https://doi.org/10.24867/GRID-2024-p91 Original scientific paper

BOOKBINDING OPERATIONAL PLANNING AND SCHEDULING OPTIMIZATION WITH DEEP LEARNING ALGORITHMS

Diana Bratić, Suzana Pasanec Preprotić, Denis Jurečić, Marko Šapina
University of Zagreb, Faculty of Graphic Arts, Zagreb, Croatia

Abstract: *This work focuses on the scheduling optimization of the bookbinding process through the application of industrial engineering principles, complemented by artificial intelligence as a key factor in achieving operational efficiency. Flexibility and the need to adapt quickly to fluctuating market conditions in the publishing industry present challenges that require precise operational planning. This includes detailed planning of resources such as time, labour and machinery as well as the optimization of technical and technological processes. Each subprocess is meticulously documented in technological specifications, which precisely describe manufacturing process, product specifications and required quality controls.*
By analysing input and output parameters in four case studies, a mathematical model was developed to optimize the production process for soft book binding. The model is based on deep learning algorithms that analyse extensive data about the production process and enable an accurate prediction of the optimal production parameters based on current demand and production capacities. For this purpose, an artificial neural network was constructed, which is considered suitable due to its ability to learn complex patterns from data and generate predictions or recommendations based on a large amount of input data. The artificial neural network consists of input, hidden and output layers, each using an activation function that determines the activation of the neurons based on the input data.
When training the artificial neural network, the weighting factors between the neurons in the hidden layers are adjusted to minimize the errors between the actual and predicted results. This architecture enables the artificial neural network to learn complicated data patterns and use the derived model for predictions and optimization of the production process in bookbinding. In addition, the model's predictions facilitate continuous monitoring and analysis of product quality, identifying potential defects or problems before they impact production. By integrating this model, bookbinderies are able to continuously improve their production process and thereby achieve greater efficiency and competitiveness in the market.

Key words: bookbinding, process, optimization, deep learning, artificial neural networks

1. INTRODUCTION

In the context of modern publishing and book production, the optimization of softcover binding processes is proving to be a decisive factor in increasing efficiency and reducing costs. The increasing demand for flexibility in production, adaptability to smaller batch sizes and dynamic market conditions require advanced approaches to the management and organization of production lines (Gašparić, Petković & Pasanec Preprotić, 2018). Traditional methods in bookbinding, based on manual planning and resource allocation, face significant limitations in terms of speed of adaptation, ability to monitor fluctuating market conditions and optimization of resource utilization (Cetinjanin, 2020). These challenges, compounded by the need to reduce costs and improve quality, have led to the development of new technological solutions based on digitalization and artificial intelligence (AI), including the use of artificial neural networks (ANN) as tools to optimize key production processes (Bratić, Pasanec Preprotić & Jurečić, 2023).
Softcover binding is a specialized segment of bookbinding that involves multiple stages, including sheet cutting, folding, gathering, binding and trimming, with each stage varying depending on the type of production line and equipment available. There are significant differences in efficiency between multi-stage lines with more complex semi-finished product flows and more flexible, modern lines characterized by fast cycles and optimized line layouts. In this study, four different technological-organizational scenarios (SS_A, SS_B, SS_C, SS_D) are analysed, ranging from traditional to more advanced production models in bookbinding. Each scenario offers different settings in terms of operational organization, machine layout and employee utilization. The aim of the analysis is to evaluate the efficiency of each scenario based on key figures such as production cycle times, resource consumption and potential to reduce downtime and losses.
The application of ANN models in this context offers the potential for significant improvements through accurate resource management, improved prediction of potential bottlenecks and more effective adjustments to changes in capacity and demand (Huang et al., 2020). ANN models are particularly beneficial

when analysing non-linear processes involving complex interactions between different variables, such as the number of sheets per book, cutting time, folding time, gathering time, binding time and trimming time. This adaptive optimization model analyses input data such as the number of sheets, the operating time for each task and the required number of employees and suggests optimal settings for production lines based on certain conditions.

By evaluating the results of the ANN model, it is also possible not only to predict optimal parameter values, but also to identify bottlenecks in the production process and strategies to minimize them (Qamar & Zardari, 2023; Aggarwal, 2018). For example, the ANN model can recommend an alternative machine layout or the use of additional labour to shorten the cycle time in multi-stage plants that suffer major losses when transporting semi-finished products (SS_A). In more flexible single-stage plants (SS_B), on the other hand, the focus may shift to optimizing machine utilization and shortening cycle times through improved operational planning.

