CHAPTER 6

DEVELOPMENT OF DEEP LEARNING MODEL FOR INK SELECTION IN PRINTED ELETRONICS

6.1 INTRODUCTION

PE is an emerging technology that involves the production of electronic components and devices using printing techniques. This technology has generated interest in recent years. Due to its potential applications in healthcare, automotive, consumer electronics and energy, the key ingredient in PE is conductive ink. Conductive ink is made from materials such as silver, copper and carbon, allowing it to be printed on the surface of conductive pipes, electrodes and circuits. These inks play an important role in maintaining electrical conductivity within PE devices to facilitate signal transmission and power transmission. Selecting the right conductive ink for a specific PE application is a complex task influenced by many factors, such as material characteristics. printing techniques and considering costs. Deep learning models by analyzing large data sets of material properties. Performance indicators and terms of use play an important role in this process. These models use multiple layers to achieve complex patterns and relationships between input and output data by training on a labelled dataset containing information about user preferences. Deep learning models can effectively predict the suitability of a specific ink flow for a given application.

This phase of research work aims to build a deep learning model to choose suitable conductive ink for printing applications based on input data. The introduced methodology involves the construction of deep learning model for conductive ink selection, followed by the optimization of hyperparameters with training algorithm. Furthermore, an evaluation and comparison a comparison of the developed model's performance with previous approaches is made.

6.2 DEEP NEURAL NETWORK

Deep learning refers to a category of machine learning models that organize themselves as neural networks, using a sequence of layers to analyze data and provide forecasted results for novel input (Alzubaidi et al. 2022). These models use multiple layers to analyze and interpret input data. Recently, deep learning models have found applications across various domains such as pattern recognition, segmentation, object detection, computer vision, video processing, classification, and more.

Among the widely adopted deep learning models, CNNs stand out for their efficacy in many tasks due to its characteristics of utilizing filters to capture geographical and temporal relationships in the data (Polaamuri et al. 2022). A CNN often has several layers, which include subsampling layers, dropout layers, convolutional layers, and non-linear processing units. A feedforward multi-layered hierarchical network is called a CNN. Each CNN layer uses a bank of convolutional kernels to perform several modifications. Convolution is a method that makes it easier to extract spatial data points provide valuable qualities. The process is a linear procedure that entails encoding the multiplication involving inputs with a specified set of weights. One may represent the convolution procedure as,

$$x = A * F \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6.1)$$

Where,

A : Input data

F: kernel

x : Convolution layer output

* : convolution operation

Figure 6.1 shows an example of convolution operation. In the convolutional layer, the matrix on the left serves as the input, the matrix in the center serves as the kernel, and the matrix on the right serves as the output of the convolutional layer. The kernel is convolved over the input data to bring about the output. The training of the networks essentially allows the weights that are employed in the kernels of these convolutional layers to be learned.

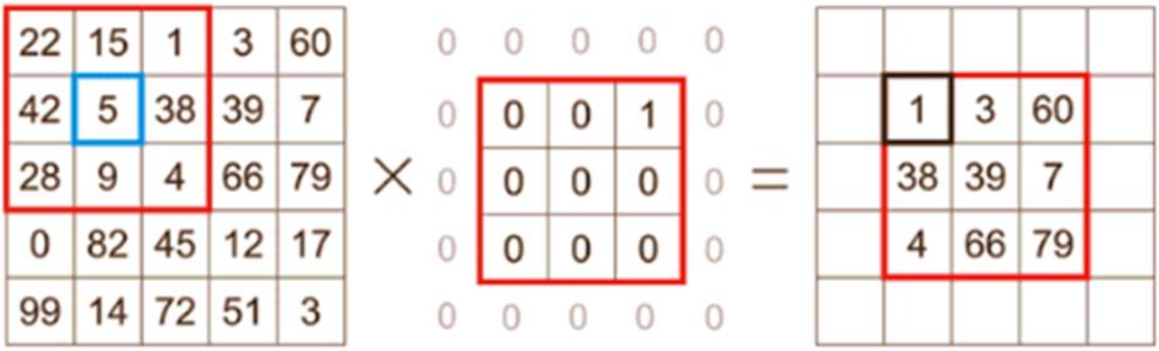

Figure 6.1 Convolution operation

The Rectified Linear Unit (ReLU) activation function receives each value subsequently after the feature map has been determined. As shown in Figure 6.2, the ReLU function is a linear activation function that, if the input is negative, makes the input equal to zero; otherwise, it leaves the input unchanged. The vanishing gradient issue is solved by the ReLU activation function, which also enables the model to perform more effectively, learn from the training data faster, and improve its overall performance. The ReLU activation function can be represented using Equation (6.2),

$$R(y) = \max(0, x) \quad \text{.................................(6.2)}$$

Where,

x: the activation function receiving input

R(y) : output of the activation function.

