Taylor and Francis Book Chapter

LOC Page

Contents

CHAPTER 1

Evaluating Targeted Productivity in Bangladesh's Garment Sector Using Machine Learning and Deep Learning with Explainable AI: A Data-Driven Method for Enhanced Production Planning

Jonayed Al-Faruk

Mymensingh Engineering College, Dept. of Computer Science and Engineering, Mymensingh-2200, Bangladesh.

Md. Rifat Hossen

Pabna University of Science and Technology, Dept. of Information and Communication Engineering, Pabna-6600, Bangladesh.

Joyonta Das

Mymensingh Engineering College, Dept. of Computer Science and Engineering, Mymensingh-2200, Bangladesh.

Ekram Hossain

Mymensingh Engineering College, Dept. of Computer Science and Engineering, Mymensingh-2200, Bangladesh.

Yeamin Safat

Mymensingh Engineering College, Dept. of Computer Science and Engineering, Mymensingh-2200, Bangladesh.

Ozifatun Jannat Akhi

Mymensingh Engineering College, Dept. of Computer Science and Engineering, Mymensingh-2200, Bangladesh.

CONTENTS

THE garment industry significantly boosts Bangladesh's economy, contributing over 80% to export earnings and employing millions of people. Effective production planning is vital for operational efficiency; however, establishing productivity targets often depends on manual judgment, resulting in unrealistic expectations. This study introduces a data-driven framework utilizing machine learning (ML) and deep learning (DL) methods to accurately predict productivity targets, supplemented by Explainable AI (XAI) tools such as SHAP and LIME. Various models—including Linear Regression, Random Forest, KNN, XGBoost, Neural Network (NN), Deep Neural Network (DNN), and Convolutional Neural Network (CNN)—were assessed through 10-fold cross-validation. Performance metrics (MAE, MSE, RMSE, R^2) identified the Neural Network as the top performer, achieving an R^2 of 0.987. The incorporation of XAI facilitated understanding of feature influences, emphasizing factors like SMV, WIP, idle time, and overtime. This methodology promotes transparent and intelligent planning, enabling factory managers to make better-informed, data-driven decisions.

1.1 INTRODUCTION

The garment sector in Bangladesh serves as the country's primary source of income, accounting for over 80% of export earnings and providing jobs for over 4 million people [1]. It is essential for economic growth and poverty alleviation. Additionally, it generates jobs for millions, particularly rural women, improving their socio-economic standing [2]. Recently, the industry has experienced a transformation focused on data to enhance productivity and planning. In industrial environments, human decisions frequently suffer from bias and are constrained by personal judgment [3]. Conversely, data-driven methods provide a more objective, quicker, and consistent way to make decisions [4, 5, 6]. A number of researchers have investigated the use of machine learning for predicting productivity in garment manufacturing. Begum et al. [7] created a hybrid framework that combines Decision Tree and CatBoost Regressors, utilizing sophisticated preprocessing techniques. Biswas et al. [8] not only predicted productivity but also estimated the necessary incentives and overtime through optimization techniques including Particle Filter and Simulated Annealing. Hosen et al. [9] implemented stacking ensemble techniques with Gradient Boosting, resulting in excellent performance. Imran et al. [10] implemented a Deep Neural Network model, which lowered prediction error relative to conventional methods. Keya et al. [11] examined female workers and discovered that Logistic Regression is effective in assessing the socio-economic and health-related factors affecting productivity. Goyzueta et al. [12] demonstrated that ensemble learning models, such as XGBoost, can surpass both traditional and deep learning methods. Although previous research has primarily concentrated on forecasting actual productivity, there has been little attention given to predicting targeted productivity, which is essential for effective production planning. Targeted productivity has traditionally been set manually, frequently ignoring real-time factory conditions. This oversight can result in unrealistic goals, inefficiencies, and resource wastage. In garment manufacturing, identifying suitable incentive levels, overtime, and workforce distribution to meet productivity goals continues to be a complicated issue. Without expert guidance, manually adjusting these parameters frequently leads to less than ideal results. By projecting productivity targets ahead of time using operational data like SMV, WIP, idle time, and worker allocation, production planning and resource management can be greatly improved. This study seeks to fill this gap by introducing a deep learning model that predicts targeted productivity based on these operational features. Utilizing historical production data, the model assists managers in establishing more realistic and optimized targets. This method minimizes the subjective elements of manual planning, allowing for better-informed decision-making. As a result, it enhances planning efficiency and narrows the disparity between expected outcomes and actual performance. Thus, the key findings from this research are:

- Estimating productivity targets proactively with operational metrics like SMV, WIP, idle time, and workforce allocation allows for smarter production planning and better resource management.
- This study compared a traditional machine learning model with advanced deep

learning models, highlighting deep learning's superior ability to identify complex patterns and enhance productivity prediction accuracy.

