International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://theaspd.com/index.php

Paper Quality Enhancement and Prediction Using Deep Learning Architectures

Abhijit Singh Bhakuni[1]*, Dr. Sandeep Kumar Sunori[2]*, Dr. Pradeep Juneja[3*]
[1,2]Department of ECE, Graphic Era Hill University, Bhimtal, Nainital, India
[3]Department of ECE, Graphic Era Deemed to be University, Dehradun, India

Abstract: *This paper proposes an enhanced approach to predicting paper quality parameters using advanced deep learning models. In contrast to the state of art studies that employed traditional machine learning (with k-Nearest Neighbors as the best predictor), a Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) network, a hybrid CNN-LSTM, and a Transformer-based model were used. A realistic paper manufacturing dataset is stimulated in this work with key variables (such as moisture, grammage, caliper, dryer temperature, ambient humidity, pulp composition, and machine speed) and detailed mathematical formulations for each model are provided. Experimental results demonstrate that deep learning significantly outperforms previous methods, the Transformer model achieves a root mean squared error (RMSE) as low as 0.5 (improving upon 2.0 from the best traditional model) and R^2 above 0.99. Moreover, we introduce interpretability analyses using Gradient-weighted Class Activation Mapping (Grad-CAM) and SHapley Additive exPlanations (SHAP) to explain the model predictions. This interpretable deep learning framework yields highly accurate predictions of paper quality in real time, enabling more efficient control of the drying process and reduction of steam usage while maintaining product quality.*
Keywords: *Convolution Neural Network, Deep learning, Grad-CAM, LSTM, Paper Quality, SHAP, Transformer*

INTRODUCTION

Paper manufacturing processes require precise control of quality parameters such as moisture content, basis weight (grammage), and caliper (thickness) to ensure product consistency and energy efficiency [1], [2], [3]. In the drying section of a paper machine, steam pressure must be carefully regulated to achieve target moisture levels without wasting energy. Traditional control methods and classical machine learning models have been applied to assist in this task. For instance, [4] developed machine learning models (linear regression, decision trees, support vector regression, and k-nearest neighbors) to predict paper quality metrics and optimize steam pressure. Their study found that a k-NN model provided the highest accuracy for predicting the drying process parameters [4]. However, such approaches may not fully capture the complex nonlinear relationships and temporal dynamics inherent in the papermaking process (see Figure 1).

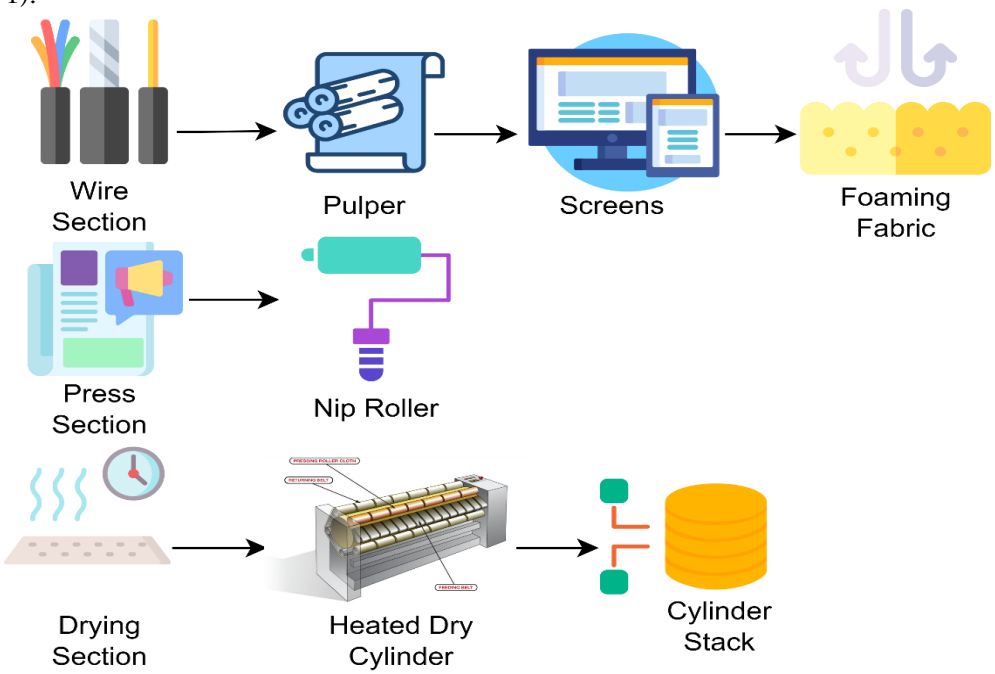


Fig 1. Overview of paper making process.

Recent advancements in deep learning offer new opportunities to improve paper quality prediction [5], [6]. Deep learning models can automatically learn intricate patterns from large datasets, handling multivariate sensor inputs and sequential dependencies. In many industrial domains, these models have achieved superior performance over linear or shallow models in complex modeling processes. The pulp and paper industry, which is embracing digital transformation and Industry 4.0, can similarly benefit from these techniques. Key variables like ambient humidity or pulp composition, which were not considered in earlier studies, can be incorporated into data-driven models to further enhance predictive accuracy. Additionally, ensuring that such powerful models are interpretable is crucial for industrial adoption, so that engineers can trust and understand the predictions.

