Handwritten Digit Prediction using Machine Learning Algorithms

Bipul Roy[1], Udit Kashyap Bhuyan[2]

[1]NIELIT KOKRAJHAR EC,
bipul.roy@nielit.gov.in

[2]NIELIT KOKRAJHAR EC
uditbhuyan99@gmail.com

***Abstract*:**
People's expectations for machines have never been higher; Everything from classifying objects in images to adding sound to silent movies can be done with the help of deep learning (DL) and machine learning (ML) algorithms. Similar to that, handwriting prediction is a significant field of study and development with numerous accomplishments. A computer's capacity to identify and decode handwriting from images, documents, touch screens, and other media is called handwriting recognition (HWR), often referred to as handwriting text recognition (HTR). Obviously, with the aid of the MNIST dataset, we code using Support Vector Machine (SVM), Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN) models in this article. To determine the best-known number, our primary objective is to compare the accuracy and execution times of the aforementioned models. Keywords: deep learning, machine learning, coding, MNIST dataset, support vector machine (SVM), multilayer perceptron (MLP) and convolutional neural network (CNN). Training, machine learning, numerical prediction, MNIST dataset, Support Vector Machines (SVMs), Multilayer Perceptron (MLPs) and Convolutional Neural Networks (CNN).

***Keywords*:** *Machine Learning, Handwritten Digit Prediction, Deep Learning, MNIST datasets, Support Vector Machines (SVM), Multi-Layered Perceptron (MLP), and Convolution Neural Network (CNN).*

(Article history: Received: *Oct,31, 2023,***and accepted** *Dec.8, 2023***)**

I. Introduction

The ability of computers to detect human digits from a variety of sources (images, text, touch screens, etc.) is referred to as the "Digit Recognition" model. It is mostly classified into 10 groups (0–9). This is always an endless search in deep learning. Digital authentication includes license certificates, mail, bank accounts, etc. It has many applications such as. [2]. We face many problems while writing the code because the writing is different for different people because the character has no visual perception. This study presents a comparison of variousML and DL algorithms for coding. SVM, multilayer perceptrons, and CNN are used for this purpose. Comparison of these algorithms was made based on accuracy, error, and training test times and validated with plots created using matplotlib. The accuracy of each model is important because accurate models lead to better decisions. Low-precision models are unsuitable for real-world applications. For example, in bank transactions, the system can determine the amount and date of the check, which is also important. If the system does not recognize the code, unexpected and serious damage will occur. Therefore, real-world applications need advanced algorithms. In order to select the most accurate algorithm with the fewest errors for a variety of coding applications, we thus compare the accuracy of various methods. For coding, this article offers a solid understanding of DL and ML approaches including SVM, CNN, and MLP. It also offers details on the algorithm's performance on tasks involving the recognition of numbers. In the rest of this article, we will talk about the work done in this field, and then we will talk about the process and use of all three algorithms for a better understanding. Then, this article presents the results and conclusions supported by our study. You will also be presented with some future improvements in this area. Notes and references are included in the article's final section.

II. LITERATURE SURVEY

2.1 Handwritten Digit Recognition: The capacity of a computer to read and comprehend readable handwritten input from sources like paper documents, images, touch screens, and other devices is referred to as HWR, or HTR [1].

2.2 Machine Learning: A precursor of some of the first end-to-end contemporary DL models, LeCun et al. concentrated on applying gradient-based learning techniques employing multi-module ML models.

2.3 Natural Language Processing (NLP): By processing and comprehending human language, NLP enables the system to evaluate the coherence, relevance, and quality of the student responses. NLP techniques allow the system to create language models, such as n-grams, or more advanced models like RNNs or transformer models. These models can learn the structure, context, and patterns of the provided answers. By analyzing the language models, the system can identify grammatical errors, logical inconsistencies, or improper use of vocabulary.

2. 4 Recurrent Neural Network (RNN): RNNs are utilized to analyze and evaluate the descriptive answers provided by students. RNNs are a kind of neural network architecture specifically designed to process sequential data, making them well-suited for handling text data like descriptive answers.

2.5 Convolutional Neural Network (CNN): In the context of Natural Language Processing (NLP), CNNs can be used to extract meaningful features from textual input, such as descriptive answers provided by students. These features are then used for tasks such as classification, similarity comparison, error detection, grading, and feedback generation.