- XAI methods such as SHAP and LIME offer clear explanations and highlight feature importance, enhancing the interpretability of predictions actionable.

The rest of the paper is organized into three sections. Section 1.2 covers the methodology and provides a detailed explanation of the proposed model. The next Section 1.3 presents the results and discussions, which include a comparative analysis of different models. Lastly, Section 1.4 summarizes the paper's findings.

1.2 METHODOLOGY

The diagram in Fig. 1.1 illustrates a systematic framework for forecasting productivity in the garment industry. This process includes collecting data, preprocessing it (which covers encoding and scaling), conducting k-fold cross-validation, training models via machine learning and deep learning techniques, and evaluating results using MAE, MSE, RMSE, and R^2 metrics.We implemented a training and testing procedure. To train a machine learning model effectively, a large dataset is essential for optimal performance. We allocated 70% of 1176 instances for training (823 instances) and 30% for testing (353 cases).

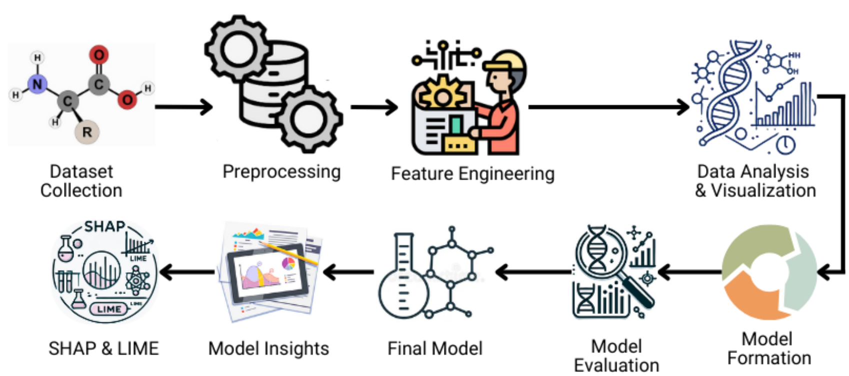

Figure 1.1 Model Architechture

1.2.1 Data Collection

For implementing and validating our study's model, we utilized the publicly available 'Productivity Prediction of Garment Employees' dataset from Kaggle [13]. This

CSV dataset features 1197 instances and 15 attributes, including date, department, quarter, and targeted productivity, providing a solid basis for predictive analysis in the garment sector. The attributes and their functions in the dataset are listed in Table 1.1.

Table 1.1 Description of Dataset Features

Feature	Description
Date	Specific date in 'month-date-year' format
Department	Department linked to the instance
Team no	Assigned team number
No of workers	Number of workers in a team
No of style changes	Number of modifications in product style
Targeted productivity	Productivity goal set by authorities
SMV	Standard Minute Value for a specific task
WIP	Number of incomplete items (Work in Progress)
Over time	Overtime minutes recorded for the team
Incentive	Monetary bonus for specific task completion
Idle time (In BDT)	Time lost due to production disruptions
Idle men	Number of workers idle during downtime
Actual productivity	Measured productivity (scale 0.0 to 1.0)

1.2.2 Data Pre-processing

Data preprocessing enhances both the accuracy and interpretability of models through techniques like handling missing values, encoding features, scaling data, splitting into training and testing sets, and applying k-fold validation. These processes promote data consistency and improve the reliability of predictions in our analysis.

1.2.2.1 Handling Missing Value:

Preparing data is essential for dependable machine learning performance. Missing values can skew results, so proper management is vital. In our research, we discovered 506 missing entries in the WIP and missing values within the datetime attribute. We resolved these issues through interpolation. Furthermore, columns lacking unique values—'department,' 'date,' and 'day'—were removed.

1.2.2.2 Quartile Detection and Removal:

Outliers, which significantly differ from surrounding data points, were identified through boxplot visualization. To enhance data quality and model performance, this study removed outliers using the Interquartile Range (IQR) method, eliminating points that fell outside the lower and upper bounds. We have identified outliers in the following columns, ‘targeted productivity,’ ‘over time,’ and ’incentive’ as depicted in Fig.1.2, and the boxplot after removing outliers is presented in Fig. 1.3.

Outliers were constrained to specific upper and lower limits. Because the incentive’s lower limit was negative, it was adjusted to zero. These limits were established using the following formulas:

$$IQR = Q3–Q1 \tag{1.1}$$

$$LowerLimit = Q1 - 1.5 * IQR \tag{1.2}$$

$$UpperLimit = Q3 + 1.5 * IQR \tag{1.3}$$

Q1 and Q3 refer to the first and third quartiles, respectively, with the interquartile range (IQR) determined by subtracting Q1 from Q3.