The non-linear activation function's output is often followed by the subsampling procedure, which helps to summarize the findings and renders the input resistant to geometric distortions. In the research, 1D CNN is designed to choose an appropriate ink for printing applications.

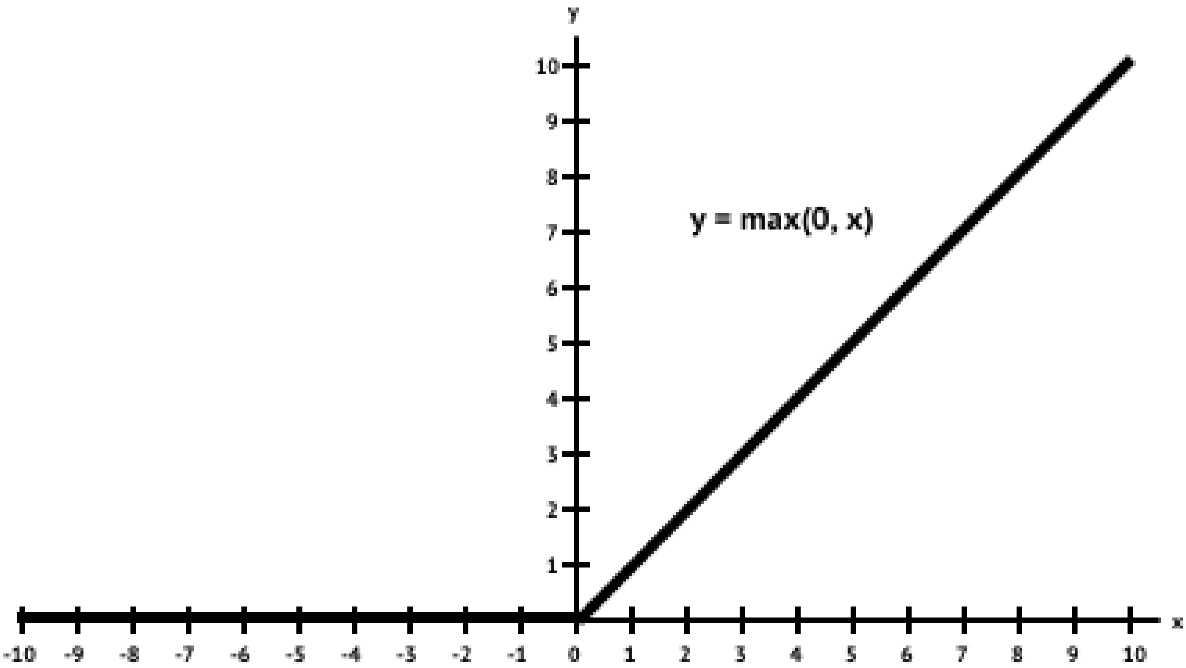

Figure 6.2 ReLU activation function

6.3 PROPOSED METHODOLOGY FOR INK SELECTION

This research phase is primarily focused on developing an automated method that uses a deep learning model to choose conductive ink for PE applications. The developed ink selection model is represented in Figure 6.3, which is a schematic illustration of the modelling process. The introduced model operates via two key phases: training and testing phases. In the traini3ng phase, the process starts with the collection of data, which encompasses all variable pertinent to the printing process. This comprehensive dataset facilitates the identification of both input and output variable crucial for the development of ink selection system. Following data collection, a preprocessing step is implemented to refine the gathered data to ensure its suitability for subsequent processes. Subsequently, a 1D CNN is designed and trained to capture the intricate relationship between input and output variables. During testing phase, the trained model is employed to select ink based on unseen data. In the context of real-time applications, the purpose of this phase is to assess the effectiveness and generalizability of the model that was constructed. Through testing, any shortcomings in the model's performance are identified and addressed to ensure its reliability and effectiveness in practical applications.