This research not only examines the effectiveness of ANN models in predicting and optimizing key parameters of the production process, but also addresses how the implementation of such advanced technologies can contribute to the long-term sustainability and competitiveness of publishing companies (Pasanec Preprotić et al., 2022, 2023). By comparing different equipment layouts and production lines and analysing key operations such as cutting, folding, gathering, binding and trimming, this study lays a solid foundation for the implementation of advanced optimization methods in bookbinding industry.

2. FRAMEWORK FOR MODELLING

The production of books, whether soft or hardcover, requires sophisticated planning and optimization due to the complexity of the finishing processes such as folding, gathering, binding and trimming. These processes are of central importance for the overall efficiency of book production. However, traditional optimization approaches based on linear programming and manual methods often lack flexibility and effectiveness in a dynamic industrial environment that requires rapid adjustments to fluctuating market conditions (Pasanec Preprotić, Stančin & Petković, 2022).

The application of artificial intelligence, in particular artificial neural networks (ANN), opens up new possibilities for optimizing these processes (Brajković, Perinić & Ikonić, 2018). Although AI has not yet gained widespread acceptance in the publishing industry, initial studies indicate that the implementation of ANN and deep learning techniques can bring significant improvements in optimizing production phases, reducing operating costs and increasing production flexibility and efficiency.

The publishing industry involves multi-stage operations involving different machines. Traditional optimization models, such as heuristic approaches and linear models, often cannot fully adapt to the variability of operating conditions, including variations in labour and machine capacity. This inflexibility can lead to suboptimal resource utilization and increased costs (Dasović, Pasanec Preprotić & Petković, 2015).

Advanced AI systems, especially ANN models, have the potential to significantly improve decision-making processes in publishing. These models facilitate the development of predictive models that accurately predict the optimal production parameters based on historical data, enabling the optimization of labour usage, machine capacity and cycle times.

In addition to optimization in publishing, AI is also successfully used in related finishing processes in the graphic arts industry, such as the optimization of flexographic printing and colour management. In these processes, technologies such as ANN are used to manage resources and optimize ink application, taking into account variables related to print quality, production speed and material costs.

The application of artificial intelligence in the optimization of book production and related finishing processes holds significant potential to increase efficiency, reduce costs and increase production flexibility. AI systems can analyse complex interactions between different production variables and suggest optimal configurations that adapt to changing production conditions. This adaptability enables faster, higher quality and more efficient production at lower costs, improving competitiveness and flexibility in today's demanding market landscape (Voulgaris & Bulut, 2018).

The production of softcover books, which was analysed on the basis of four case studies (SS_A, SS_B, SS_C and SS_D), includes processes such as cutting, folding, gathering, binding and trimming. These processes require the interaction of different types of machines: high-speed cutters, folding machines, gathering machines, rotary binders and three-knife trimmers. The efficiency and duration of each process depends on the spatial arrangement of the equipment, the number of employees and the specifics of the working environment.

Case study A illustrates a two-floor production plant that significantly limits the potential for continuous operation due to the need to transport semi-finished products between floors. This system is less efficient than a single-storey plant where all processes are on the same level, eliminating the time losses associated with transportation. In contrast to case studies B, C and D, which were conducted in single-level production environments, systems with flexible machine layouts allow for better resource utilization and higher productivity.

The efficiency of the production processes varies depending on the arrangement of the machines and their ability to process in parallel. For example, the use of two folding machines in case studies B and C enables a significant reduction in the time required to fold printed sheets, while case study A shows considerably longer times due to the suboptimal layout of the machines and the inability to use a three-knife trimmer. Of particular note is case study C, where the inline binding system offers even greater efficiency compared to the roto-binder used in the other case studies.

When analysing the time required to perform each technological operation, it became clear that the greatest losses occur in plants with poorly organized production flow charts. In this context, the parallel operation of machines with multiple functions, such as folding machines, contributes significantly to reducing the overall production time.

3. IMPLEMENTATION OF ARTIFICIAL INTELLIGENCE IN THE OPTIMIZATION PROCESS

To further increase the efficiency of book production, a mathematical model was developed to optimize the production process by applying deep learning algorithms. This model is based on the analysis of input and output parameters derived from four case studies and uses an artificial neural network (ANN) that enables the prediction of optimal production parameters based on current demand and available capacity.

The artificial neural network (ANN) is divided into three main layers: the input layer, the hidden layers and the output layer. The input layer of the model receives data on the most important parameters of the production process, including the number of printed sheets, available machines, labour resources and the duration of individual technological operations. The hidden layers apply complex activation functions that allow the network to recognize patterns in the data, while the output layer generates forecasts that are used to optimize key production parameters (Haykin, 2008).