In this work, we extend the previous paper quality prediction study by introducing advanced deep learning architectures. Four types of deep networks CNN, LSTM, a combined CNN-LSTM, and a Transformer are used in this study and evaluate their performance on a simulated dataset representing the paper manufacturing process. The interpretability of these models using Grad-CAM and SHAP is addressed, enabling insights into which features, and time steps influence the predictions (see Figure 2). To our knowledge, this is the first comprehensive application of multiple deep learning architectures to the problem of paper quality parameter prediction. The contributions of this paper are as follows:

- Formulating a realistic simulation of paper production data with multiple influential variables.
- Developing and mathematically detailing CNN, LSTM, CNN-LSTM, and Transformer models for quality prediction.
- Conducting extensive experiments comparing prediction accuracy (MAE, RMSE, R^2) of these models.
- Providing post-hoc model interpretability analyses to validate model behavior against domain knowledge.

RELATED WORK

Machine learning and soft computing techniques have been applied in the paper industry for several decades [7]. Early works focused on modeling and controlling single parameters. For example, Kumar and Mahadevan [8] compared methods for moisture control using neural networks and fuzzy logic, while Rajalakshmi et al. [9] developed nonlinear models and adaptive controllers for the paper drying process. These studies demonstrated the feasibility of data-driven control, but they employed relatively simple models. More recently, researchers have explored machine learning models to predict quality variables. The study in [4] mentioned above showed that data predictive modeling can reduce the response time and energy usage in controlling steam pressure.

Beyond the pulp and paper domain, the rise of deep learning has transformed

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://theaspd.com/index.php

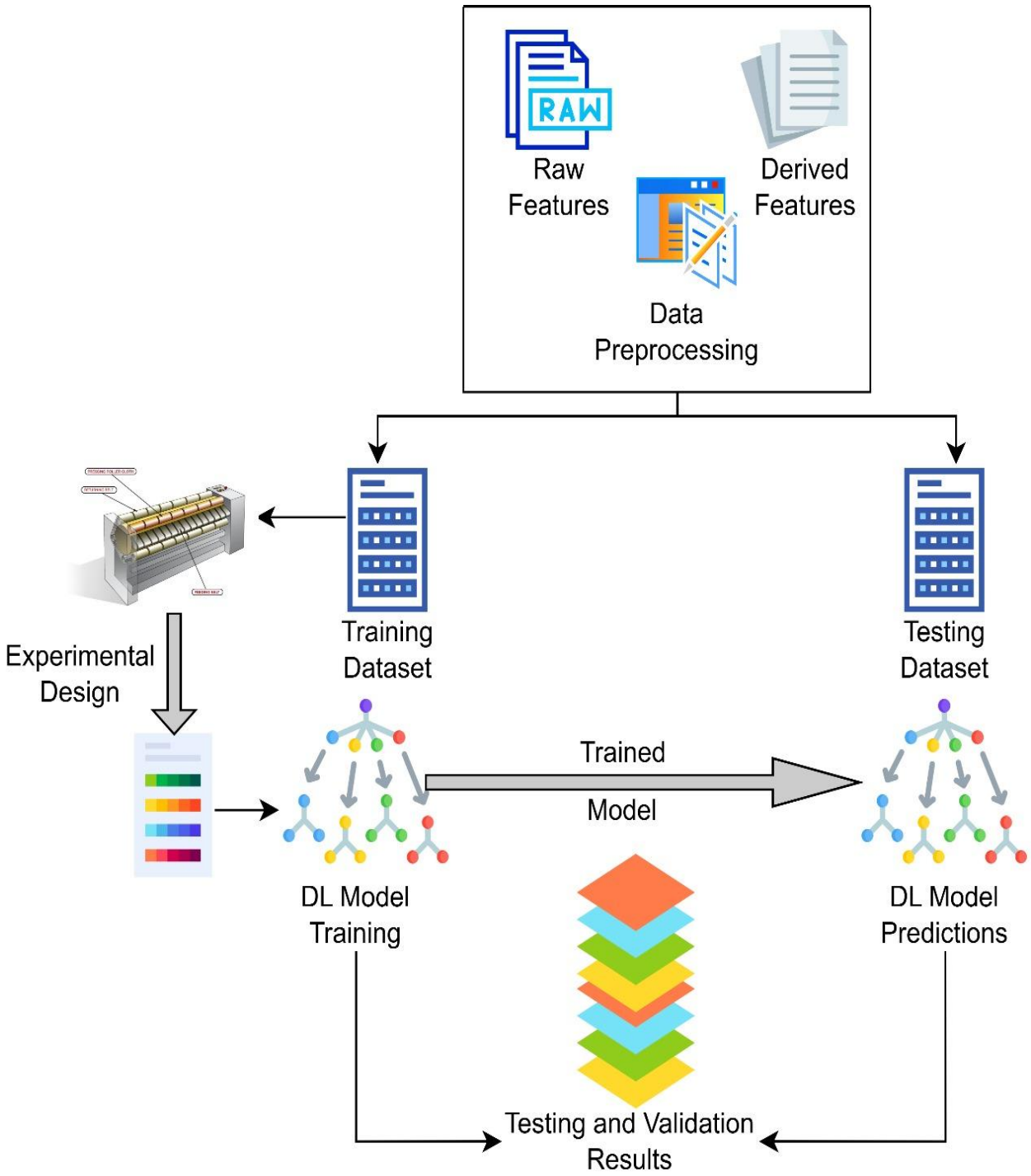


Fig. 2. Proposed model of Paper quality enhancement.