III. METHODOLOGY

Comparison of algorithms (SVM, MLP, and CNN) is based on the attributesof each algorithm such as data set, epoch number, algorithm complexity, and the accuracy and characteristics of each algorithm under ideal conditions. equipment (Ubuntu 20.04 LTS, i5 7th generation processor) and runtime algorithm employed to complete the program.

3.1 DATASET

Mobile behavior recognition is a broad field of research with detailed information on how to use it, including large data sets, popular algorithms, benchmarks, and extractionapproaches. high school students and US Census Bureau employees. The NIST repository is made up of two NIST repositories: Special Repository 1 and Special Repository 3. The MNIST repository (Modified National Institute of Standards and Technology) is a subset of this repository. With a 28x28 pixel bounding box and anti-aliasing, MNIST includes 70,000 digital image labels in total (60,000 in the training set and 10,000 in the test set). Each of these images has a corresponding Y value that indicates which number it represents.

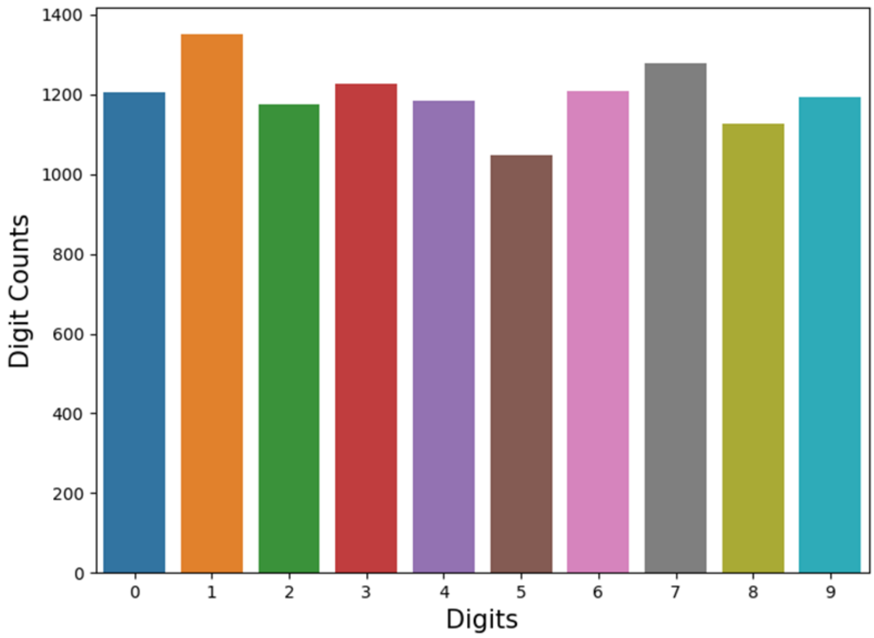

Figure1. Bar chart showing the MNIST dataset (labels containing all training samples).

Figure2. Draw a random MNIST code.

a. SUPPORT VECTOR MACHINE(SVM)

SVM is a supervised MLmodel. Here we usually organize the data in an n-dimensional space where n is the number of features, some coordinates represent the valuesof the features, and perform the division by finding a hyperplane separating the two groups. It will select the hyperplane separating the groups. SVM selects the cloud vectors that help create the plane. These conditions are called support vectors, so the algorithm is called a vector machine. There are two main types of SVM: linear SVM and nonlinear SVM. In this paper, we use linear support vector machines for handwritten digit recognition [10].