During the training phase of the artificial neural network, the weighting factors between the neurons in the hidden layers are continuously adjusted to minimize the error between the actual and predicted results. This process allows the network to learn complicated patterns in the data and use the resulting model to make predictions and optimize the production process. Optimization includes predicting the most efficient machine scheduling, optimal workforce utilization and reducing production downtime, which ultimately leads to greater efficiency (Russel & Norvig, 2020; Ertel, 2017).

One of the key benefits of this model is its ability to continuously monitor production quality. The model analyses output parameters such as production duration, number of finished units and material consumption and suggests real-time adjustments based on this data to avoid potential errors or problems in the production process.

3.1 Definition of the input and output parameters

Input data for the model

To successfully optimize the softcover binding process, the ANN model uses the following key input parameters that define the capacity and operating cycle of each machine:

- N_S - number of sheets per book
- T_{cu} - time for cutting the sheets per unit (in seconds)
- t_s - time for folding the sheets per unit (in seconds)
- t_{co} - time for collating (in seconds)
- t_b - time for binding (in seconds)
- t_t - time for trimming (in seconds)
- R - labour force (number of employees)
- P - number of units to be produced (in pieces)

These parameters are crucial for optimization, and the ANN uses their combination to make predictions about optimal production configurations.

Output data from the model
The aim of the model is to predict the following output parameters:

- T_{opt} - optimal cycle time for each machine
- S_{opt} - optimal machine configuration
- W_{opt} - optimal work configuration for each segment of production
- Q_{opt} - maximum number of units produced within a given time frame

Based on the model, the system can generate optimal recommendations for resources and time intervals that minimize production time while maintaining product quality. The optimized production cycle function aims to minimize the total production cycle time T_{total}, which is defined as the sum of the time required for all production steps and can be expressed by Equation (1):

$$T_{total} = P \cdot (t_{co} + t_s + t_{cu} + t_b + t_t) \quad (1)$$

Where is:

- T_{total} - total time required for production
- P - number of units produced

Under real conditions, however, the model also takes into account the dynamics of personnel scheduling and machine utilization. By using an ANN, this model can be optimized to predict configurations that minimize the total time T_{total} while maximizing the number of units produced.

3.2 Architecture of the Artificial Neural Network (ANN)

The artificial neural network consists of three primary layers (Qamar & Zardari, 2023):
Input layer: contains neurons that receive input parameters such as the number of sheets, production times and available labour.

- Hidden layers: one or more hidden layers utilize activation functions to recognize patterns within the data. These layers allow the network to learn complex relationships between input parameters.
- Output layer: predicts the optimal production settings and provides output data, including optimal cycle times, machine configurations and the expected number of units produced.

3.3 Activation Functions

Activation functions such as ReLU (Rectified Linear Unit) are used so that the neural network can process non-linear relationships in the Equation (2):

$$f(x) = max\,(0, x) \quad (2)$$

The ReLU function is well suited due to its ability to quickly learn patterns from large amounts of numerical data. For the output layer, a sigmoid or softmax function is used to normalize the results.

3.4 Learning Algorithm

The ANN uses the backpropagation algorithm to learn from data. During network training, the weighting factors W_1 and W_2 are adjusted to minimize the error, i.e. to reduce the difference between the actual and predicted results (mean squared error) using the gradient descent algorithm. The error function E is used to measure this difference and can be expressed by Equation (3):

$$E = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (3)$$

Where is:

- y_i - the actual output value
- $\hat{y}_i$ - the predicted output value
- n - the number of data samples

Since this refers to the mean squared error, the error or loss function to be optimized can also be written by Equation (4):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (4)$$

The mathematical model of the neural network is based on the approximation of a function that maps the input values *X* to the target values *y* and can be expressed by Equation (5):

$$y = f(W_2 \cdot f(W_1 \cdot X + b_1) + b_2 \quad (5)$$

Where is:

- X - input data (N_A, t_s, t_c, t_b, t_t, R, P)
- W_1, W_2 - weighting factors between the layers
- b_1, b_2 - biases
- f - activation function (ReLU)
- y - model output (predicted value for optimization)

Based on the ANN predictions, the optimization of bookbinding production can be expressed by an optimization function that minimizes the total production time T_{total}, which can be expressed mathematically by Equation (6):

$$MinT_{total} = P \cdot (\sum_{i=1}^{n} t_i \cdot x_i) \quad (6)$$

Where is:

- t_i - time for each machine i
- x_i - number of units processed by machine i
- P - number of units produced

Using this mathematical model and ANN optimization, a simulation of four scenarios is performed:

- Scenario A (SS_A): A multi-stage plant with material transportation problems between machines, resulting in longer cycle times.
- Scenario B (SS_B): A single-level plant with a flexible arrangement of machines and labour in which cycle speed is improved.
- Scenario C (SS_C): A seamless binding line where machine integration is most efficient and allows for shorter production cycles.
- Scenario D (SS_D): A standard line with a typical layout, but with optimized planning and manpower distribution.