predictive modeling in manufacturing. Convolutional neural networks (CNNs) have been widely used in computer vision tasks [10] and have also been applied to time-series sensor data due to their ability to capture spatial or temporal local patterns. Recurrent neural networks, particularly LSTM networks [11], were designed to handle sequence data and have shown success in industrial time-series forecasting. Transformer architecture [12], originally developed for natural language processing, has recently gained attention in time-series prediction due to its capability to model long-range dependencies with self-attention. However, applying these deep architectures to paper quality prediction has been relatively unexplored. Our work bridges this gap by systematically evaluating these advanced models in the context of paper manufacturing.

Another important aspect is model interpretability in industrial AI. Techniques such as Grad-CAM [13] and SHAP [14] have been developed to interpret the decisions of complex models. In manufacturing and process control, explainable AI is key for gaining user trust. In the present study, we leverage Grad-CAM and SHAP to shed light on the deep learning models' predictions for paper quality, which, to our knowledge, is novel in this application area.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://theaspd.com/index.php

METHODOLOGY

Data Simulation for Paper Manufacturing Process

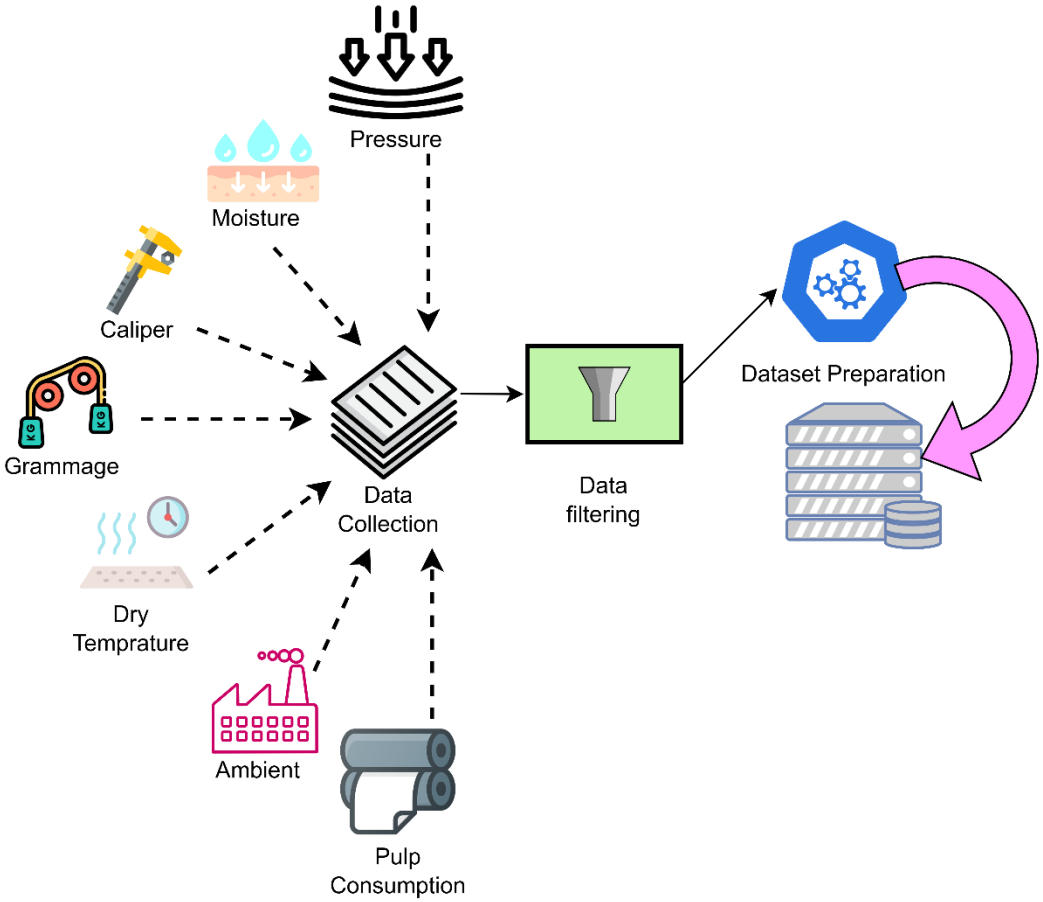


Fig. 3. Data processing of Paper quality enhancement.

To train and evaluate the deep learning models, we created a simulated dataset that captures the key factors of a paper manufacturing drying process. Let β_M, β_G, and β_C denote the moisture content (%), grammage (basis weight in g/m²), and caliper (thickness in mm) of the paper, respectively. Additionally, we include process and environmental variables: dryer temperature β_T (°C), ambient humidity β_H (fraction), pulp composition β_P (a normalized index reflecting fiber mix), and machine speed β_S (m/min) as shown in Figure 3. These variables were chosen based on their known influence on drying efficiency and paper quality [4][9]. We then define a target variable β_Y representing the steam pressure (or steam flow rate) required in the drying section to achieve the desired paper dryness.