1.2.2.3 Feature Encoding:

This process converts categorical variables into numerical formats since machine learning models learn only from numeric data. We need to encode categorical data like ‘department’, ‘Team no’, ‘month’, ‘quarter’, and ‘year’ into numeric values. A label encoder was used for this conversion. The label encoder utilized is from scikit-learn [14], which functions by replacing strings with integers starting from zero up to (number of classes - 1).

1.2.2.4 Feature Scalling:

Feature scaling is vital to prevent numerical features from disproportionately influencing the model, ensuring equal contribution to performance. In our study, it was essential for gradient descent algorithms because they are sensitive to input feature scaling. Techniques include standardization, Min-Max normalization, robust scaling, and MaxAbs scaling. Standardization was chosen based on the dataset’s nature, as it works well for features that are approximately normally distributed. Also known as Z-score normalization, this method transforms the data to have a mean of 0 and a standard deviation of 1. The StandardScaler() function from ‘scikit-learn’ is utilized for this purpose, and the subsequent formula [15] computes the standard score of a sample:

$$z = \frac{X - \mu}{\sigma} \tag{1.4}$$

Where mu represents the mean and sigma denotes the standard deviation of the training data.

1.2.2.5 Visualization:

We created a correlation heatmap Fig. 1.6 to show relationships in our dataset. Analysis revealed that smv and ’no of workers’ correlate positively with ’wip’ and ’incentive’, while ’no of style change’ has a weak negative correlation with ’targeted productivity’ and ’actual productivity’. We displayed histograms 1.5 to examine individual feature distributions. These histograms show skewed distributions for incentive, ’idle time’, and ’no of style change’, while ’team’ and ’idle men’ show consistent distributions.

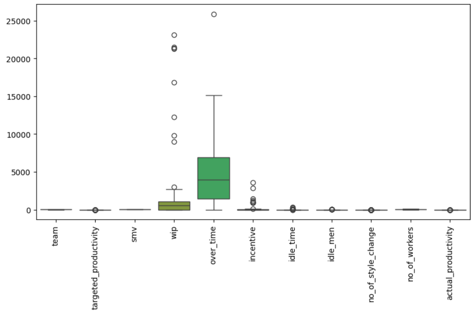

Figure 1.2 Boxplot befor removing outliers

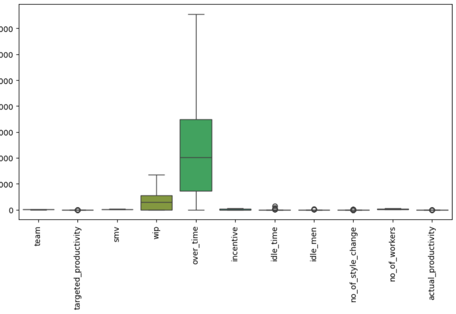

Figure 1.3 Boxplot after removing outliers

Furthermore, we utilized density plots Fig. 1.4 to examine the smooth probability distribution of the numeric variables. These visualizations aided in identifying the spread, modality, and skewness of the data. Variables like 'actual productivity' and 'targeted productivity' exhibited right-skewed distributions, suggesting a predominance of lower values.

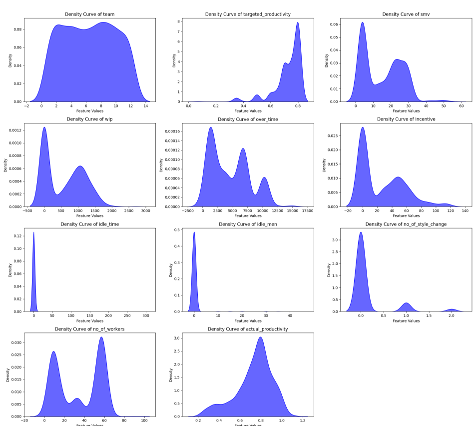

Figure 1.4 Density Plots of Dataset Features

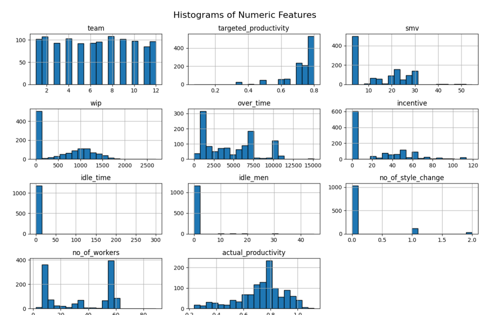

Figure 1.5 Density Plots of Dataset Features

1.2.3 Model Classification

In our research, we implemented four machine learning algorithms and three deep learning models for experimentation. A brief overview of all these algorithms is provided below-

1.2.3.1 Linear Regression:

Linear Regression is a straightforward and widely used supervised machine learning algorithm for regression issues. It aims to model the relationship between a single dependent variable (target) and one or more independent variables by fitting a straight

