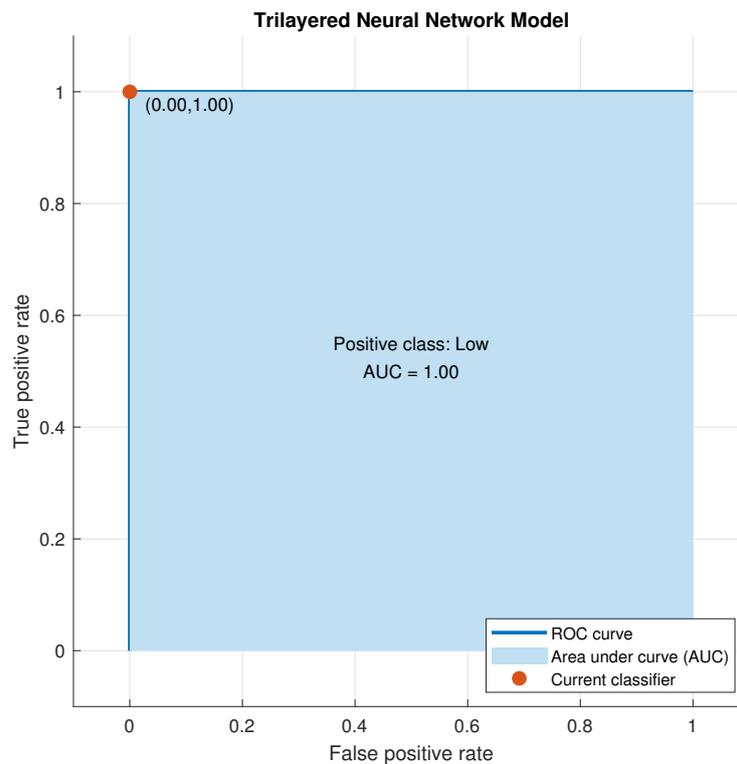


Figure 21. Testing ROC for Trilayered Neural Network for Low Class.

3.9.4. Parallel Coordinates Plot

Parallel coordinates plots are typically used to visualize high-dimensional data in a two-dimensional representation. These plots help in easy identification of trends over multiple data dimensions or features. Figure 22 shows the parallel plot for the validation predictions of the trilayered neural network LM soft sensor. In this figure, the six data

features (manufacturing flexibilities) are depicted on the X axis and their levels are marked on the respective Y axes. The figure shows trendlines belonging to ten auto component manufacturers depicted in different colors corresponding to the respective prediction labels. The three high and one medium correctly predicted LM class companies are depicted using blue and yellow unbroken lines. The broken/dashed lines indicate flexibilities of the misclassified firms. This figure also shows that the correctly predicted high LM class firms have higher flexibilities than their medium LM class counterpart, except in case of the product flexibility.