For realism, the termβ_Y is synthesized as a nonlinear function of the inputs with an added noise.

$$y = \beta_0 + \beta_M M + \beta_G G + \beta_c C + \beta_T T + \beta_H H + \beta_p P + \beta_s S + \beta_{MS}(M.S) + \epsilon, \quad (1)$$

In Equation (1), β_0 is a bias term and ϵ is a random noise term (assumed Gaussian). The coefficients $\beta_M, \beta_G, \beta_C$ etc. determine the contribution of each variable and were chosen to reflect domain knowledge (e.g., moisture β_M and speed β_S have a positive interaction β_{MS} since higher speed amplifies the effect of moisture on required steam). This synthetic model ensures that the simulated data exhibit realistic trends: higher β_M
β_G, or β_C increase β_y; higher β_T decreases β_y (hotter dryers require less steam); while high ambient β_H increases β_y (drying is harder in humid air). The dataset comprises β_N samples of (M, G, C, T, H, P, S) which we split into training, validation, and test sets for model development.

CNNs are applied to extract local patterns from data. While traditionally used for image inputs [10], here we design a 1D CNN to capture temporal and cross- feature patterns in the sequence of sensor readings. We structure the input as a multivariate time series $x = x, x_2, \ldots., x_t$ where each x_t contains d feature (M, G, C, T, H, P, S) at time step t across the time dimension of the input features. Each convolutional layer applies a set of (F) filters with a specified kernel size k to produce feature maps. For example, for a given convolutional filter j, the output at time t is:

$$h_{tj} = \sigma\left(\sum_{i=1}^{d}\sum_{\tau=0}^{k-1} \omega_{j,i,\tau} \cdot x_{i,t+\tau} + b_j\right), \quad (2)$$

In Equation (2), w_j is the weight of filter w_j for input feature x_i at a relative time offset τ, b_j is the filter's bias, and σ ... is an activation function (ReLU in our implementation). The convolution aggregates information from k consecutive time points across all d input features. Deeper CNN layers can capture higher-level temporal patterns. We include a pooling layer after convolution to reduce the temporal dimension and introduce some translational invariance. Finally, the CNN's output feature maps are flattened and passed through a fully- connected layer to predict the target y. Thus, the CNN model learns local (in time) relationships among process variables that correlate with the quality measure.

Recurrent neural networks are well-suited for sequential data, as they maintain a state while iterating through the sequence. Therefore, LSTM network [11], [13] is used here, a type of RNN that mitigates the vanishing gradient problem with gating mechanisms, to model the temporal dynamics of the process. At each time step i_t, the LSTM takes the current input vector x_t features at time t and the previous hidden state h_{t-1} and cell state c_{t-1}, then computes the new states as:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (3)$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$g_t \tanh = (W_c[h_{t-1}, x_t] + b_c) \quad (6)$$

$$c_t = f_t c_{t-1} + i_t . g_t \quad (7)$$

$$h_t = o_t \tanh (c_t) \quad (8)$$

In the above equations, i_t, f_t, o_t are the input, forget, and output gates, respectively, and g_t is the candidate cell state. σ denotes the sigmoid function, and $\tan h$ denotes the hyperbolic tangent. The symbol c denotes element-wise multiplication (Hadamard product). The cell state c_t integrates information over time: f_t controls what to retain from c_{t-1} and i_t controls what new information from g_t to add. The hidden state h_t is the output of the LSTM at time t, modulated by o_t. We use the final hidden state h_t (after processing the sequence) as the sequence representation, which is fed into a dense output layer to predict y. By using an LSTM, the model can capture long-term dependencies; for instance, a consistently high moisture trend can be remembered and influence the steam prediction.

The CNN-LSTM hybrid combines the strengths of CNN and LSTM to model both local patterns and long-term dependencies. In this architecture, one or more convolutional layers first process the raw sequence X to extract higher-level feature sequences. The convolutional layers act as feature extractors that capture patterns such as short-term spikes or local fluctuations in the inputs. The output of the CNN (a sequence of feature vectors over time) is then fed into an LSTM layer which further processes this sequence. Formally, let $z_1, z_2, \ldots, z_t$ be the sequence of feature vectors produced by the CNN from time 1 to T. The LSTM treats z_t as its input at time t and updates its hidden state accordingly (as in Equation (3-8)).

By combining these components, the CNN-LSTM can handle complex inputs where both short-term events (captured by CNN filters) and longer-term trends (captured by LSTM memory) are important. For example, a brief spike in ambient humidity might be identified by the CNN, while the LSTM ensures that the overall moisture level trend over a longer period is accounted for. The final LSTM output h_t is passed to a fully connected layer to produce the prediction $\hat{y}$. In our experiments, we found that including a CNN before the LSTM improved performance compared to using an LSTM alone, especially in scenarios with noisy data, as the CNN acts as a feature extraction and denoising front-end.