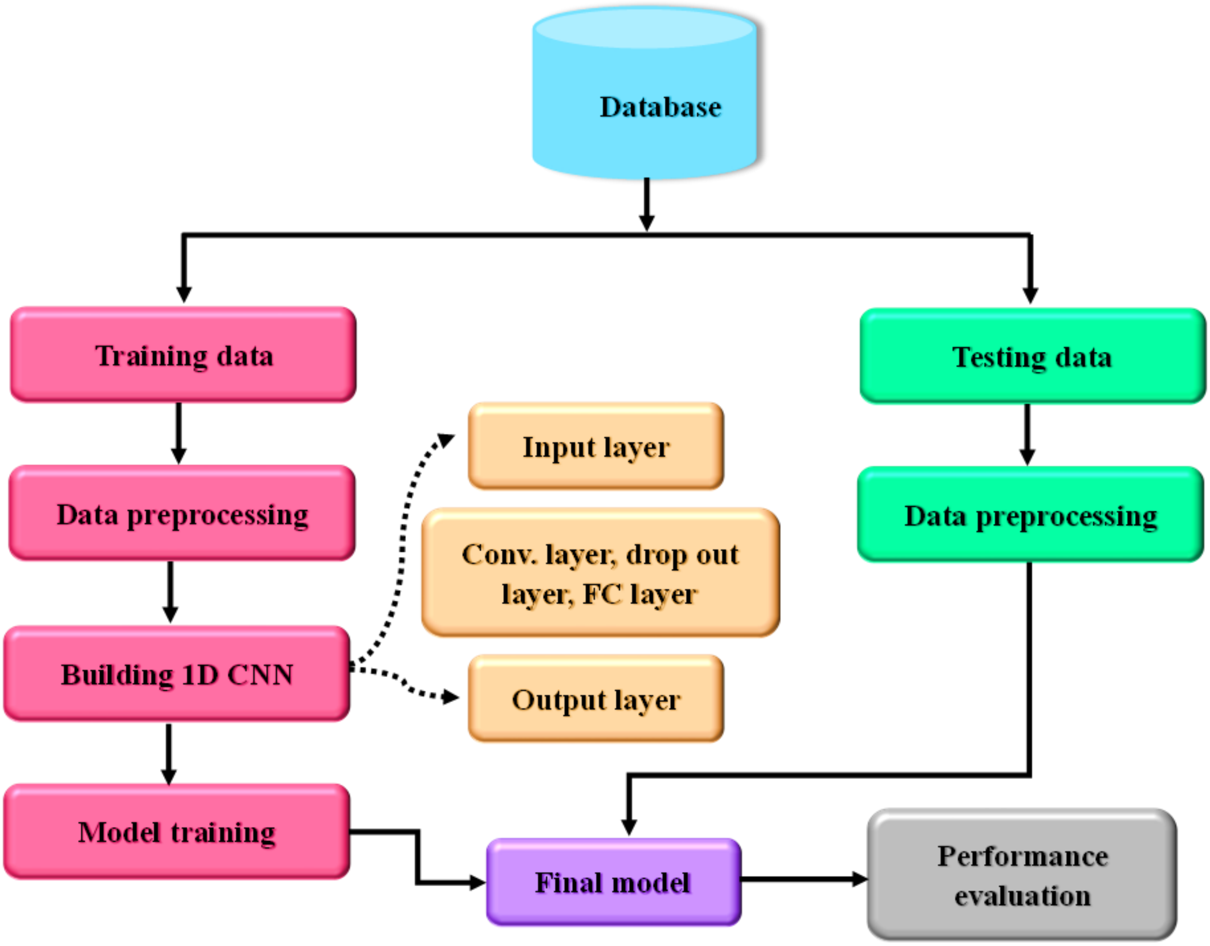

Figure 6.3 Pipeline of the developed ink selection model

The developed model involves several crucial processes including:

- Data collection
- Data partition
- Data normalization and
- Modelling deep learning model

6.3.1 Data Collection

The first step in the process involves gathering both input and output variables essential for subsequent analysis. Data collection plays a significant role in ensuring availability of datasets crucial for training and testing the model. Data is collected based on the characterization of material properties and the exploration of conductive inks. The deep learning model leverages these material attributes to discern the most suitable ink for printing applications. Specifically, the investigation delved into three primary varieties of conductive inks: carbon, copper, and silver inks. In ensuring the efficacy of printing operations, a comprehensive assessment of various material characteristics becomes imperative. Print performance is affected by product life, quality, handling and use, GSM, calliper/thickness, brightness, tear resistance, and moisture content.