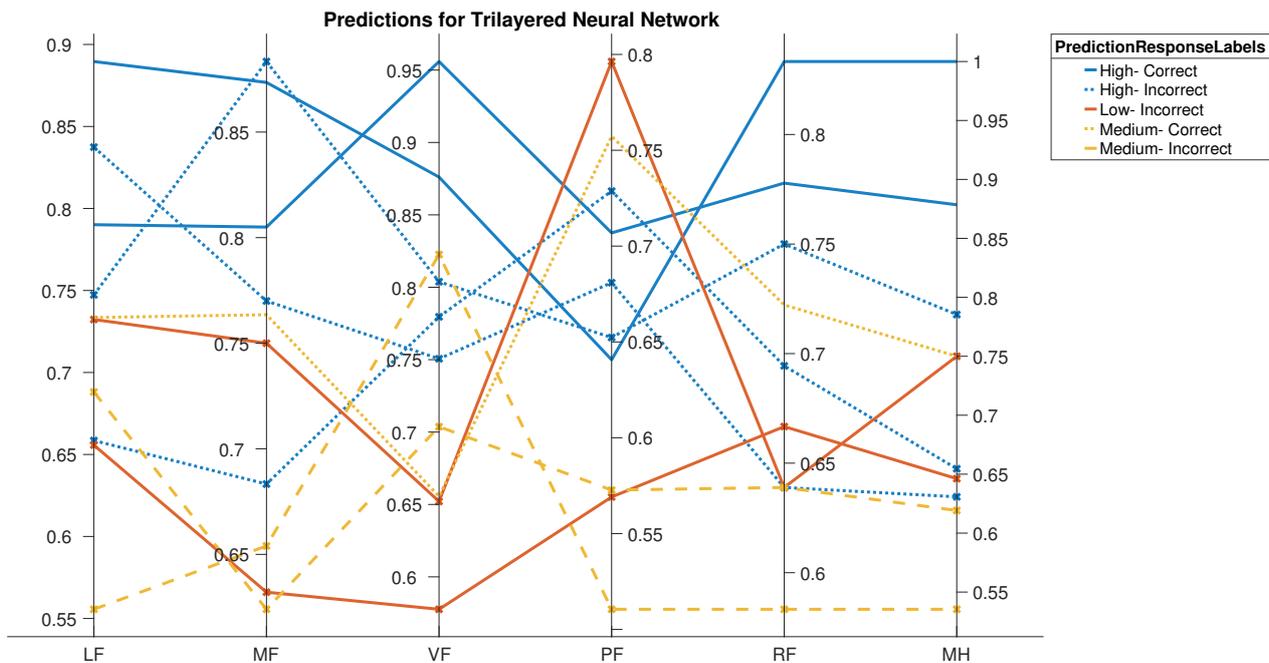


Figure 22. Validation Predictions: Trilayered Neural Network LM Soft Sensor.

4. Conclusions

Lean manufacturing attainment is quintessential for the auto component firms to survive and thrive in modern competitive markets. Effective management of manufacturing flexibilities leads to improvement in the lean manufacturing level of an enterprise. Due to the complex interplay of various flexibilities, it is a challenging task to predict their combined influence on lean manufacturing. The present study applied a machine learning approach to develop an effective lean manufacturing soft sensor, to predict/classify the lean attainment of an auto component firm based on the existing levels of its manufacturing flexibilities. As many as 29 models were explored, belonging to seven different machine learning architectures, viz. ANN, KNN, SVM, decision trees, discriminants, Naive Bayes, and ensembles. The trilayered neural network LM soft sensor attained the highest testing accuracy of 80%. The fine, medium, and coarse trees, the narrow and wide neural networks, the quadratic and cubic SVMs and the ensemble RUSBoosted tree—all these models scored 60% testing accuracies. The medium and bilayered neural networks, the Gaussian, and Kernel Naive Bayes, the medium, coarse, cosine, cubic and weighted KNNs, the linear, fine, medium, and coarse Gaussian SVMs, the linear discriminant, and the ensemble boosted, bagged and subspace discriminant models attained only 40% testing accuracies. The fine KNN and the ensemble subspace KNN could achieve only 20% testing accuracy, whereas the quadratic discriminant was unable to model the flexibilities-LM dataset used in this study.

The best performing model was further analysed by scatter plots of predicted LM classes versus flexibilities, validation and testing confusion matrices, receiver operating characteristics (ROC) curves, and the parallel coordinate plot for identifying manufacturing flexibility trends for the predicted LM levels. The scatter parallel coordinate plots showed

that the correctly predicted high LM class firms had higher flexibilities than their medium LM class counterpart, except in case of the product flexibility. The confusion matrices and the ROC curves revealed the exact class-wise predictive performance of the trilayered neural network soft sensor for validation and testing datasets. Moreover, the precision, recall and F1 scores obtained from the testing confusion matrix as well as the ROC curves confirmed that the trilayered neural network soft sensor was insensitive/robust to the relatively unbalanced dataset employed in the present study. Hence, this work establishes that machine learning based soft sensors can be developed to suitably classify the lean manufacturing level of an auto component manufacturing firm based on its manufacturing flexibilities.