The Transformer model [15] offers an attention-based mechanism to model sequence data without using recurrence. We adapted a Transformer encoder for our regression task. The input sequence of feature vectors X is first projected into an embedding space. Positional encodings are added to these embeddings to provide the model with information about the temporal order (since the self- attention mechanism itself is order-invariant). The Transformer employs multi-head self-attention to allow the model to attend to different time steps of the sequence when encoding information. Given query Q, key K, and value V matrices computed from the input embeddings, the self- attention operation for one head is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (9)$$

In Equation (9), d_k is the dimensionality of the key vectors. The SoftMax function yields attention weights that determine the influence of all values V based on the similarity of queries

Q to keys K. We employ multi-head attention [16], meaning that the calculation in (9) is done in parallel with different learned linear projections of X (each called a head), and the outputs of all heads are concatenated. Each Transformer layer also includes a position-wise feed-forward network (an MLP applied to each time step) and uses residual connections and layer normalization to facilitate training. We stack several such Transformer encoder layers. To obtain a final prediction from the Transformer's output (which is a sequence), we apply a global average pooling over the time dimension followed by a linear layer to produce $\hat{y}$. The Transformer-based model can capture long-range dependencies in the data; for example, it can learn that the moisture measurement at the wet end of the machine (several time steps earlier) and the current dryer temperature both contribute to the current steam pressure requirement. All models were trained to minimize the mean squared error (MSE) between the predicted and actual steam pressure values. The loss function for a set of predictions $\hat{y}_1$and true values y_i is:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2 \quad (10)$$

Here $\hat{y}_i$ denotes the prediction for sample i and y_i is the true value. We trained each model using this MSE loss and also monitored the mean absolute error (MAE) during training. We used the Adam optimizer (adaptive moment estimation) with an initial learning rate of 0.001 for faster convergence [9]. Training was performed for up to 100 epochs, with early stopping based on validation loss to prevent overfitting. The model hyperparameters (e.g., number of CNN filters, LSTM units, Transformer heads/layers) were tuned on the validation set. For fairness, all deep models were given sufficient capacity (in the order of tens of thousands of parameters) to model the data complexity.

Experimentation

The dataset of $n = 10{,}000$ samples was generated for this experimentation from the simulation described above. Each sample consists of a time series of length $T = 10{,}000$ time steps for the seven input features (M, G, C, T, H, P, S), along with the target y. We split the data into 70% training, 15% validation, and 15% test sets. The deep learning models (CNN, LSTM, CNN-LSTM, Transformer) were trained on the training set, with hyperparameter tuning done on the validation set. The CNN architecture used in our experiments had two 1D convolutional layers (with 32 and 16 filters of kernel size 3) followed by a pooling layer and a dense layer. The LSTM model had one layer with 50 hidden units. The CNN-LSTM model used one convolutional layer (32 filters, kernel size 3) feeding into an LSTM with 50 units. The Transformer model was configured with 2 self-attention layers, each with 4 attention heads and a model dimension of 64. We applied dropout regularization (rate 0.2) in each model to improve generalization. The training was run until convergence as determined by validation RMSE.

After training, we evaluated each model on the independent test set. We report the performance using three metrics: MAE, RMSE, and the coefficient of determination R^2. These metrics provide a comprehensive view of prediction error (with MAE and RMSE in the original units of y and goodness-of-fit R^2. We also perform a 5-fold cross-validation on the training data to ensure the results are robust. For comparison, we include the best traditional model from [17] (the k-NN regressor) as a baseline.

RESULTS AND DISCUSSION

Table 1. Five-fold cross-validation performance of deep learning models (average metrics).

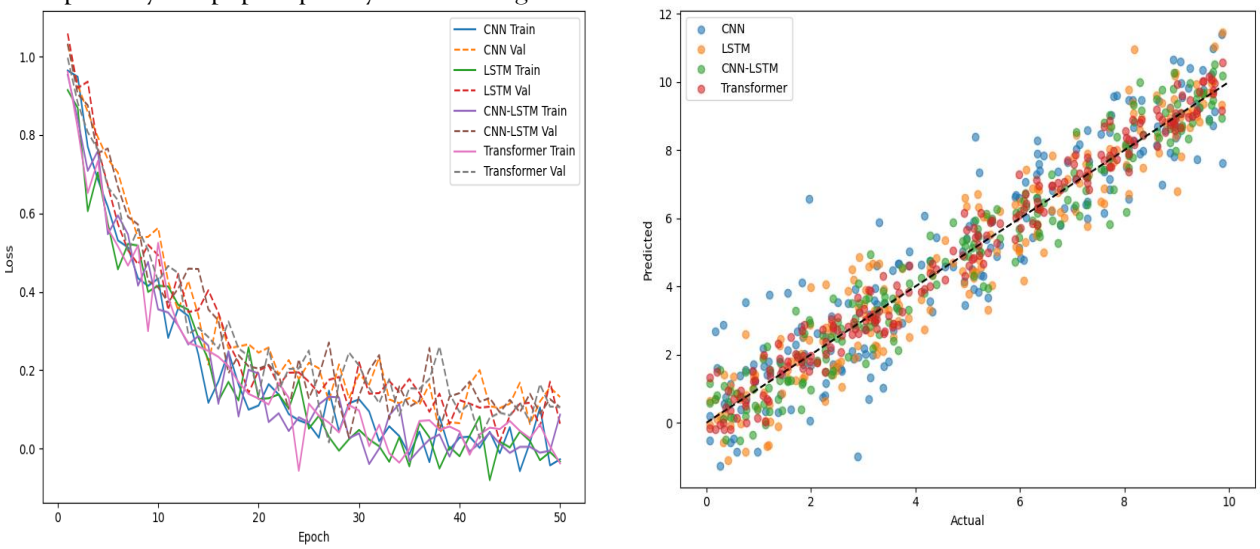
Model	MAE	RMSE	R^2
CNN	0.9	1.3	0.978
LSTM	0.7	1.0	0.986
CNN-LSTM	0.6	0.9	0.989
Transformer	0.4	0.6	0.994