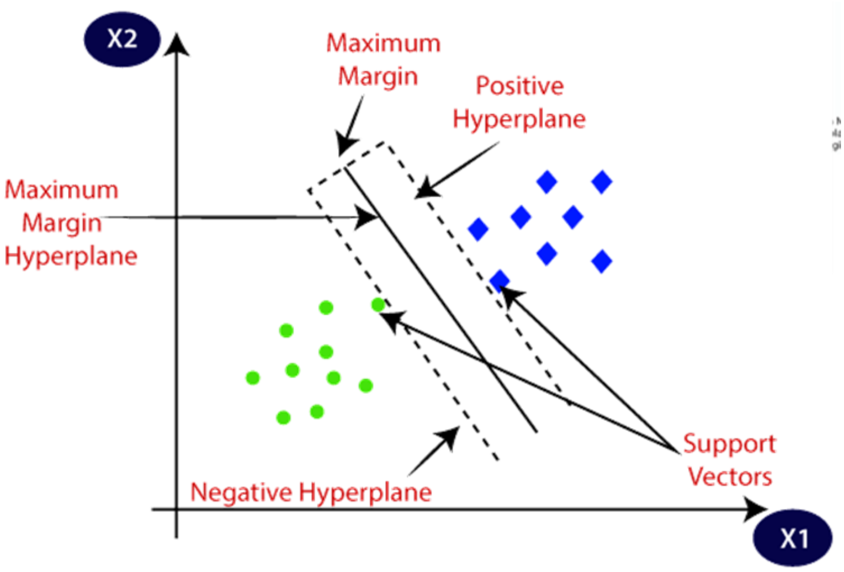

Figure3. This figure explains the working of SVM classification supporting vectors and hyperplanes.

b. MULTI-LAYERED PERCEPTRON(MLP)

Multilayer perceptrons (MLPs) are a type of feed-forward artificial neural network (ANN). This is three layers: the input layer, the hidden layer, and the output layer. Each layer consists of many nodes, also called neurons, and each of these interacts with each of the other layers in the next layer. There are 3 layers in simple MLP, but depending on the problem, the number of hidden layers can increase up to one, and the number of nodes is not limited. The number of nodes in the input layer and output layer depends on the number of features in the dataset and the number of visible groups, respectively. Due to the instability of the model, it is difficult to determine the specific number of primitive layers or the number of nodes in the primitive layer and therefore should be chosen as sim. Each layer of the model can have a different activation function. It uses a supervised learning technique called backpropagation for learning. In MLP, one of the links has a weight that is adjusted during model training to keep it synchronized with all links.

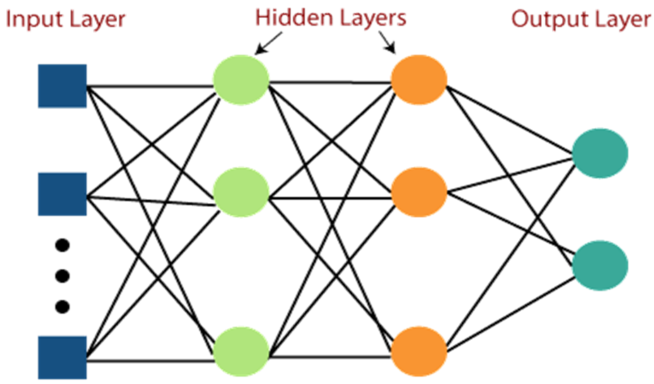

Figure4. This figure shows the basic architecture of a multilayer sensor with different network characteristics.

c. CONVOLUTIONAL NEURAL NETWORK

CNN is a deep learning algorithm widely used in image recognition and classification. It is a type of deep neural network that requires very little preprocessing. It feeds the image in small pieces rather than individual pixels, so the network can better detect obscure patterns (lengths) in the image. CNN has 3 layers consisting of the input layer, output layer, and layer by layer, layer by layer (max pooling and average pooling), all connected layers (FC), and many hidden layers including layer normalization. CNN uses a filter (kernel), one of the weights, to extract features of the input image. CNN uses different functions of each layer to add some parameters. As we move towards CNN, we see that the height and width decrease as the number of channels increases. Finally, the resulting column matrix is used to predict the output.

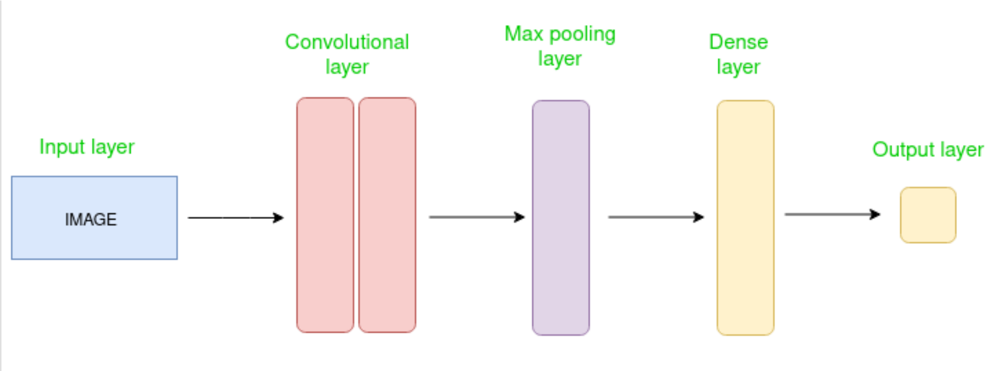

Figure5. This figure shows the architecture of each layer of the CNN in a flowchart.