6.3.2 Data Division

Following data collection, the gathered data is partitioned into two distinct subsets. The first subset is known as training data. The training forms the basis for training and enables the system to learn patterns and relationships within the input and output data. Meanwhile, the second subset is named testing dataset. The testing data is solely used for assessing the performance of the model and its generalization capabilities. The data is split into two sets using a 70:30 ratios for the training and testing sets.

6.3.3 Data Normalization

Preprocessing is conducted via data normalization method. Data normalization is a crucial step aimed at standardizing the data to a uniform scale typically ranging from 0 to 1. It facilitates optimal model performance by mitigating the effects of varying scales and ensuring the uniformity across different features, thereby improving the model's ability to converge during training phase. The min-max method is employed for normalization according to the following formula:

$$x_{norm} = \frac{y - \min(y)}{\max(y) - \min(y)} \quad \text{.....................................(6.3)}$$

Where,

x : Original value

x_{norm} : Normalized value

min(x) : Minimum value and

Max(x) : Maximum value

The normalized data is then reshaped into three dimensions, corresponding the row, column, and channel.

6.3.4 Modelling Deep Learning Model

The proposed 1D CNN consists of several distinct layers to process input data, extract relevant features, and generate predictions. The model's capacity to learn complicated data patterns and correlations depends on each layer. Figure 6.4 illustrates the structure of the developed 1D CNN.As shown in Figure 6.4, the 1D CNN architecture includes an input layer, followed by convolutional layers, ReLU activation function, a flatten layer, dropout layer, a Fully Connected (FC) layer, and an output layer. The transition from the input layer to the output layer involves forward propagation, where input data is processed to extract features and produce predictions. Concurrently, parameters adjustments like kernels and weights propagate from output to the input layer is known as backpropagation. Both forward and backward propagation mechanisms are used during the training phase to optimize model parameters and minimize the loss function. The hyperparameters governing the architecture of the developed 1D CNN are detailed in Table 6.1.

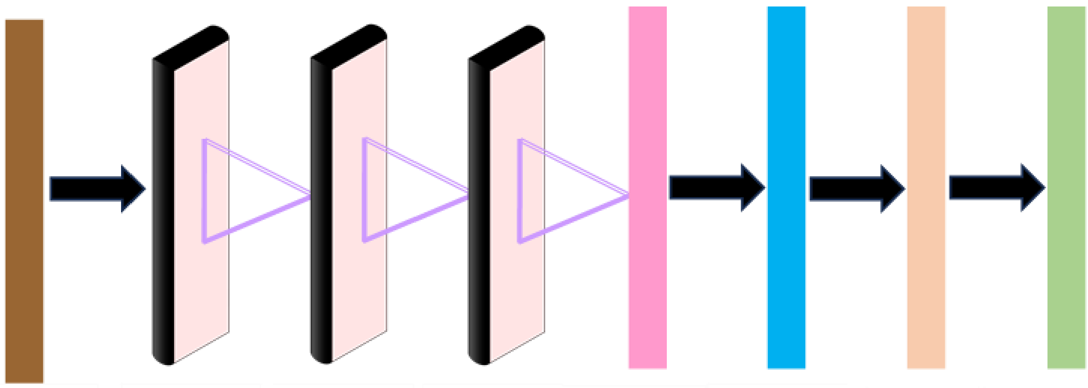

Figure 6.4 Structure of the developed 1D CNN

Table 6.1 Hyperparameters of the developed 1D CNN

Parameters	**Output shape**	**No. of filters/ Kernel size /stride**
Input layer	100x8x1	-
Convolutional layer 1	100x6x16	16/3, 1
Convolutional layer 2	100x4x16	16/3,1
Convolutional layer 3	100x2x16	16/3,1

Flatten layer	100,32	-
Drop out layer	0.2	-
Fully connected layer	128	Linear
Classification layer	3	Softmax

Input layer

The raw input data, usually represented as a 1D array, enters the system via the input layer. In this specific architecture, the input data has a shape of 100 X 8 X 1, indicating the dimension of the input data

Convolutional layers (Conv.layers)

Feature extraction via the application of filters is the responsibility of the convolutional layers, which conduct convolutions on the input data. In this architecture, three convolutional layers are used to generate feature maps.