4. GENERATING THE OPTIMIZATION CODE

To optimize the bookbinding production process using artificial neural networks (ANN), the Python programming language was used along with the following libraries: NumPy (for mathematical operations), TensorFlow (for building and training the model), Scikit-learn (for the machine learning functions and model evaluation), and Matplotlib (for the visualization of the data).
This code provides a basic implementation for optimizing the bookbinding production process using ANN. It can serve as a basis for more complex optimization models specifically tailored to the bookbinding industry. For further refinement, you can use more advanced architectures (such as RNNs or LSTMs for time-dependent data) and methods such as genetic algorithms or Bayesian optimization for parameter tuning.

4.1 Generating Simulated Data

The original data for four case studies is provided, including input parameters such as the number of sheets, cutting, folding, gathering, binding and trimming times, as well as the number of employees and order sizes. To enhance the data set for training the model, synthetic data is generated using a script that generates random values for these input parameters related to the bookbinding process (Figure 1).

```python
import numpy as np

# Original data
X = np.array([
    [54000, 15000, 8, 6, 1],  # Case Study A
    [54000, 15000, 8, 7, 1],  # Case Study B
    [54000, 15000, 9, 6, 1],  # Case Study C
    [54000, 15000, 9, 7, 1]   # Case Study D
])

y = np.array([37, 24, 18, 35])  # Total time to complete operations

# Generate synthetic data
def generate_synthetic_data(X, y, num_samples):
    np.random.seed(42)
    synthetic_X = []
    synthetic_y = []
    for _ in range(num_samples):
        sample = np.random.uniform(low=np.min(X, axis=0), high=np.max(X, axis=0))
        synthetic_X.append(sample)
        noise = np.random.normal(0, 1.5)  # Adding minor noise
        synthetic_y.append(np.dot(sample, np.array([0.0001, 0.0001, 0.5, 0.5, 0.5])) + noise)
    return np.array(synthetic_X), np.array(synthetic_y)

# Generate additional data
X_synthetic, y_synthetic = generate_synthetic_data(X, y, num_samples=100)

# Combine real and synthetic data
X_combined = np.vstack([X, X_synthetic])
y_combined = np.hstack([y, y_synthetic])
```

Figure 1: Code for generating simulated data

The original data is defined with the parameters of the production process and the actual execution times, while synthetic data is generated to expand the data set and improve the accuracy of the model during training.

4.2 Data Preparation and Splitting

The data is scaled and split into a training data set and a test data set so that the model can learn from part of the data and be tested on unseen data (Figure 2).

```python
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Scale the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_combined)

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_combined, test_size=0.25, random_state=42)
```

Figure 2: Code for data preparation and splitting

The data is scaled to ensure that all input parameters are on the same scale and then split into training (75%) and test (25%) data to evaluate the performance of the model.

4.3 Building the Artificial Neural Network (ANN)

An artificial neural network (ANN) is created and trained to predict time based on input parameters. TensorFlow is used to define, compile and create a multi-layer neural network model. The model is then trained on the simulated data to improve its prediction accuracy (Figure 3).

```python
import tensorflow as tf

# Build the ANN model
inputs = tf.keras.Input(shape=(X_train.shape[1],))
x = tf.keras.layers.Dense(256, activation='relu')(inputs)
x = tf.keras.layers.Dense(128, activation='relu')(x)
x = tf.keras.layers.Dense(64, activation='relu')(x)
outputs = tf.keras.layers.Dense(1)(x)

model = tf.keras.Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
history = model.fit(X_train, y_train, epochs=300, batch_size=8, verbose=1, validation_split=0.2)
```

Figure 3: Code for constructing an ANN model

The model consists of an input layer and several hidden layers with 256, 128 and 64 neurons. The ReLU activation function is used for all layers except the output layer. The Adam optimizer and the Mean Squared Error (MSE) loss function are used to train the network. The model is trained for 300 epochs with the training data.