The successful lean manufacturing level classification of auto component manufacturing firms based on their flexibilities using machine learning soft sensors holds great importance in today's scenario of competitive and dynamic markets, supply chains, and consumer preferences. Manufacturing firms can frequently measure their lean manufacturing levels by deploying machine learning-based soft sensors and take suitable actions for improvement based on their flexibility data/levels. In the present study, exploration of multiple architectures within the same model group was achieved by varying few hyperparameters. The remaining hyperparameters were kept/assumed constant. Moreover, the present work did not include an in-depth algorithm-level analysis of why some machine learning models performed better than others for the given dataset. In a future work, all hyperparameters will be varied/optimized, coupled with a methodical analysis of machine learning models' performances to overcome these limitations of the present study to further improve/maximize the LM classification accuracy of the machine learning soft sensors.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/asi6010022/s1>, File S1: Survey questionnaire.

Author Contributions: Conceptualization, N.S. and R.S.; methodology, R.S. and P.S.; software, P.S.; validation, R.S. and P.S.; formal analysis, R.S.; investigation, N.S., R.S. and P.S.; writing—original draft preparation, R.S.; writing—review and editing, R.S., P.S. and N.S.; visualization, R.S., P.S. and N.S. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by the Symbiosis International (Deemed University), Pune, Maharashtra state, India.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Browne, J.; Dubois, D.; Rathmill, K.; Sethi, S.P.; Stecke, K.E. Classification of flexible manufacturing systems. *FMS Mag.* **1984**, *2*, 114–117.
2. Gerwin, D. An agenda for research on the flexibility of manufacturing processes. *Int. J. Oper. Prod. Manag.* **1987**, *7*, 38–49. [[CrossRef](#)]
3. Sethi, A.K.; Sethi, S.P. Flexibility in manufacturing: A survey. *Int. J. Flex. Manuf. Syst.* **1990**, *2*, 289–328. [[CrossRef](#)]
4. Shewchuk, J.P.; Moodie, C.L. Definition and classification of manufacturing flexibility types and measures. *Int. J. Flex. Manuf. Syst.* **1998**, *10*, 325–349. [[CrossRef](#)]
5. Bhamu, J.; Sangwan, K.S. Lean manufacturing: Literature review and research issues. *Int. J. Oper. Prod. Manag.* **2014**, *34*, 876–940. [[CrossRef](#)]
6. Sajjan, M.; Shalij, P.; Ramesh, A. Lean manufacturing practices in Indian manufacturing SMEs and their effect on sustainability performance. *J. Manuf. Technol. Manag.* **2017**, *28*, 772–793.
7. Vasanthakumar, C.; Vinodh, S.; Ramesh, K. Application of interpretive structural modelling for analysis of factors influencing lean remanufacturing practices. *Int. J. Prod. Res.* **2016**, *54*, 7439–7452. [[CrossRef](#)]
8. Zhu, X.; Lin, Y. Does lean manufacturing improve firm value? *J. Manuf. Technol. Manag.* **2017**, *28*, 422–437. [[CrossRef](#)]