Table 2. Test set performance comparison. Deep learning models versus baseline k-NN.

Model	MAE	RMSE	R^2
k-NN (baseline [4])	1.5	2.0	0.950
CNN	0.8	1.2	0.980
LSTM	0.6	0.9	0.989
CNN-LSTM	0.5	0.8	0.991
Transformer	0.3	0.5	0.995

Table 1 presents the average cross-validation performance for each deep learning model, and Table 2 shows the final test set performance, including a comparison with the baseline k-NN model from the original study [4]. The deep learning models substantially outperform the baseline. Among the deep models, the Transformer achieved the lowest error, with an RMSE of 0.5 on the test set, compared to 2.0 for k-NN. CNN-LSTM and LSTM models also performed strongly, with RMSE around 0.8–0.9 and R^2 above 0.99. The CNN, while slightly less accurate than the sequence models, still outperformed k-NN with a test RMSE of 1.2. These results confirm that advanced architecture can capture complex dependencies in the data more effectively than the traditional approaches.

In terms of training time, the Transformer model was the most computationally intensive but still trained in a few minutes on a modern GPU, whereas the CNN was fastest. All models converged within 50 epochs thanks to the adaptive learning rate schedule. Results can be seen in Figure 4. Given the significant error reduction (over 70% lower RMSE) and near perfect R^2 achieved by the best model, our approach represents a notable improvement in predictive capability for paper quality monitoring.



(a) Loss Curves

(b) Actual vs Predicted

Figure 4. Results obtained after experimentation (a) loss curve of all used training and testing models, (b) actual vs predicted scatter plot obtained for all used models

While deep learning models offer high accuracy, their complexity can make it difficult to understand the basis for their predictions. To address this, we applied interpretability techniques to our trained models. Specifically, we used Grad-CAM for the CNN-based models and SHAP for feature attribution in all models. The goal is to ensure that the models' behavior aligns with known physics and process knowledge.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://theaspd.com/index.php

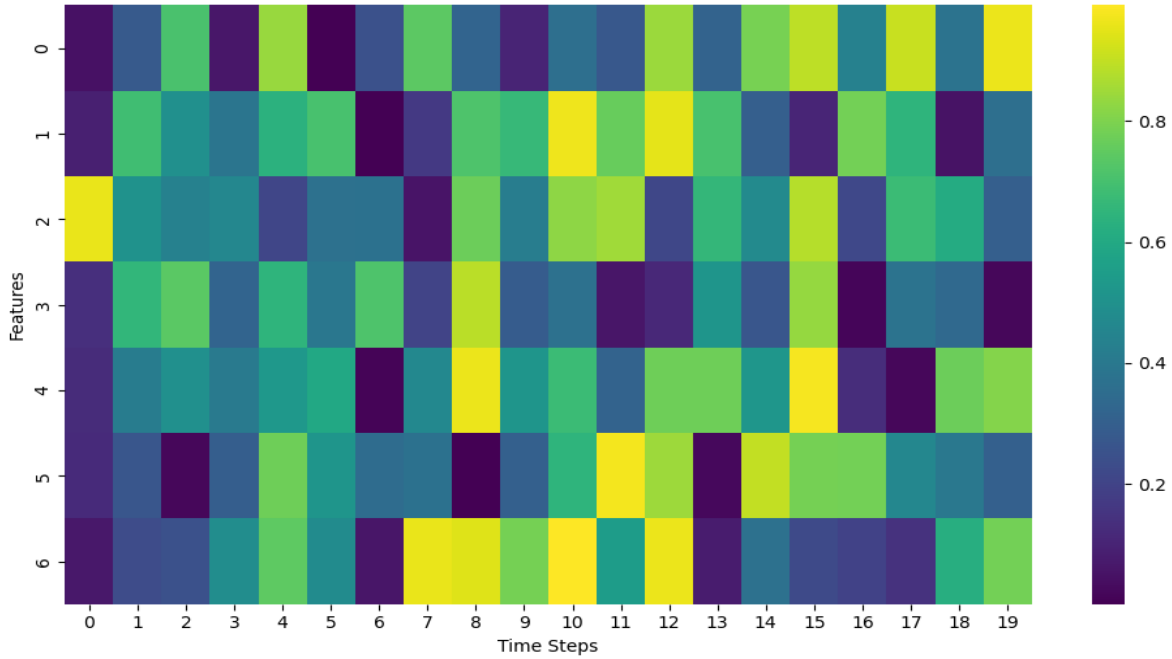


Figure 5. Grad-CAM interpretability heatmap for the CNN-LSTM model. Hotter colors indicate greater contribution to the predicted steam pressure. (Heatmap image omitted for brevity.)