4.4 Model Evaluation, Simulation of Changes, and Optimization

Once the artificial neural network (ANN) has been successfully trained, it is applied to test data to predict results. In this phase, the Mean Squared Error (MSE) is calculated, which serves as a measure of the effectiveness of the model (Figure 4). MSE is defined as the average squared difference between the actual and predicted values and allows the accuracy of the model to be quantified. A lower MSE value indicates a better fit of the model, while a higher value may indicate problems with overfitting or insufficient learning. In this case, the calculated Mean Squared Error (MSE) after successful training of the artificial neural network (ANN) is 25.86, indicating that there is a significant difference between the actual and predicted values, indicating the need for further optimization to improve prediction accuracy.

```
from sklearn.metrics import mean_squared_error

# Predict and evaluate
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print("True values:", y_test)
print("Predicted values:", y_pred.flatten())
```

Figure 4: Code for predicting and evaluating an ANN model with Mean Squared Error

To further illustrate the accuracy of the model, the results are visualized on a graph that compares the actual values with the predicted values. This visualization allows for easy identification of discrepancies between expected and actual results and thus a quick assessment of the quality of the model. In cases where the MSE is too high, an analysis should be performed to determine which hyperparameters of the model such as the number of layers, the number of neurons in each layer or the chosen optimizer need to be adjusted. This will improve the model's ability to adapt to true data and predict optimal parameters in the context of bookbinding (Figure 5).

```
import matplotlib.pyplot as plt

# Visualize results
plt.figure(figsize=(10, 6))
plt.plot(range(len(y_test)), y_test, 'o-', label='Actual values')
plt.plot(range(len(y_pred)), y_pred.flatten(), 'x-', label='Predicted values')
plt.xlabel('Index')
plt.ylabel('Time (hours)')
plt.title('Actual vs Predicted Values')
plt.legend()
plt.grid(True)
plt.show()
```

Figure 5: Code for visualizing true and predicted values

After the model evaluation, the research continues with the simulation of different input values to analyse how these changes affect the predicted production time. In this simulation, random samples are generated within predefined ranges for the relevant production parameters (Figure 6). The model is then used to predict how these parameters will affect the total time required to complete the production process. This approach enables a better understanding of the dynamic relationships between input variables and output results, which can help optimize the entire process.

```python
# Simulating changes in input parameters
def simulate_changes():
    num_samples = 50
    param_ranges = {
        'cutting': (50000, 60000),
        'folding': (10000, 20000),
        'gathering': (5, 15),
        'binding': (4, 8),
        'trimming': (1, 2)
    }

    samples = []
    for _ in range(num_samples):
        sample = [np.random.uniform(*param_ranges['cutting']),
                  np.random.uniform(*param_ranges['folding']),
                  np.random.uniform(*param_ranges['gathering']),
                  np.random.uniform(*param_ranges['binding']),
                  np.random.uniform(*param_ranges['trimming'])]
        samples.append(sample)

    samples = np.array(samples)
    samples_scaled = scaler.transform(samples)

    predictions = model.predict(samples_scaled)
    return samples, predictions

samples, predictions = simulate_changes()
```

Figure 6: Code for simulating changes in input parameters

The simulation results are also visualized to easily interpret the effects of the different parameters on the predicted time. Each subplot in this visualization shows the relationship between an input parameter and the predicted time, allowing the identification of key factors that significantly affect production efficiency. This information is crucial for decision makers as it helps to direct resources to the most important aspects of the process.

In addition, the predicted results can be used to assess how effectively the model fits the true data. To further optimize the results, the objective function is used to calculate the predicted time for a given set of parameters. At this stage, the minimize function from the SciPy library plays an important role as it enables the identification of optimal parameter values that minimize the predicted time (Figure 7). This procedure leads to the identification of the optimal parameters and the corresponding minimized times, which can significantly increase efficiency and reduce production costs.

```python
from scipy.optimize import minimize

# Optimization using SciPy
def objective_function(params):
    params_scaled = scaler.transform([params])
    predicted_time = model.predict(params_scaled)
    return predicted_time[0][0]

# Initial parameter values (approximately)
initial_params = [54000, 15000, 8, 6, 1]

# Optimization
result = minimize(objective_function, initial_params, method='Nelder-Mead', options={'disp': True})

print("Optimal parameters:", result.x)
print("Minimized predicted time:", result.fun)
```

Figure 7: Code for predicting and assessing model performance

The simulation results are also visualized to easily interpret the effects of the different parameters on the predicted time (Figure 8). Each subplot in this visualization shows the relationship between an input parameter and the predicted time, allowing the identification of key factors that significantly affect production efficiency. This information is crucial for decision makers as it helps to direct resources to the most important aspects of the process.

```
# Visualizing changes
plt.figure(figsize=(12, 8))
plt.subplot(2, 3, 1)
plt.scatter(samples[:, 0], predictions, label='Predicted Time vs Cutting')
plt.xlabel('Cutting')
plt.ylabel('Predicted Time (hours)')
plt.title('Impact of Cutting')
plt.legend()
plt.grid(True)

plt.subplot(2, 3, 2)
plt.scatter(samples[:, 1], predictions, label='Predicted Time vs Folding')
plt.xlabel('Folding')
plt.ylabel('Predicted Time (hours)')
plt.title('Impact of Folding')
plt.legend()
plt.grid(True)

# (Continue for other parameters)
```