9. Asadi, N.; Fundin, A.; Jackson, M. The essential constituents of flexible assembly systems: A case study in the heavy vehicle manufacturing industry. *Glob. J. Flex. Syst. Manag.* **2015**, *16*, 235–250. [[CrossRef](#)]
10. Wei, Z.; Song, X.; Wang, D. Manufacturing flexibility, business model design, and firm performance. *Int. J. Prod. Econ.* **2017**, *193*, 87–97. [[CrossRef](#)]
11. Boyle, T.A. Towards best management practices for implementing manufacturing flexibility. *J. Manuf. Technol. Manag.* **2006**, *17*, 6–21. [[CrossRef](#)]
12. Kaur, S.P.; Kumar, J.; Kumar, R. Impact of Flexibility of Manufacturing System Components on Competitiveness of SMEs in Northern India. *J. Eng. Proj. Prod. Manag.* **2016**, *6*, 63–76. [[CrossRef](#)]
13. Koste, L.L.; Malhotra, M.K. A theoretical framework for analyzing the dimensions of manufacturing flexibility. *J. Oper. Manag.* **1999**, *18*, 75–93. [[CrossRef](#)]
14. Parker, R.P.; Wirth, A. Manufacturing flexibility: Measures and relationships. *Eur. J. Oper. Res.* **1999**, *118*, 429–449. [[CrossRef](#)]
15. Chauhan, G.; Singh, T.; Sharma, S. Cost reduction through lean manufacturing: A case study. *Int. J. Ind. Eng. Pract.* **2009**, *1*, 1–8.
16. Chauhan, G.; Singh, T.; Sharma, S. Role of machine flexibility in lean manufacturing. *Int. J. Appl. Eng. Res.* **2009**, *4*, 25–34.
17. Sushil. Multiple Perspectives of Flexible Systems Management. *Glob. J. Flex. Syst. Manag.* **2012**, *13*, 1–2. [[CrossRef](#)]
18. Shahu, R.; Pundir, A.K.; Ganapathy, L. An empirical study on flexibility: A critical success factor of construction projects. *Glob. J. Flex. Syst. Manag.* **2012**, *13*, 123–128. [[CrossRef](#)]
19. Gupta, Y.P.; Somers, T.M. Business strategy, manufacturing flexibility, and organizational performance relationships: A path analysis approach. *Prod. Oper. Manag.* **1996**, *5*, 204–233. [[CrossRef](#)]
20. Chauhan, G.; Singh, T.P.; Sharma, S.K. Flexibility implications in manufacturing system: A framework. *Int. J. Eng. Res. Ind. Appl.* **2008**, *1*, 83–98.
21. Chauhan, G.; Singh, T. Measuring parameters of lean manufacturing realization. *Meas. Bus. Excell.* **2012**, *16*, 57–71. [[CrossRef](#)]
22. Chauhan, G.; Singh, T. Development and validation of resource flexibility measures for manufacturing industry. *J. Ind. Eng. Manag.* **2013**, *7*, 21–41. [[CrossRef](#)]
23. Kaur, S.P.; Kumar, J.; Kumar, R. The relationship between flexibility of manufacturing system components, competitiveness of SMEs and business performance: A study of manufacturing SMEs in Northern India. *Glob. J. Flex. Syst. Manag.* **2017**, *18*, 123–137. [[CrossRef](#)]
24. Mendes, L.; Machado, J. Employees' skills, manufacturing flexibility and performance: A structural equation modelling applied to the automotive industry. *Int. J. Prod. Res.* **2015**, *53*, 4087–4101. [[CrossRef](#)]
25. Zhang, Q.; Vonderembse, M.A.; Lim, J.S. Manufacturing flexibility: Defining and analyzing relationships among competence, capability, and customer satisfaction. *J. Oper. Manag.* **2003**, *21*, 173–191. [[CrossRef](#)]