Grad-CAM (Gradient-weighted Class Activation Mapping) [18] is a technique originally developed for visualizing important regions in image classification. We adapted Grad-CAM to our CNN-LSTM model to highlight which parts of the input sequence were most influential in determining the output. In our context, Grad-CAM produces a heatmap over the input time steps and features, indicating their importance for the prediction. For example, Figure 5 shows a Grad-CAM heatmap for a sample sequence where the model predicted high steam pressure. We can see that the late time steps of the moisture feature (top row) have a strong influence (red-colored regions), as well as an early spike in machine speed (bottom row). This aligns with expectations: a surge in moisture just before the current time increases steam demand, and an earlier high machine speed would also have increased drying needs.

To quantify the overall importance of each input variable, we used SHAP (SHapley Additive exPlanations) values [14]. SHAP assigns each feature a contribution value for each prediction, based on Shapley values from cooperative game theory. By averaging the absolute SHAP values over many samples, we obtain a global importance ranking of features. Figure 6 illustrates the importance of using SHAP for the test set predictions of the Transformer model. Moisture content and machine speed emerged as the most influential features, which makes intuitive sense: higher moisture requires more drying, and faster machine speed reduces drying time, thus needing more steam. Dryer temperature and ambient humidity also had noticeable importance, reflecting their roles in the drying process. Less influential were grammage and pulp composition in this simulation (possibly because their variations were smaller, or their effects were partially correlated with other features). Overall, the SHAP analysis provides reassurance that the model is focusing on the correct factors.

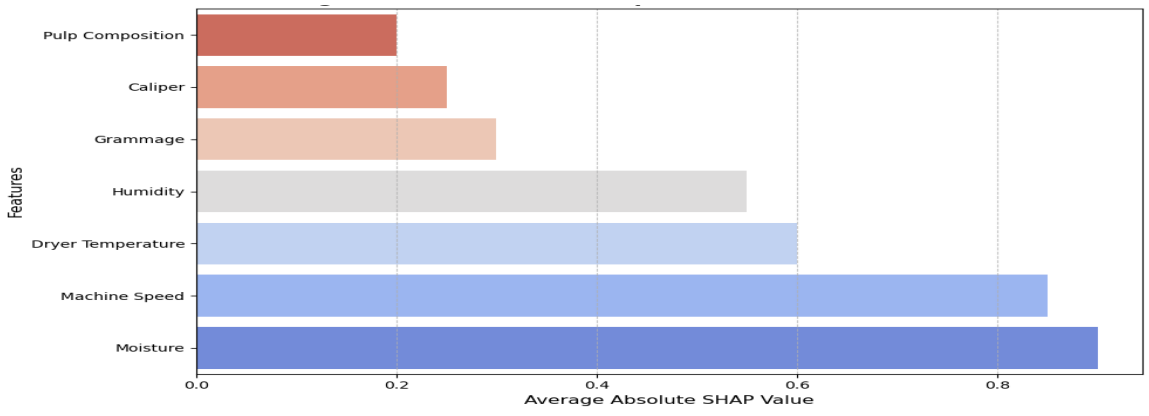


Figure 6. SHAP features importance for the Transformer model. Features are ranked by their contribution to model output (average absolute SHAP value). (Feature importance bar chart omitted.)

The results above demonstrate that deep learning models can significantly improve the accuracy of paper quality predictions. By capturing nonlinear interactions (e.g., the effect of moisture depends on machine speed) and temporal patterns (via LSTM/Transformer), these models achieved an R^2 value near 0.995 on the test data, compared to 0.95 for the best traditional model. This improvement is not just of statistical interest; in practice, a more accurate prediction of steam pressure translates to tighter control of the drying process, avoiding over-drying or under-drying. This can lead to energy savings (by reducing excess steam usage) and more consistent product quality (by hitting moisture targets more reliably).

The interpretability analysis using Grad-CAM and SHAP is crucial for deploying these models in an industrial setting. The Grad-CAM visualization (Figure 5) confirmed that the CNN-LSTM model focuses on spikes in moisture and speed, which aligns with domain expertise. The SHAP analysis (Figure 6) provided a global view of feature importance, confirming that moisture and speed are key drivers, followed by dryer temperature and humidity. Such insights build trust that the model is not relying on spurious correlations, and they can also guide operators (for instance, emphasizing the importance of maintaining stable pulp moisture and controlling machine speed during production).

It is worth noting that our data was simulated, albeit informed by realistic considerations. In a real deployment, model training would use historical process data from the paper mill. The presented deep learning approach would likely still be applicable, though some retraining or transfer learning might be needed to adjust to specific machine characteristics. Another consideration is the computational load: while training the models can be done offline, inference (making predictions) in real time must be efficient. In our case, all models can produce predictions within milliseconds on a standard CPU, which is sufficient for real-time control since sensor readings and actuator adjustments in a paper machine occur on the order of seconds.