Figure 8: Code for visualizing changes

The graphical visualization on Figure 9 shows a comparison between the true and predicted values, allowing a clear assessment of the accuracy of the model by visually representing the differences between the expected results and the outputs of the model. The predicted results can be used to assess how effectively the model fits the true data and predicts the optimal parameters. If it turns out that the error is too high, it is recommended to adjust the hyperparameters of the network, including the number of layers, neurons and optimizer.

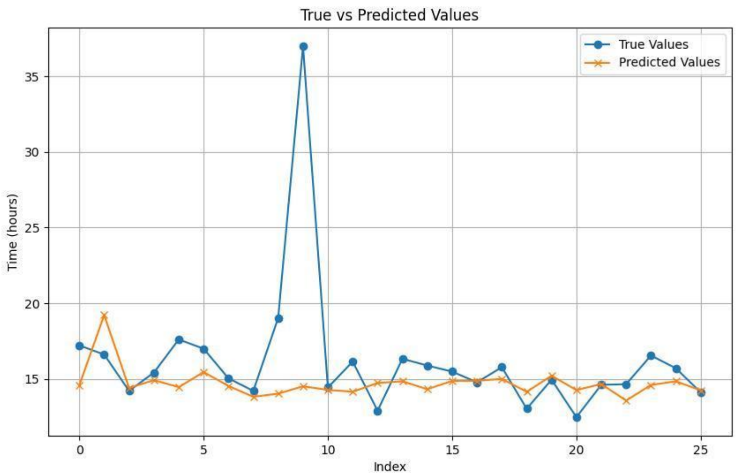

Figure 9: Comparison of true and predicted values

The graph shows comparisons between the true and predicted values of the time required for the production process, based on input parameters such as cutting time, folding time, totalizing time, binding time, trimming time, number of sheets, number of employees and number of orders. This analysis aims to evaluate the effectiveness of the artificial neural network (ANN) developed based on the given input variables.

The X-axis represents the indices of the samples created in the case analysis, while the Y-axis shows the time in hours, which is crucial for evaluating the efficiency of the production process. The blue line represents the actual values of the processing time, while the orange line illustrates the predicted values of the model.

The graph shows significant deviations between the actual and predicted values, especially for index 10, where the actual value was 37 hours while the predicted value was only 18 hours. This deviation may indicate unforeseen complications in the process, such as increased resource requirements or changes in the production flow that were not accounted for in the model, such as suboptimal binding times or longer summation times.

When developing the model, it is important to consider the input parameters and their impact on the predicted time. The binding and folding times showed a significant impact on the total processing time, which means that further analysis should focus on more accurate measurement and optimization of these parameters. If the model does not account for variations in these key factors, major deviations in predictions may occur. Therefore, it is recommended to perform additional iterations of the model to adjust the hyperparameters (e.g. the number of layers, the number of neurons or the choice of optimizer) that could improve the prediction accuracy.

When analysing the production process data, the plot of true and predicted values is an important visualization that provides insight into the accuracy of the artificial neural network (ANN) model relative to the true data. When discussing the overlap of points in this graph, it is important to consider several aspects. The overlap of points indicates situations where the actual and predicted values are in the same or very similar positions in the graph, suggesting a high accuracy of the predictions. For example, if the points representing the actual and predicted values at indices 0, 1 and 2 are almost identical, this means that the model has successfully predicted the processing times for these samples. This may mean that the input parameters used in this part of the model were selected correctly or that the production conditions were stable and consistent.

In places where the points do not overlap or are clearly separated, such as at index 10, where the actual value differs significantly from the predicted value, this may indicate that unexpected problems or processing time deviations have occurred that were not accounted for in the model. For example, longer binding times or delays in summation may have significantly affected the overall time. In cases where the points are far apart, additional data may need to be collected to allow for better modelling. It is possible that the model does not account for all relevant input variables or interactions between them that affect the final result. If there is a major overlap between the actual and predicted values, one can rely on the model's ability to predict future outcomes given similar input data. If there is no overlap, this can help identify areas that require further investigation or model adjustments. This includes reviewing the input variables used, considering new features or redefining existing features.