26. De Toni, A.; Tonchia, S. Manufacturing flexibility: A literature review. *Int. J. Prod. Res.* **1998**, *36*, 1587–1617. [[CrossRef](#)]
27. Ali, M.; Ahmad, Z. A simulation study of FMS under routing and part mix flexibility. *Glob. J. Flex. Syst. Manag.* **2014**, *15*, 277–294. [[CrossRef](#)]
28. Ali, M.; Murshid, M. Performance evaluation of flexible manufacturing system under different material handling strategies. *Glob. J. Flex. Syst. Manag.* **2016**, *17*, 287–305. [[CrossRef](#)]
29. Mishra, R.; Pundir, A.K.; Ganapathy, L. Manufacturing flexibility research: A review of literature and agenda for future research. *Glob. J. Flex. Syst. Manag.* **2014**, *15*, 101–112. [[CrossRef](#)]
30. Solke, N.S.; Singh, T. Analysis of relationship between manufacturing flexibility and lean manufacturing using structural equation modelling. *Glob. J. Flex. Syst. Manag.* **2018**, *19*, 139–157. [[CrossRef](#)]
31. Solke, N.S.; Shah, P.; Sekhar, R.; Singh, T. Machine Learning-Based Predictive Modeling and Control of Lean Manufacturing in Automotive Parts Manufacturing Industry. *Glob. J. Flex. Syst. Manag.* **2022**, *23*, 89–112. [[CrossRef](#)]
32. Shah, P.; Sekhar, R.; Kulkarni, A.J.; Siarry, P. *Metaheuristic Algorithms in Industry 4.0*; CRC Press: Boca Raton, FL, USA, 2021.
33. Kuo, Y.H.; Kusiak, A. From data to big data in production research: The past and future trends. *Int. J. Prod. Res.* **2019**, *57*, 4828–4853. [[CrossRef](#)]
34. Ejsmont, K.; Gladysz, B.; Corti, D.; Castano, F.; Mohammed, W.M.; Martinez Lastra, J.L. Towards Lean Industry 4.0 Current trends and future perspectives. *Cogent Bus. Manag.* **2020**, *7*, 1781995. [[CrossRef](#)]
35. Jamwal, A.; Agrawal, R.; Sharma, M.; Giallanza, A. Industry 4.0 Technologies for Manufacturing Sustainability: A Systematic Review and Future Research Directions. *Appl. Sci.* **2021**, *11*, 5725. [[CrossRef](#)]
36. Tortorella, G.L.; Narayanamurthy, G.; Thurer, M. Identifying pathways to a high-performing lean automation implementation: An empirical study in the manufacturing industry. *Int. J. Prod. Econ.* **2021**, *231*, 107918. [[CrossRef](#)]
37. Gupta, S.; Modgil, S.; Gunasekaran, A. Big data in lean six sigma: A review and further research directions. *Int. J. Prod. Res.* **2020**, *58*, 947–969. [[CrossRef](#)]
38. Belhadi, A.; Kamble, S.S.; Zkik, K.; Cherrafi, A.; Touriki, F.E. The integrated effect of Big Data Analytics, Lean Six Sigma and Green Manufacturing on the environmental performance of manufacturing companies: The case of North Africa. *J. Clean. Prod.* **2020**, *252*, 119903. [[CrossRef](#)]
39. Grover, P.; Kar, A.K. Big data analytics: A review on theoretical contributions and tools used in literature. *Glob. J. Flex. Syst. Manag.* **2017**, *18*, 203–229. [[CrossRef](#)]
40. Valamede, L.S.; Akkari, A.C.S.; Cristina, A. Lean 4.0: A new holistic approach for the integration of lean manufacturing tools and digital technologies. *Int. J. Math. Eng. Manag. Sci.* **2020**, *5*, 851–868. [[CrossRef](#)]