d. VISUALIZATION

In this study, we use the MNIST dataset (i.e., numerical data) to compare deep learning and machine learning algorithms at different levels (e.g., SVM, ANN-MLP, CNN) in terms of completion time, complexity, accuracy, and number. To see the data obtained by analyzing the algorithm in detail, we use the bar charts and graphs in tabular format of the matplotlib module, which gives us the most accurate view of the algorithm's progress in numerical analysis. Diagrams are provided for each major section of the program to provide a visual representation of each section to support the results.

e. SOFTWARE TOOLS

Python: Python is a popular programming language widely used for machine learning and programming languages. It provides a rich ecosystem of libraries and frameworks that facilitate the development of automated auditors.
TensorFlow: This process builds and trains our models for tasks such as text classification or sequence analysis.
Optical Character Recognition (OCR) Libraries: These libraries help convert text or text into machine-readable format.
Google Colab: It enables collaboration between projects by allowing us to share our books with others.

IV. IMPLEMENTATION

We have discussed in detail the implementation of each algorithm explicitly below to create a flow of this analysis to create a fluent and accurate comparison.We use three different algorithms to compare algorithms based on accuracy, execution time, complexity, and number of challenges (in deep learning): -

- Support Vector Machine Classifier
- ANN - Multilayer Perceptron Classifier
- Convolutional Neural Network Classifier

  We have detailed the use of each algorithm detailed below to create the flow of this analysis so that the comparison is good and accurate.

A. PRE-PROCESSING

Preprocessing is the first step in machine and deep learning and aims to improve input data by reducing unnecessary impurities and reprocessing. To simplify and parse the input data, we convert all existing images in the data into 2D images, for example (28,28,1). Each pixel value of the image is between 0 and 255, so we normalize the pixel value by converting the data to "float32" and dividing by 255.0, making the input properties go from 0.0 to 1.0. Then we do a single-bit encoding to convert the y values into 0 and 1 so that each number has an array, for example, the output value 4 will be converted into an array of 0 and 1, i.e., [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]

B. SUPPORT VECTOR MACHINE

SVM in Scikit-learn [16] supports dense (numpy.ndarray and numpy.asarray variants) and sparse (all scipy.sparse) sample vectors as input. In scikit-learn, SVC, NuSVC, and LinearSVC are classes that can perform various data classifications. In this paper, we use LinearSVC to classify MNIST data using the system used with the help of LIBLINEAR [17]. Various scikit-learn libraries such as NumPy, matplotlib, pandas, Sklearn, and seaborn are used for the implementation. First, we will download the MNIST dataset, then upload it and read this CSV file using pandas. Some patterns are then drawn and converted into matrices, and then the features are normalized and scaled. Finally, we built an SVM model and confusion matrix to evaluate the accuracy of the model [9].

C. MULTI-LAYERED PERCEPTRON

The process of collecting code from a multilayer perceptron (also known as a feed-forward artificial neural network) was done with the help of the Keras module for its robustness. Delay layer to capture a 28x28 pixel image as input. After creating the array, we add Dense layers as in the image below, and display the different layers. A block diagram is provided here for reference. Once you have the training and testing data, you can follow the steps below to train the neural network in Keras..