When analysing the performance of production lines, it is important to consider the specific characteristics of each of the four case studies: SS_A, SS_B, SS_C, and SS_D. These studies illustrate different aspects of the production process and analysing them helps us understand how input parameters and operational specifics affect the actual and predicted values of processing time.

The SS_A line represents a more complex operation involving a multi-level drive, which results in an increase in the time required to transport the semi-finished products. It is expected that this increase in complexity will lead to higher actual production time values as additional losses, e.g. related to transportation, will affect the overall efficiency of the process. The graph clearly shows a pronounced peak around index 9, which represents the moment when the actual values of processing time are significantly higher than the predicted ones. This deviation shows that the model has difficulty predicting the time for more complex processes such as SS_A. The inadequately predicted time can be attributed to the fact that the model does not recognize all the factors that contribute to increased processing time, such as a higher number of sheets, additional time for folding, binding and trimming. This finding emphasizes the need to improve the model to account for all relevant aspects of complex production lines.

The SS_B line is characterized by faster production cycles and a flexible schedule, which should lead to reduced processing times and less performance variation. The graph shows that the predicted values for SS_B generally match the actual values. This high degree of agreement indicates that the model effectively predicts processing times for standardized operations. Minimal deviations between the predicted and actual times indicate that the input parameters, such as the folding time (t_s) and the summation time (t_c), are optimally set for this type of operation. This analysis confirms that the model can work successfully with simpler and standardized processes and provides useful information for further planning and optimization.

Since the SS_C line is the fastest and most efficient line, it is expected to have the shortest processing time. The model is expected to predict relatively consistent and low values for this line, which is optimized for maximum efficiency. The graph clearly shows that the predicted values are very close to the actual values, indicating that the model is effectively tracking the operations on this line. This agreement can be attributed to optimized input parameters such as a low binding time (t_b) and trimming time (t_t). This example shows that the model is able to accurately estimate the processing time in situations where the processes are standardized and efficient and illustrates the importance of selecting and configuring the input parameters correctly.

The SS_D line represents a standard process, but with an optimized schedule that should result in moderate processing times. The model also shows a close match between predicted and actual values in this line. This close agreement indicates that the optimized schedules have contributed to the predicted times matching the actual times. As there are no significant deviations, it can be concluded that the input

parameters for SS_D are optimally set, which confirms the effectiveness of the model in predicting processing times for standard processes.
Input parameters such as the number of sheets per book (*NS*), the folding time (t_s), the summing time (*tc*), the binding time (t_b), the trimming time (t_t), the labour input (*R*) and the production units (*P*) play a crucial role in determining the total production time and significantly affect the prediction accuracy of the model. In the SS_A line, the high number of bends and the increased times for folding and binding explain the significant deviations between the actual and predicted times. For example, the increased binding time can significantly affect the overall processing time, especially in the context of a multi-story operation where transportation losses are high. This analysis emphasizes the need for further investigation and optimization of certain parameters to reduce variation and increase efficiency. The SS_C line, which uses seamless binding, demonstrates how optimized processes can lead to the lowest processing times. Low binding and trimming times contribute to the successful predictions of the model and underline the importance of correct input parameter settings for standardized processes.
Based on the analysis of the graphical representation of the actual and predicted values, important insights can be gained on how the specific characteristics of each line affect the performance of the model. The model shows high accuracy in predicting processing times for standardized and optimized lines, such as SS_B, SS_C and SS_D, while larger deviations occur for more complex processes such as SS_A. These results provide valuable information for future process optimization.
The practical application of the results of this model can make an important contribution to production optimization. For example, flexible planning according to the "*just-in-time*" principle helps to reduce inventory and optimize resources. However, to achieve maximum efficiency, it is advisable to further refine the model to better adapt it to complex lines such as SS_A. This may mean generating more data for complex lines and taking into account additional factors that slow down production, such as transportation losses and specific challenges that occur in such environments.
This analysis not only provides useful insights for optimizing production lines, but also underlines the importance of continuous improvement and model adaptation to achieve the best results in modern production processes.