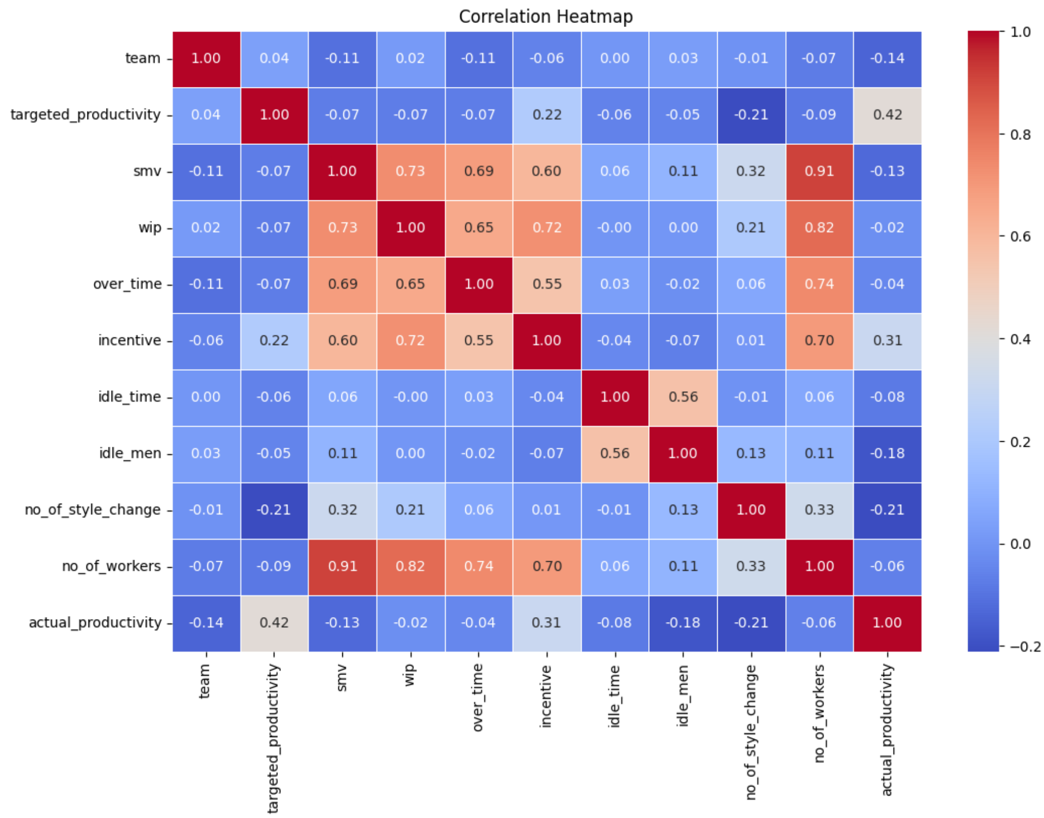

Figure 1.6 Heatmap depicting the strength of relationships between features.

line (simple linear regression) or a hyperplane to the data. The formula for simple linear regression is:

$$y = mx + by \tag{1.5}$$

1.2.3.2 Random Forest:

The random forest algorithm is a meta-learning machine learning technique [16]. To determine an overall classification for a specified set of inputs, the random forest employs multiple random tree classifications. The advantages stem from its substantial learning ability, robustness, and the practicality of the hypothesis space [17].

1.2.3.3 K- Nearest Neighbor:

KNN Regression is a supervised learning algorithm that is non-parametric and instance-based. It predicts a continuous output by averaging the target values of