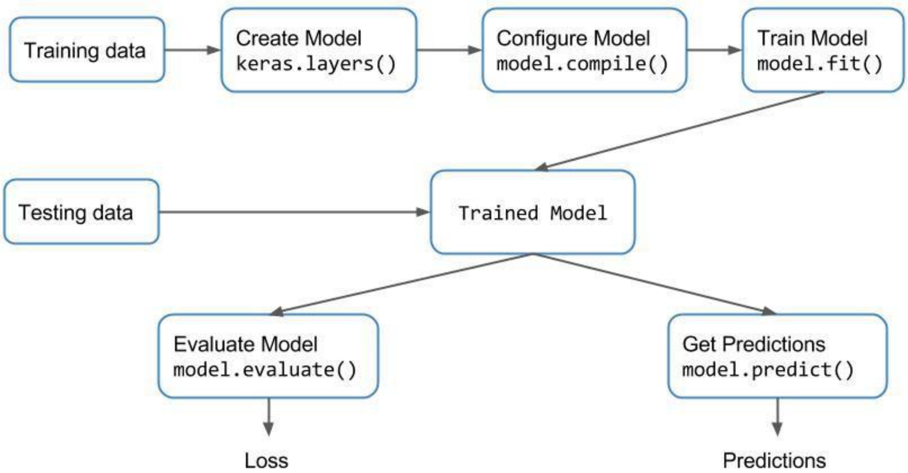

Figure6. The sequential block diagram of the multilayer sensor model was created using the Keras module.

We utilized a neural arrangement with 4 covered-up layers and a yield layer with 10 units (i.e. add up to a number of names). The number of units within the covered-up layers is kept to 512. The input to the organize is the 784-dimensional cluster changed over from the 28 x 28 picture. We utilized the Successive shows for building the arrangement. Within the Consecutive show, able to fair stack up layers by including the specified layers one by one. We utilized the Thick layer, moreover called a completely connected layer since we are building a feedforward arrangement in which all the neurons from one layer are associated with the neurons within the past layer. Separated from the Thick layer, we added the ReLU actuation work which is required to introduce non-linearity to the demonstration. This will assist the organize learn non-linear choice boundaries. The final layer could be a softmax layer because it could be a multiclass classification issue.

D. CONVOLUTIONAL NEURAL NETWORK

The implementation of neural network coding was done using Keras. It is an open-source neural network library for building and implementing deep learning models. We use the Sequential class in Keras, which allows us to create layer-by-layer models. The input image size is set to 28 (height), 28 (width), and 1 (number of channels). Then, we create a model where the first layer is the Conv layer. This technique uses a matrix to convolution the height and width of the input data and extract features from it. This matrix is called the filter or kernel. The values in the filter matrix are weighted. We use 32 filters with string 1 in each dimension (3,3). The convolution filter of the input data produces an initial image whose size is given by: ((N + 2P - F)/S) + 1, where N = length of the input image, P = padding, F = Filter Dimensions and S = stride. In this process, the depth (number of channels) of the output image is equal to the filters used. We use the Relu activation function to add nonlinearity. Then, another convolutional layer is used where we use 64 filters of the same length (3,3) as stride 1 and the Relu function. Then, a pooling layer is used for these layers, which reduces the remaining images and computations in the network. We use MAX pooling, which keeps only the highest value in the pool. The depth of the network layer remains unchanged. We keep the pool size (2,2) as step 2 so every 4 pixels will be one pixel. To avoid overfitting the model, a Dropout layer is used, which discards some randomly selected neurons to simplify the model. We set the probability of exiting the node to 0.25 or 25%. A flattening layer is then used, which involves flattening column matrices (vectors) from, for example, the two-dimensional matrix. This sentence vector will be entered into the attached process. This set has 128 neurons with a failure rate of 0.5% or 50%. After using the ReLu activation function, the output is fed to the output process, which is the final process of the model. This set has 10 neurons representing clusters (numbers 0 to 9) and uses the SoftMax function to perform classification. This function returns the probability distribution for all 10 groups. The category with the biggest results is urine. Figure 7.