5. CONCLUSIONS

The use of artificial neural networks (ANN) brings significant improvements in the optimization of production processes, resulting in higher efficiency, lower costs and greater flexibility in adapting to market demand. Analysis of case studies has shown that ANNs can significantly reduce production time and improve machine scheduling, ensuring high product quality.
Based on the analysis of the results of processing time prediction in book production, the Mean Squared Error (MSE) is calculated to be 25.86, indicating a significant difference between the actual and predicted values. This high MSE indicates that the model did not learn sufficiently well, possibly due to the small size of the data set. To improve the accuracy of the model, it is recommended to increase the size of the dataset and fine-tune the hyperparameters such as the number of epochs and the stack size.
This deep learning-based model can be further adapted to the specific requirements of different production lines and enables continuous monitoring and optimization in real time. Further development may include advanced optimization methods and quality monitoring systems that improve the model's ability to adapt to changing production conditions. Overall, the application of artificial intelligence in bookbinding offers significant potential to increase efficiency, reduce costs and improve the quality of the end product, ensuring competitiveness in the market.
In addition, this model can serve as a basis for more complex optimization models specifically tailored to the bookbinding industry. More complex architectures such as RNN or LSTM networks for time-dependent data can be added and methods such as genetic algorithms or Bayesian optimization can be implemented for further parameter optimization.

6. REFERENCES

Aggarwal, C. C. (2018) *Neural Networks and Deep Learning: A Textbook*. Yorktown Heights, Springer International Publishing.

Brajković, T., Perinić, M. & Ikonić, M. (2018) Production planning and optimization of work launch orders using genetic algorithm. *Technical Gazette*. 25 (5), 1278-1285. Available from: doi: 10.17559/TV-20161207195125

Bratić, D., Pasanec Preprotić, S. & Jurečić, D. (2023) Bookbinding process optimization using ANN. In: Özcan, A., Tutak, D., Arman Kandirmaz, E., Özomay, Z., Oğuz, M. & Sesli, Y. (eds.) *Proceedings of the 4th International Printing Technologies Symposium PrintIstanbul 2023, 5-6 October 2023, Istanbul, Turkey.* Printing Industry Education Foundation (BASEV), pp. 331-336.

Cetinjanin, A. (2020) *Planiranje poslovanja proizvodnih procesa u malom i srednjem poduzetništvu*. Master thesis. University of Zagreb, Faculty of Graphic Arts.

Dasović, E., Petković, G. & Pasanec Preprotić, S. (2015) Oblikovanje i budućnost knjižnog uveza u svijetu e-knjige. *Technical Journal*, 9 (4), 440-445.

Ertel, W. (2017) *Introduction to Artificial Intelligence*. Cham, Springer International Publishing.

Gašparić, S., Petković, G. & Pasanec Preprotić, S. (2018) Critical analysis of marketing in Croatian publishing. *Acta Graphica.* 28 (3), 93-100. Available from: doi: 10.25027/acta%20graphica.v28i3.142

Haykin, S. (2009) *Neural Networks and Learning Machines*. New Jersey, Pearson.

Huang, B., Gopaluni, B., Tulsyan, A., Chachuat, B., Lee, J. M., Amjad, F., Damarla, S. K., Woo, J. H. & Lawrence, N. (2020) Modern machine learning tools for monitoring and control of industrial processes: A survey. In: Findeisen, R., Hirche, R., Janschek, K. & Mönnigmann, M. (eds.) *Proceedings of the 21st IFAC World Congress 2020, 11-17 July 2020, Berlin, Germany*. IFAC, pp. 218-229.

Pasanec Preprotić, S., Stančin, D. & Petković, G. (2022) Projektiranje procesa u nakladničkom uvezu knjiga - Analiza radnog procesa. *Polytechnic & Design.* 10 (4), 222-232. Available from: doi: 10.19279/TVZ.PD.2022-10-4-01

Pasanec Preprotić, S., Vukoje, M., Petković, G. & Rožić, M. (2022) Sustainable approach to book designing concepts in bindery sector: An overview. In: Vladić, G. (ed.*) Proceedings of the 11th International Symposium on Graphic Engineering and Design GRID 2022*, 3-5 November 2022, Novi Sad, Serbia. University of Novi Sad, Faculty of Technical Science, Department of Graphic Engineering and Design, pp. 629-645.

Pasanec Preprotić, S., Vukoje, M., Petković, G. & Rožić, M. (2023) Novel Approaches to Enhancing Sustainable Adhesive System Solutions in Contemporary Book Binding: An Overview. *Heritage*. 6 (1), 628-646. Available from: 10.3390/heritage6010033

Qamar, R. & Zardari, B. A. (2023) Artificial Neural Networks: An Overview. *Mesopotamian Journal of Computer Science*. 1, 130–139. Available from: doi: 10.58496/MJCSC/2023/015

Russell, S. & Norvig, P. (2020) *Artificial Intelligence: A Modern Approach*. Hoboken, Pearson.

Voulgaris, Z. & Bulut, Y. (2018) *AI for Data Science: Artificial Intelligence Frameworks and Functionality for Deep Learning, Optimization, and Beyond*. Basking Ridge, Technics Publications.