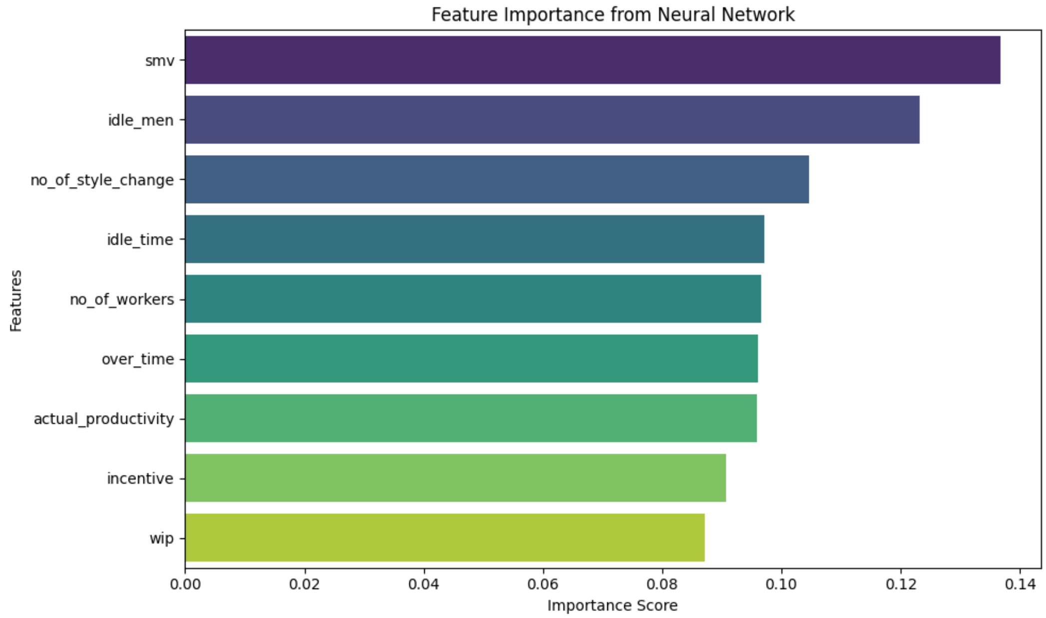

Figure 1.7 Contribution of individual features to the model's prediction in a step-by-step breakdown.

the K nearest neighbors from the training set data.

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2} \tag{1.6}$$

1.2.3.4 XGBoost:

Gradient boosting machines (GBMs) are regression strategies similar to boosting [18]. These machine-learning systems are effective in various applications [19]. In GBMs, new models are fitted sequentially to enhance response variable estimation accuracy. This technique constructs new base learners aligned with the negative gradient of the loss function for the entire ensemble [3]. Essentially, we combine weak learners to create a robust model for specific problems. This is calculated using the following formula [20].

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \tag{1.7}$$

1.2.3.5 Neural Network:

A feedforward neural network was developed with TensorFlow to tackle the regression task. It featured an input layer, two hidden layers employing ReLU activation, and one output neuron for predicting continuous values. The model was trained with

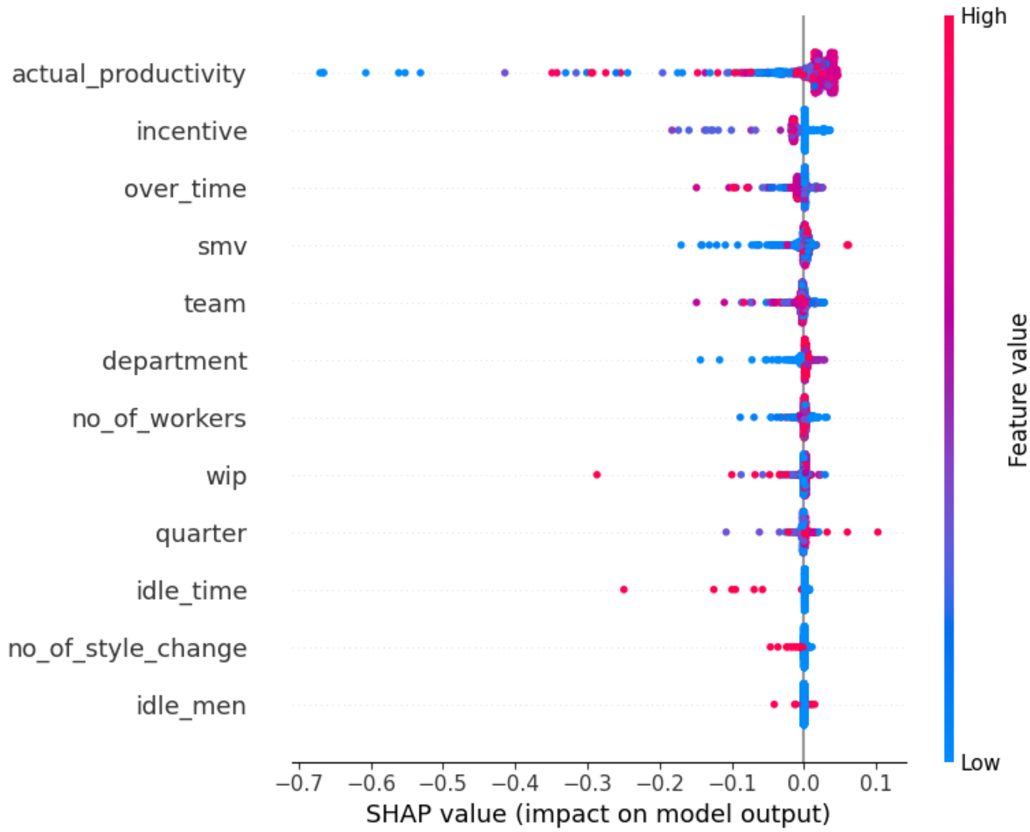

Figure 1.8 Feature importance visualization using explainable AI (XAI) techniques.

the Adam optimizer and used mean squared error (MSE) for the loss function. To evaluate its performance on unseen test data, metrics such as MAE, RMSE, and R^2 score were applied. The network successfully identified non-linear relationships in the dataset, showcasing reliable predictive accuracy.