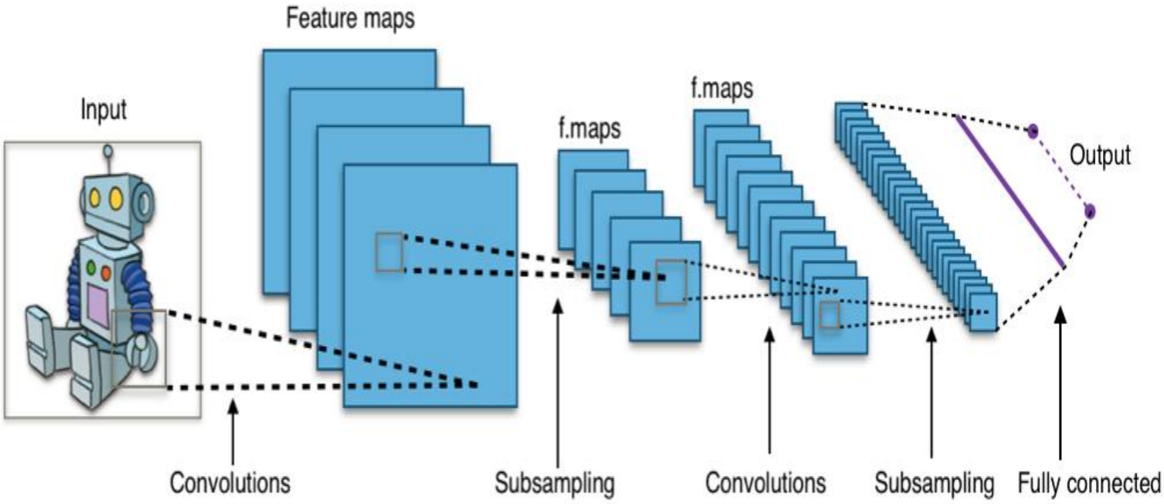

Figure7. Detailed architecture of convolutional neural networks and appropriate specification of each layer

V. RESULT

After using three algorithms, SVM, MLP, and CNN, we compare their accuracy and execution times with the help of images that we try to understand clearly. Above we determine the training and testing accuracy of each model. After all models were completed, we found that SVM had the highest accuracy on the training data, while CNN had the highest accuracy on the test data. We also compare execution time to better understand how the system works. Generally speaking, the running time of an algorithm depends on how it works. Therefore, we trained the deep learning model and SVM model up to 30 times according to specific instructions to obtain suitable results. SVM has the shortest execution time and CNN has the longest execution time.

Table 1. This table represents the performance of each model. There are 5 columns in the table; The 2nd column represents the model name, the 3rd and 4th columns represent the training and testing accuracy of the model, and the 5th column represents the running time of the model.

Sl. No	Model Name	Training rate	Testing Rate	Execution
1	SVM	99.98%	94.005%	1:35 min
2	MLP	99.92%	98.85%	2:32 min
3	CNN	99.53%	99.31%	44.02 min

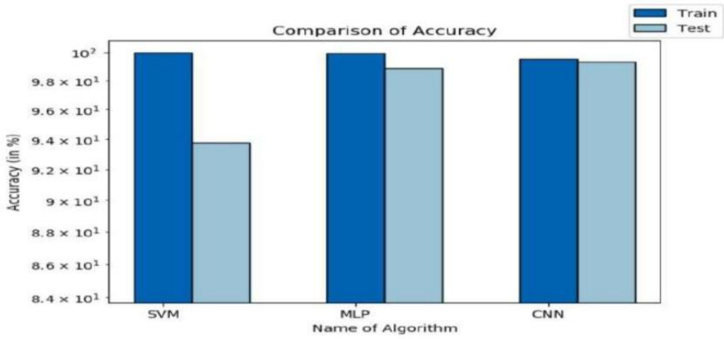

Figure8. Bar graph showing the accuracy comparison (SVM (Train: 99.98%, Test: 93.77%), MLP (Train: 99.92%, Test: 98.85%), CNN (Train: 99.53%, Test: 99.31%)).

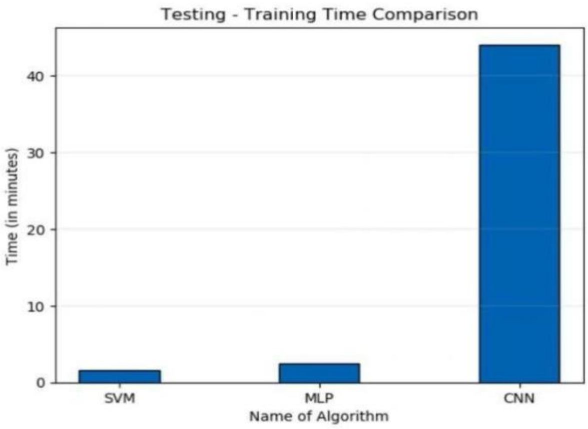

Figure9. The graph shows the comparison of SVM, MLP, and CNN execution times (SVM: 1.58 minutes, MLP: 2.53 minutes, CNN: 44.05 minutes).