ReLU function

The output feature maps use the ReLU activation function element-wise after each convolutional layer. The non-linearity that was provided by ReLU to the network made it possible for it to learn more efficiently the complicated connections that were within the data.

Flatten layer

To obtain the output from the convolutional layers ready for input into the succeeding FC layer, the flattened layer is responsible for reshaping it into a single vector. This transformation converts the 2D feature maps into a 1D array while preserving the values.

Dropout

To reduce overfitting and enhance the generalization capabilities of the CNN, a dropout layer is used after the third convolutional layer. Dropout layer randomly deactivates a fraction of neurons during training to improve performance.

FC layer

The input vector from the previous layer is flattened and sent to the fully connected layer, which applies biases after matrix multiplication using a set of learned weights. This layer facilitates learning complex combinations of features and prepares the data for classification task.

Output layer

The final CNN layer generates output based on learned representations from the previous levels. In this architecture, softmax activation function is used to compute probabilities across labels.

Training phase

During the training phase, the developed CNN model undergoes training using training dataset. In this process, to reduce prediction errors and optimize performance, the model repeatedly adjusts its parameters to represent the underlying patterns and connections in the data.

Testing phase

Following training process, the efficacy and generalization capabilities of the trained model are evaluated using the testing dataset. The predicted accuracy and robustness of the model are evaluated at this essential stage of assessment. Table 6.2 provides the pseudocode for the constructed 1D CNN model.

Table 6.2. Algorithmic procedure of the developed model

Step 1: Gather input data and output labels
Step 2: Pre-process the data to a common range [0,1]
Step 3: Split the data into training and testing data
Step 4: Design a CNN and initialize its hyperparameters
Step 5: for each iteration do
Step 6: Fit the model utilizing training samples
Step 7: Calculate the error between actual and predicted labels
Step 8: Back propagate error and adjust the hyperparameters of the CNN
Step 9: if loss is better do
Step 10: Save the network
Step 11: end if
Step 12: Tune the hyperparameters
Step 13: end for
Step 14: Test the network using testing data
Step 15: Analyse the model's performance

6.4 EXPERIMENTAL RESULTS

In the MATLAB2022a environment, the suggested system has been put into operation. In the quantitative results that were generated by the ink selection system that was developed for printing applications, this part provides a complete examination. Using a comprehensive examination of the numerical results, this section seeks to offer valuable insights into the efficiency and reliability of the ink selection system.

6.4.1 Evaluation Metrics

A comprehensive evaluation of the constructed system is conducted by analysing many metrics, including recall, precision, accuracy, and F1-score. TP signifies instances where the system correctly identifies the positive class, while TN represents accurate identification of negative class. FP occurs when the system incorrectly identifies the positive class, and FN results from misclassifying the negative class. It is important to highlight that this research entails multiclass classification. The TP category serves as the focal point for computation, while the remaining labels are encompassed in the negative category. The metrics are listed below:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad \text{(6.4)}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad \text{(6.5)}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad \text{(6.6)}$$

$$\text{F1} - \text{score} = 2\ \text{X}\ \frac{\text{Precision X Recall}}{\text{Precision+Recall}} \quad \text{(6.7)}$$

$$\text{BCR} = 1/2\ \text{X}(\text{Recall} + \text{Specificity}) \quad \text{(6.8)}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP+TN}} \quad \text{(6.9)}$$

$$\text{MCR} = 1 - \text{Accuracy} \quad \text{(6.10)}$$

6.4.2 Performance Analysis

The CNN developed in this research phase was trained using the Adam optimizer. The Adam optimizer is widely used training algorithm known for its effectiveness. Figure 6.5 illustrates the accuracy plot of the developed model throughout the training process across various epochs. The gradual enhancement in training accuracy depicted in the plot signifies the continuous refinement of optimized parameters as the model progresses via individual epochs until convergence is achieved. Notably, the remarkable achievement of a high training accuracy of 99.5% at convergence is indicative of an evaluation of the suggested model's efficacy in accurately and consistently identifying the data.