1.2.3.6 Deep Neural Network:

A DNN model created with TensorFlow was employed to forecast continuous values based on various input features. It comprised three hidden layers (128, 64, and 32 neurons) utilizing ReLU activation and a 30% dropout rate. The model was trained with the Adam optimizer and MSE loss, and its effectiveness was assessed through MSE, RMSE, MAE, and R^2 metrics.

1.2.3.7 Convulational Neural Network:

A 1D CNN was utilized for regression on structured time-series data by converting inputs into a 3D format. The model comprised convolutional and max pooling layers, succeeded by dense layers, and was optimized using Adam and MSE loss. The evaluation metrics validated its proficiency in detecting local feature patterns.

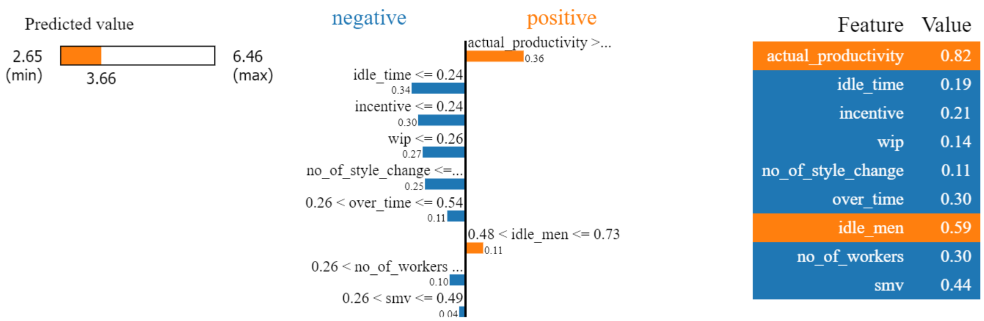

Figure 1.9 LIME analysis showing local feature contributions to model predictions.

1.2.4 Model Evaluation

We evaluated the predictive capability of our proposed models we utilizing four standard regression metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2 Score). These metrics provide important insights into the accuracy and reliability of our predictions. The evaluated models comprise traditional machine learning algorithms such as Linear Regression, K-Nearest Neighbors (KNN), alongside ensemble methods like Random Forest and XGBoost. Furthermore, we assessed deep learning models including Neural Networks (NN), Deep Neural Networks (DNN), and Convolutional Neural Networks (CNN). The performance of each model was evaluated with a held-out test dataset. According to Table 1.2, the Neural Network model recorded the lowest MAE and RMSE, along with the highest R^2 Score, demonstrating its superior generalization capabilities. Conversely, linear models and certain ensemble methods showed elevated error rates, indicating a restricted ability to grasp the dataset's underlying complexity.

Table 1.2 Overview of Performance Metrics

Name	Equation	Meaning
MSE	$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$	Reflects how well the model predicts outcomes
MAE	$\frac{1}{n}\sum_{i=1}^{n}\lvert Y_i - \hat{Y}_i\rvert$	Captures the average difference between predicted and actual values
RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$	Represents the square root of the average squared prediction errors
R^2	$1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$	Shows the proportion of variation in the target variable explained by the model

1.2.5 Final Model

The proposed neural network model exhibited better predictive performance than traditional machine learning algorithms, demonstrating its ability to capture complex non-linear relationships in garment productivity data.

1.2.6 Model Insights

In this study, We utilize Explainable AI (XAI) methods like SHAP and LIME to enhance the interpretability of our neural network model that predicts garment worker productivity. SHAP (SHapley Additive exPlanations) offers a reliable and theoretically sound approach to measure how much each input feature contributes to model predictions, providing an overarching view of feature significance [21]. Conversely, LIME (Local Interpretable Model-agnostic Explanations) aims to create interpretable models that are locally accurate in explaining individual predictions, thus making the behavior of complex models comprehensible to humans [22]. We integrate global and local interpretability tools to validate our model's decisions, boost transparency, and enhance trust and usability in practical applications within the garment industry.

1.3 RESULT AND DISCUSSION

This section provides a comparison of the performance among various machine learning models, from traditional regressors to deep learning architectures, along with a thorough analysis of feature importance and interpretability. We employ evaluation metrics and explainable AI methods. A thorough comparison involving seven models, such as traditional regressors, ensemble methods, and deep learning architectures, was conducted using evaluation metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the R^2 score. The results are presented in Table 1.3.