Furthermore, we visualized the performance measures of deep learning models and how they ameliorated their accuracy and reduced the error rate concerning the number of epochs. The significance of sketching the graph is to know where we should apply an early stop so that we can avoid the problem of overfitting as after some epochs, the change in accuracy becomes constant.

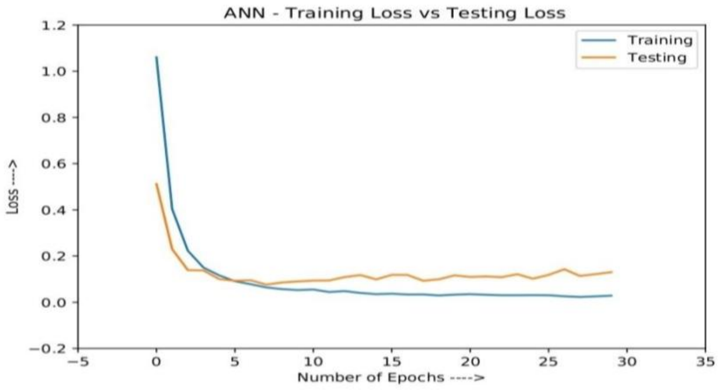

Figure10. This figure shows the learning loss as the number of epochs increases (loss rate and number of epochs) in a multilayer perceptron.

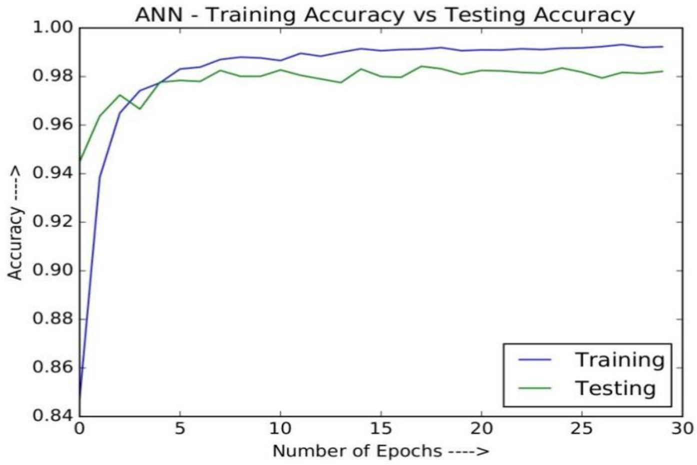

Figure11. This figure shows how training changes with increasing time (accuracy and number of epochs) on a multilayer perceptron.

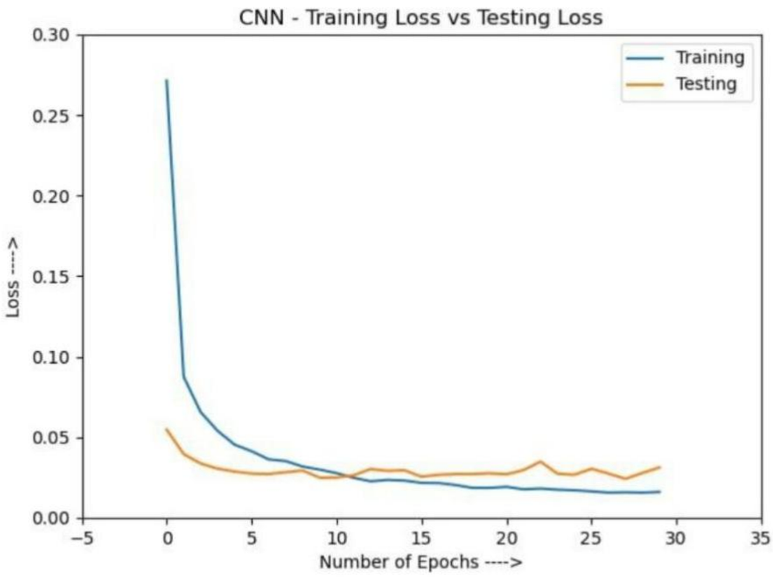

Figure12. The graph shows the change in CNN learning loss with an increasing number of epochs (loss rate and number of epochs).