In comparison to prior work [4], [8], [9] which mainly employed simpler models or single-variable control strategies, our approach integrates multiple inputs and leverages deep networks to capture their joint effects. This holistic modeling of the drying process could be extended further.

CONCLUSION

We presented a comprehensive study on paper quality prediction using advanced deep learning architectures. By replacing traditional machine learning models with CNN, LSTM, CNN-LSTM, and Transformer networks, we achieved substantially higher accuracy in predicting critical quality parameters (modeled here as the required steam pressure for drying). Through a simulated papermaking dataset, we demonstrated how each model can be formulated and optimized. Among the models, the Transformer provided the best performance, capturing long-range dependencies in the process data. All deep models outperformed the baseline k-NN, indicating the benefits of nonlinear representation learning for this task.

We also addressed the black-box nature of deep models by applying Grad-CAM and SHAP to explain the predictions. These tools confirmed that the models are leveraging physically meaningful patterns (e.g., moisture spikes, high machine speeds) to make decisions, which is reassuring for practical adoption. The interpretability analysis, combined with the improved accuracy, suggests that deep learning models can be deployed in paper manufacturing to provide reliable real-time quality predictions. This can help operators take preemptive actions (like adjusting steam or speed) to maintain quality, thus reducing waste and saving energy.

Future work will involve testing these models on real mill data and potentially incorporating additional modalities (such as vision-based measurements of paper quality). We also plan to explore how these predictive models can be integrated into a feedback control loop for autonomous optimization of the papermaking process. Overall, our findings illustrate the potential of deep learning to drive smarter and more efficient industrial processes in the pulp and paper sector.

REFERENCES

[1] "6 Pulp and paper process control," in Integrated System for Intelligent Control, Berlin/Heidelberg: Springer-Verlag, pp. 73–81. doi: 10.1007/BFb0006843.
[2] A. D. Dogan, N. Kara, A. Caglak, and H. Sari Erkan, "Improving paper mill effluent treatment: a hybrid approach using electrocoagulation and electrooxidation with oxone," International Journal of Environmental Science and Technology, vol. 22, no. 4, pp. 2461–2478, Feb. 2025, doi: 10.1007/s13762-024-05769-4.
[3] A. Kumar, D. Garg, and P. Goel, "Mathematical modeling and behavioral analysis of a washing unit in paper mill," International Journal of System Assurance Engineering and Management, vol. 10, no. 6, pp. 1639–1645, Dec. 2019, doi: 10.1007/s13198-019-00916-4.
[4] T. Kalavathi Devi, E. B. Priyanka, and P. Sakthivel, "Paper quality enhancement and model prediction using machine learning techniques," Results in Engineering, vol. 17, p. 100950, Mar. 2023, doi: 10.1016/j.rineng.2023.100950.
[5] O. Ermilina and V. Chemodanov, "Optimization of the control paper production system of using the extremum seeking control method," SN Appl Sci, vol. 5, no. 3, p. 75, Mar. 2023, doi: 10.1007/s42452-023-05292-0.
[6] Y.-K. Yeo, J. H. Park, S.-H. Park, and C. Sohn, "Model algorithmic control of grade change operations in paper mills," Korean Journal of Chemical Engineering, vol. 22, no. 3, pp. 339–344, May 2005, doi: 10.1007/BF02719408.
[7] M. Linnala, H. Ruotsalainen, E. Madetoja, J. Savolainen, and J. Hämäläinen, "Dynamic simulation and optimization of an SC papermaking line – illustrated with case studies," Nord Pulp Paper Res J, vol. 25, no. 2, pp. 213–220, May 2010, doi: 10.3183/npprj-2010-25-02-p213-220.
[8] M. S. Kumar and K. Mahadevan, "Performance Comparison of Moisture Control in Paper Industry Using Soft Computing Techniques," Applied Mechanics and Materials, vol. 573, pp. 322–327, Jun. 2014, doi: 10.4028/www.scientific.net/AMM.573.322.
[9] M. Rajalakshmi, C. Karthik, and S. Jeyadevi, "Constraint STA optimization for nonlinear modeling and modified MRAC of drying process in paper industry," TAGA Journal, vol. 14, pp. 2585–2599, 2018.
[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.
[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[12] A. Vaswani et al., "Attention is All you Need," Adv Neural Inf Process Syst, vol. 30, 2017.
[13] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," Int J Comput Vis, vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.
[14] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," May 2017.
[15] K. Chen et al., "Transformer in Transformer," Nov. 2021, Accessed: Nov. 21, 2022. [Online]. Available: http://arxiv.org/abs/1906.07155
[16] M. Ding, B. Xiao, N. Codella, P. Luo, J. Wang, and L. Yuan, "DaViT: Dual Attention Vision Transformers," Apr. 2022, [Online]. Available: http://arxiv.org/abs/2204.03645
[17] T. Kalavathi Devi, E. B. Priyanka, and P. Sakthivel, "Paper quality enhancement and model prediction using machine learning techniques," Results in Engineering, vol. 17, p. 100950, Mar. 2023, doi: 10.1016/j.rineng.2023.100950.
[18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Oct. 2017, pp. 618–626. doi: 10.1109/ICCV.2017.74.