The Neural Network (NN) demonstrated the best overall performance, with an MAE of 0.080, MSE of 0.010, RMSE of 0.100, and an R^2 score of 0.987. This indicates that the model can explain nearly all the variability in the target variable. Traditional ensemble models, including Random Forest and Gradient Boosting (GBoost), followed with moderate R^2 scores of 0.33 and 0.19, respectively. Conversely, deeper models like DNN and CNN underperformed, likely due to data limitations and overfitting, leading to higher MAE and RMSE values. To understand the contribution of each input variable toward productivity prediction, feature importance was analyzed using the neural network model. As illustrated in Fig. 1.7, smv, idle men, and no of style change were identified as the top three most influential features. These variables are directly related to process efficiency and human resource utilization. This ranking indicates that enhancing the standard minute value (SMV), along with minimizing idle labor and frequent style changes, could greatly boost productivity. Meanwhile, features such as incentives and work-in-progress (WIP) received the lowest importance scores, reflecting a lesser impact. A SHAP-inspired interpretability method was utilized to improve the clarity of model decisions. Fig. 1.8 illustrates the feature contributions for a particular prediction. Positive factors (shown in orange), like actual productivity and idle men, increased the prediction, while negative factors (shown in blue), such as idle time, incentive, and wip, lowered the predicted value. Table 1.4 displays the fold-wise performance of the top-performing model, highlighting its reliability and advantages compared to traditional models in k-fold cross-validation. Additionally, the feature value table next to the force plot shows the precise input values utilized for prediction, allowing domain experts to validate the model's reasoning. This visualization provides important insights into how real-time operational parameters affect predicted productivity, supporting decision-making in manufacturing settings. The LIME analysis illustrates how operational features affect productivity predictions. Increased actual productivity (0.82) and a moderate incentive (0.21) enhance productivity, while a rise in idle men (0.59) and overtime (0.30) diminishes efficiency. The model shows sensitivity to thresholds such as SMV (0.44) and idle men (0.59), and feature interactions (e.g., low WIP combined with an optimal number of workers) collectively influence results. This highlights the NN's capability to pinpoint crucial productivity drivers for process optimization, as shown in Fig. 1.9.

Table 1.3 Model-wise Performance

Model	**MAE**	**MSE**	**RMSE**	**R^2 Score**
Linear Regression	0.052	0.027	0.166	0.037
KNN	0.045	0.026	0.160	0.069
Random Forest	0.039	0.019	0.130	0.330
XGBoost	0.0436	0.024	0.150	0.150
NN	**0.080**	**0.010**	**0.0100**	**0.987**
DNN	0.209	0.066	0.258	0.923
CNN	0.220	0.081	0.284	0.905

Table 1.4 Fold-wise Performance of Best Model

Fold	**MAE**	**MSE**	**RMSE**	**R^2 Score**
1	0.0870	0.0120	0.1130	0.9880
2	0.0816	0.0102	0.1008	0.9874
3	0.0887	0.0131	0.1143	0.9855
4	0.0856	0.0114	0.1067	0.9862
5	0.0809	0.0098	0.0992	0.9879
6	0.0710	0.0079	0.0891	0.9921
7	0.0879	0.0118	0.1086	0.9872
8	0.0810	0.0106	0.1030	0.9849
9	0.0772	0.0094	0.0970	0.9881
10	0.0784	0.0096	0.0981	0.9907

This study emphasizes the success of integrating ML, DL, and XAI methods to predict garment productivity accurately. The Neural Network (NN) excelled beyond all other models, identifying intricate patterns even in the absence of extra features. While the study showed promise, it did not investigate time-based models such as RNNs or Temporal CNNs, which might enhance accuracy. Table 1.5 demonstrates

Table 1.5 Comparison of Previous Studies on Productivity Prediction Models

Study	Best Model	MAE	R^2 Score	Prediction Target	Dataset (with Size)	XAI
[7]	CatBoost Regressor	–	Higher than Decision Tree	Actual team productivity	–	–
[8]	HGBR	0.0428	0.7142	Actual productivity + required incentive & overtime	1197	–
[9]	Stacking Ensemble	0.04	0.65	Actual productivity	1197	–
[10]	Deep Neural Network (DNN)	0.086	–	Actual productivity (0–1)	1197	–
[11]	Logistic Regression	–	–	Worker status (stay/leave)	512	–
[12]	Gradient Boosting	0.084	–	Actual productivity (0–1)	1197	–
Proposed	**Neural Network (NN)**	**0.080**	**0.987**	**Targeted Productivity**	**1197**	**YES (SHAP, LIME)**

that our method outperforms traditional models. The use of SHAP and LIME improved the model's transparency, ensuring that this framework is both interpretable and practical for real-time applications in garment manufacturing.