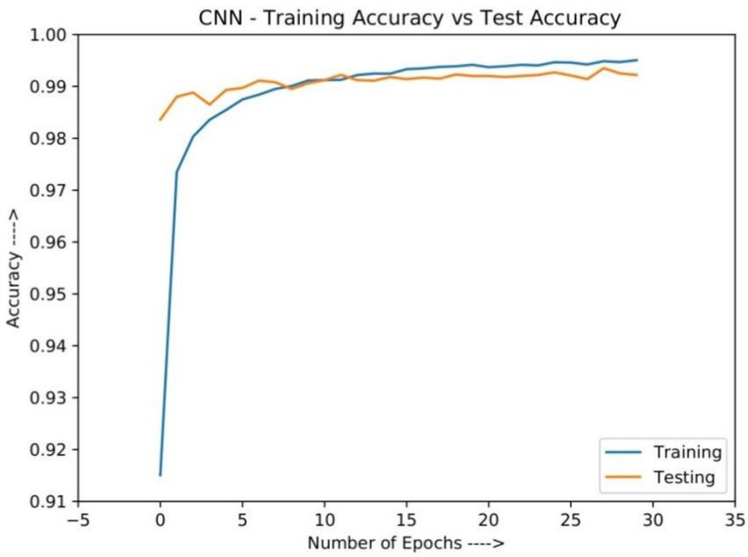

Figure13. This figure shows the CNN training accuracy as the number of epochs increases (accuracy vs. number of epochs).

VI. CONCLUSION

In this study, we implemented our coding model based on deep learning and machine learning algorithms using the MNIST dataset. We compare them according to their features to evaluate the most accurate model. Support vector machine is one of the simple techniques and hence it is faster than most algorithms and this can provide the highest training but due to its simplicity, it cannot perform complex and confusing image classification based on MLP and CNN algorithms. We found that CNN gave the most accurate results regarding coding. Therefore, this leads us to the conclusion that CNN is the best solution for all prediction problems, including image-based inputs. Then, by comparing the execution times of the algorithms, we decided that due to some model limitations, it was not useful to increase the number of epochs without changing the algorithm configuration, and at some point after that we found that the model started to grow. It overfits the dataset and gives us inaccurate predictions.

VII. FUTURE SCOPE

The future development of application-based machines and deep learning algorithms is unlimited. In the future, we may learn more dense or hybrid algorithms than existing algorithms and use more data to complete solutions to different problems. In the future, the applications of these algorithms will be determined by the difference between the above methods, from citizens to high-level officials, with its development in the future, we can know advanced applications that can be used for secret or government purposes. Our aim is that schools and ordinary people can use this system in hospital applications for detailed diagnosis, treatment, and care of patients, in monitoring systems to track suspicious activity in fingerprint and retina scanners, and in-database filtering applications, we use it on a National Volunteer basis. Forces Equipment Inspection and many other large and small group investigations. Advances in this field can help us create safe, informed, and efficient environments by using these algorithms in both routine and advanced applications (e.g., business or government). Applications based on artificial intelligence and deep learning are the future of the technological world because they are very realistic and useful in many big problems.

REFERENCES

[1] “Handwriting recognition”:https : //en.wikipedia.org/wiki/Handwritingrecognition
[2] “What can a digit recognizer be used for?”: https : //www.quora.com/W hat−can−a−digit−recognizer− be − used − f or
[3] ”Handwritten Digit Recognition using Machine Learning Algorithms”, S M Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana & Abdullah Al Jobair.
[4] ”Handwritten Digit Recognition Using Deep Learning”, Anuj Dutt and Aashi Dutt.
[5] ”Handwritten recognition using SVM, KNN, and Neural networks”, Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sharif.
[6] ”Recognition of Handwritten Digit using Convolutional Neural Network in Python with Tensorflow and Comparison of Performance for Various Hidden Layers”, Fathma Siddique, Shadman Sakib, Md. Abu Bakr Siddique.
[7] ”Advancements in Image Classification using Convolutional Neural Network” by Farhana Sultana, Abu Sufian & Paramartha Dutta.
[8] ”Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18 FFDG PET/CT images” Hongkai Wang, Zongwei Zhou, Yingci Li, Zhonghua Chen, Peiou Lu, Wenzhi Wang, Wanyu Liu, and Lijuan Yu.
[9] https : //github.com/rishikakushwah16/SV M digit recognition using MNIST dataset
[10] “Support Vector Machine Algorithm”: https : //www.javatpoint.com/machine−learning−support− vector − machine − algorithm