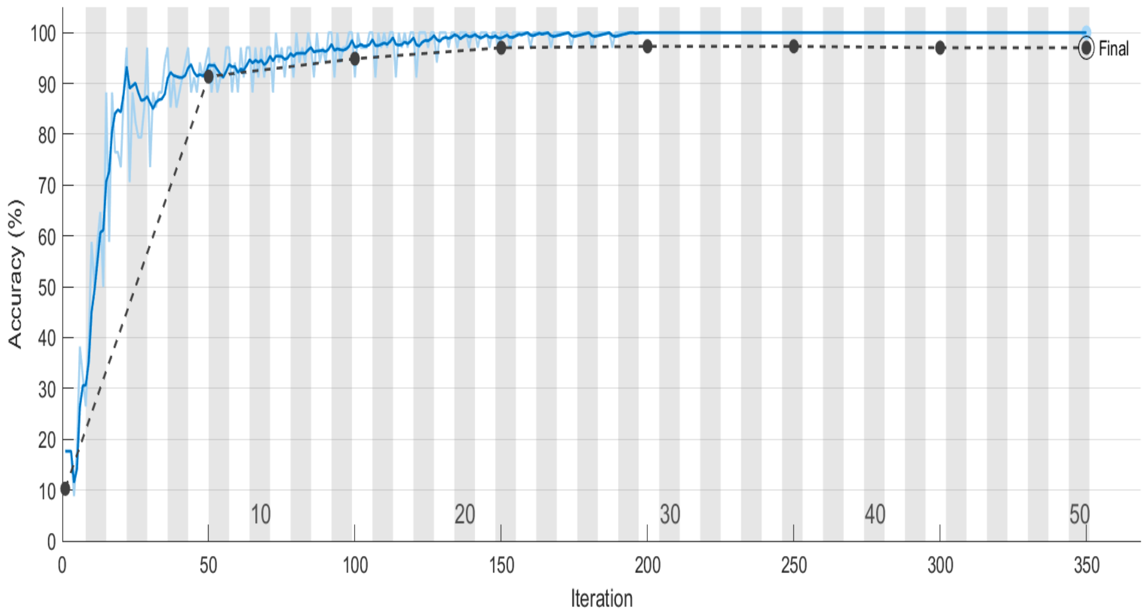

Figure 6.5 Accuracy over epochs

Figure 6.6 depicts the loss plot observed during the training phase. A lower loss value at convergence, as indicated by the training loss of 0.002, signifies a superior-performing model. This low loss value shows that the 1D CNN has successfully minimized error during training, reinforcing it capacity to precisely identify features and patterns in the data.