1.4 CONCLUSION

This study combines machine learning and deep learning models with SHAP and LIME explainable AI techniques, aiming to enhance both prediction accuracy and model interpretability. Various models were implemented, including Logistic Regression (LR), Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost, alongside deep learning frameworks like Neural Networks (NN), Deep Neural Networks (DNN), and Convolutional Neural Networks (CNN). All models underwent evaluation through 10-fold cross-validation to guarantee robust performance. The results indicate that the Neural Network (NN) model outperformed all others, positioning it as the leading approach in this research. The integration of XAI techniques with machine learning and deep learning facilitates more accurate and transparent predictions regarding targeted productivity in the garment sector. Future work may include integrating real-time production data, testing the model across multiple factories, and developing a decision-support system for factory managers to dynamically set productivity targets.

Bibliography

[1] Rahman, K. M. & Chowdhury, E. H. (2020). Growth trajectory and developmental impact of ready-made garments industry in Bangladesh. *Bangladesh's economic and social Progress: From a basket case to a development model*, 267–297. Springer.

[2] Jahan, M. (2012). Women workers in Bangladesh garments industry: a study of the work environment. *International Journal of Social Science Tomorrow*, 1(3),

1–5.

[3] Haque, S., Hoque, M. A., Khan, M. A. I., & Ahmed, S. (2021). Covid-19 detection using feature extraction and semi-supervised learning from chest x-ray images. *2021 IEEE Region 10 Symposium (TENSYMP)*, 1–5. IEEE.

[4] McKinsey Analytics. (2016). The age of analytics: competing in a data-driven world. *McKinsey Global Institute Research.*

[5] de la Fuente-Mella, H., Fuentes, J. L. R., & Leiva, V. (2020). Econometric modeling of productivity and technical efficiency in the Chilean manufacturing industry. *Computers & Industrial Engineering*, 139, 105793. Elsevier.

[6] Bousdekis, A., Lepenioti, K., Apostolou, D., & Mentzas, G. (2021). A review of data-driven decision-making methods for industry 4.0 maintenance applications. *Electronics*, 10(7), 828. MDPI.

[7] Begum, Z., Anjan, A., Abhivardhan, G., Tej, M. V., & Hashwanthratna, E. (2025). Machine learning models for analysing and predicting team productivity in garment manufacturing. *International Journal of Information and Electronics Engineering*, 15(4), 171–179.

[8] Biswas, M. E., Shahzamal, M., & Haque, M. D. (2024). Machine learning approach to estimate requirements for target productivity of garments employees. *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, 921–926. IEEE.

[9] Hosen, M. H., Tasnia, N., Amran, M., Chowdhury, R., Uddin, A., & Saha, A. (2024). Stacking ensemble techniques for productivity forecasting in Bangladesh's garment industry. *2024 IEEE International Conference on Computing, Applications and Systems (COMPAS)*, 1–6. IEEE.

[10] Al Imran, A., Amin, M. N., Rifat, M. R. I., & Mehreen, S. (2019). Deep neural network approach for predicting the productivity of garment employees. *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 1402–1407. IEEE.

[11] Keya, M. S., Emon, M. U., Akter, H., Imran, M. A. M., Hassan, M. K., & Mojumdar, M. U. (2021). Predicting performance analysis of garments women working status in Bangladesh using machine learning approaches. *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, 602–608. IEEE.

[12] Goyzueta, C. A. R., De la Cruz, J. E. C., & Machaca, W. A. M. (2021). Advantages of assembly machine learning models for predicting employee productivity in a garment manufacturing company. *2021 IEEE Engineering International Research Conference (EIRCON)*, 1–4. IEEE.

[13] Siri, S. (2024). Productivity prediction of garment employees. Retrieved from `https://www.kaggle.com/datasets/ishadss/productivity-prediction-ofgarment-employees` (Accessed: Jul. 08, 2024).

[14] Bisong, E. (2019). *Building machine learning and deep learning models on Google cloud platform.* Springer.

[15] scikit-learn. (2024). StandardScaler. Retrieved from `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html` (Accessed: Jul. 08, 2024).

[16] Livingston, F. (2005). Implementation of Breiman's random forest machine learning algorithm. *ECE591Q Machine Learning Journal Paper*, 1, 1–13.

[17] Ao, Y., Li, H., Zhu, L., Ali, S., & Yang, Z. (2019). The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. *Journal of Petroleum Science and Engineering*, 174, 776–789. Elsevier.

[18] Bikmukhametov, T., & Jäschke, J. (2019). Oil production monitoring using gradient boosting machine learning algorithm. *IFAC-PapersOnLine*, 52(1), 514–519. Elsevier.

[19] Torlay, L., Perrone-Bertolotti, M., Thomas, E., & Baciu, M. (2017). Machine learning–XGBoost analysis of language networks to classify patients with epilepsy. *Brain Informatics*, 4, 159–169. Springer.

[20] Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54, 1937–1967. Springer.

[21] Lundberg, S. M., & Lee, S. I. (2017). Consistent feature attribution for tree ensembles. *arXiv preprint arXiv:1706.06060.*

[22] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.