41. Bi, Z.; Jin, Y.; Maropoulos, P.; Zhang, W.J.; Wang, L. Internet of things (IoT) and big data analytics (BDA) for digital manufacturing (DM). *Int. J. Prod. Res.* **2021**, *1*, 1–18. [[CrossRef](#)]
42. Abd Rahman, M.S.B.; Mohamad, E.; Abdul Rahman, A.A.B. Development of IoT—enabled data analytics enhance decision support system for lean manufacturing process improvement. *Concurr. Eng.* **2021**, *29*, 1–13. [[CrossRef](#)]
43. Vlachos, I.P.; Pascazzi, R.M.; Zobolas, G.; Repoussis, P.; Giannakis, M. Lean manufacturing systems in the area of Industry 4.0: A lean automation plan of AGVs/IoT integration. *Prod. Plan. Control.* **2021**, *1*–14. [[CrossRef](#)]
44. Singh, R.; Bhanot, N. An integrated DEMATEL-MMDE-ISM based approach for analysing the barriers of IoT implementation in the manufacturing industry. *Int. J. Prod. Res.* **2020**, *58*, 2454–2476. [[CrossRef](#)]
45. Anosike, A.; Alafropatis, K.; Garza-Reyes, J.A.; Kumar, A.; Luthra, S.; Rocha-Lona, L. Lean manufacturing and internet of things—A synergetic or antagonist relationship? *Comput. Ind.* **2021**, *129*, 103464. [[CrossRef](#)]
46. Zhang, K.; Qu, T.; Zhou, D.; Thüerer, M.; Liu, Y.; Nie, D.; Li, C.; Huang, G.Q. IoT-enabled dynamic lean control mechanism for typical production systems. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1009–1023. [[CrossRef](#)]
47. Weichert, D.; Link, P.; Stoll, A.; Rüping, S.; Ihlenfeldt, S.; Wrobel, S. A review of machine learning for the optimization of production processes. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 1889–1902. [[CrossRef](#)]
48. Kutschenreiter-Praszkiwicz, I. Machine learning in SMED. *J. Mach. Eng.* **2018**, *18*, 31–40. [[CrossRef](#)]
49. Khayyati, S.; Tan, B. Data-driven control of a production system by using marking-dependent threshold policy. *Int. J. Prod. Econ.* **2020**, *226*, 107607. [[CrossRef](#)]
50. Wang, B.; Tao, F.; Fang, X.; Liu, C.; Liu, Y.; Freiheit, T. Smart manufacturing and intelligent manufacturing: A comparative review. *Engineering* **2020**, *7*, 738–757. [[CrossRef](#)]
51. Sekhar, R.; Singh, T.; Shah, P. Machine learning based predictive modeling and control of surface roughness generation while machining micro boron carbide and carbon nanotube particle reinforced Al-Mg matrix composites. *Part. Sci. Technol.* **2022**, *40*, 355–372. [[CrossRef](#)]
52. Sabahi, S.; Parast, M.M. The impact of entrepreneurship orientation on project performance: A machine learning approach. *Int. J. Prod. Econ.* **2020**, *226*, 107621. [[CrossRef](#)]
53. Dubey, R.; Gunasekaran, A.; Childe, S.J.; Bryde, D.J.; Giannakis, M.; Foropon, C.; Roubaud, D.; Hazen, B.T. Big data analytics and artificial intelligence pathway to operational performance under the effects of entrepreneurial orientation and environmental dynamism: A study of manufacturing organisations. *Int. J. Prod. Econ.* **2020**, *226*, 107599. [[CrossRef](#)]
54. Purohit, K.; Srivastava, S.; Nookala, V.; Joshi, V.; Shah, P.; Sekhar, R.; Panchal, S.; Fowler, M.; Fraser, R.; Tran, M.K.; et al. Soft sensors for state of charge, state of energy, and power loss in formula student electric vehicle. *Appl. Syst. Innov.* **2021**, *4*, 78. [[CrossRef](#)]
55. Sekhar, R.; Shah, P.; Panchal, S.; Fowler, M.; Fraser, R. Distance to empty soft sensor for ford escape electric vehicle. *Results Control Optim.* **2022**, *9*, 100168. [[CrossRef](#)]
56. Shah, P.; Kulkarni, R.; Sekhar, R. Soft Sensors For Urban Water Body Eutrophication Using Two Layer Feedforward Neural Networks. *IAENG Int. J. Comput. Sci.* **2022**, *49*, 778–807.
57. Sekhar, R.; Singh, T.; Shah, P. ARX/ARMAX modeling and fractional order control of surface roughness in turning nano-composites. In Proceedings of the 2019 International Conference on Mechatronics, Robotics and Systems Engineering (Morse), Bali, Indonesia, 4–6 December 2019; pp. 97–102.
58. Sekhar, R.; Singh, T.; Shah, P. Micro and nano particle composite machining: Fractional order control of surface roughness. In Proceedings of the Third International Conference on Powder, Granule and Bulk Solids: Innovations and Applications PGBSIA, Patiala, India, 26–28 February 2020; pp. 35–42.
59. Sekhar, R.; Singh, T.; Shah, P. System identification of tool chip interface friction while machining CNT-Mg-Al composites. *AIP Conf. Proc.* **2021**, *2317*, 020019.
60. Jatti, V.S.; Sekhar, R.; Shah, P. Machine Learning Based Predictive Modeling of Ball Nose End Milling using Exogeneous Autoregressive Moving Average Approach. In Proceedings of the 2021 IEEE 12th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), Cape Town, South Africa, 13–15 May 2021; pp. 68–72.
61. Sun, Y.; Wong, A.K.; Kamel, M.S. Classification of imbalanced data: A review. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 687–719. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.