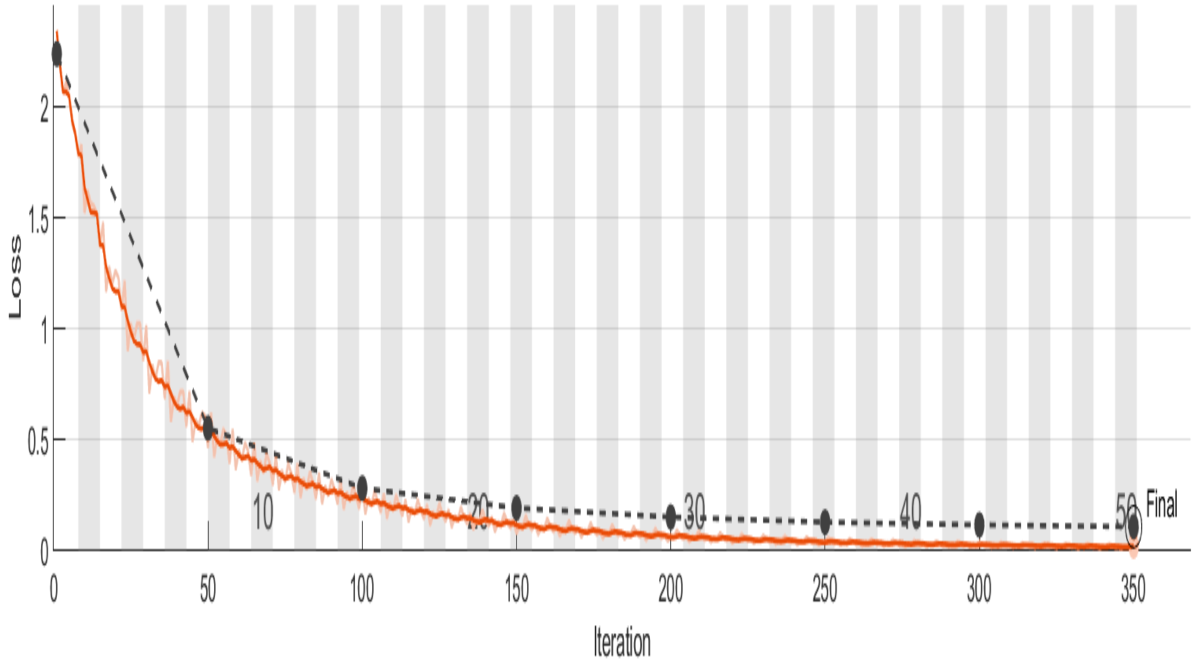

Figure 6.6 Loss over epochs

The optimal parameters of the developed CNN are tabulated in Table 6.3. These parameters were selected following an exhaustive analysis process. The performance of the model is visually represented in Figure 6.7. Upon through examination of Figure 6.7, it becomes evident that the introduced model demonstrated strong and impressive results. The accuracy percentage attained by the model was 98.48%, proving its ability to correctly classify instances. Additionally, the model exhibited a recall of 98.86%, indicating its proficiency in accurately identifying relevant instances from the dataset. Moreover, the model yielded precision of 96.65%, indicates the ratio of accurately detected cases out of all cases categorized as positive. Additionally, the model achieved an F1 score of 97.74%, a BCR of 98.57%, and a low MCR of 1.52, highlighting its overall performance in achieving a balance between precision and recall. These indicators confirm the robustness and effectiveness of the designed model in dealing with the ink selection task.

Table 6.3 Optimized parameters of 1D CNN

Parameter	Value
Epochs	50
Activation	ReLU
Dropout	0.2
Learning rate	0.01
Optimizer	Adam

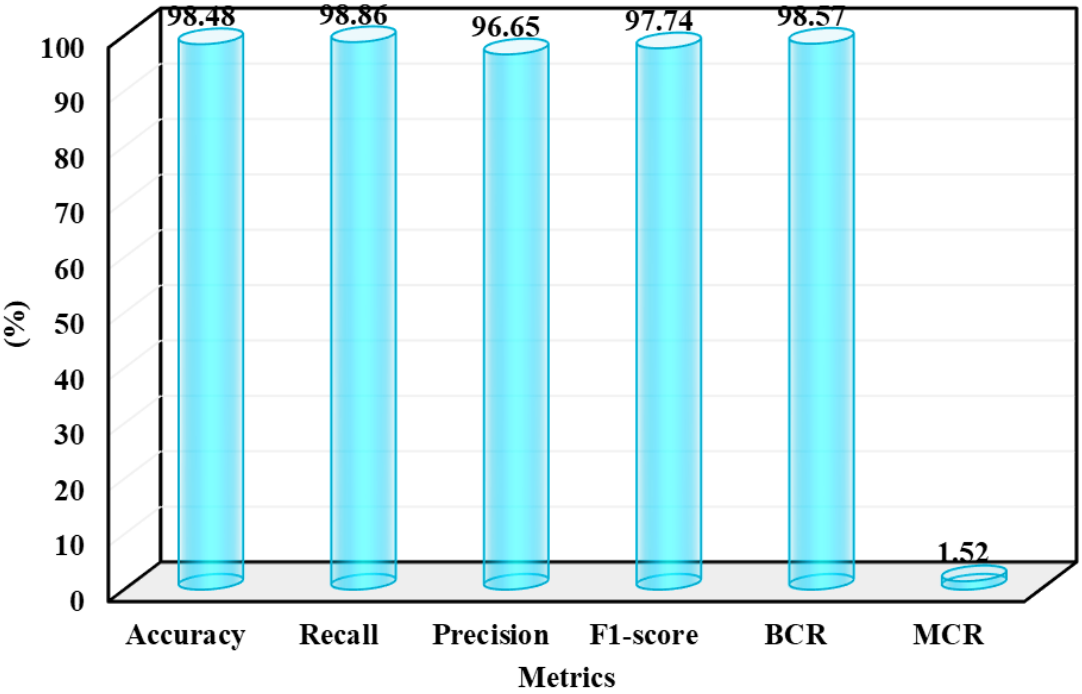

Figure 6.7 Performance of the designed 1D CNN

6.5 CHAPTER SUMMARY

The main objective of this research phase is to create deep learning for selecting suitable conductive inks for printing applications. This chapter briefly introduces deep learning models. The datasets used for the experiments, which were preprocessed using the min-max method, are also summarized. Following preprocessing, a CNN was constructed and trained with Adam optimizer to discern the relationships between input and output variables. The performance of the developed model was evaluated.