MLHandwrittenRecognition:Handwritten Digit Recognition using Machine Learning Algorithms

Farah Essam [a], Hanien Samy [a], Judy Wagdy [a], Steven Albert [a] , Diaa Salama AbeElminaamb [a,b,*]

[a] Department of Computer Science, Misr International University, Cairo, Egypt

[b] information systems department; Faculty of Computers and artificial intelligence; Benha university, Egypt

*Corresponding Author: Diaa Salama [***diaa.salama@miuegypt.edu.eg***]

ARTICLE DATA

Article history:
Received 16 Oct 2022
Revised 12 Jan 2023
Accepted 12 Jan 2023
Available online

Keywords:
OCR
Handwritten digit
Machine Learning
Computer Vision

ABSTRACT

Handwritten digit recognition has remained a topic of interest to computer vision scientists. Its origination precedes the emergence of the machine as it is a crucial component of the digital transformation of the majority of institutions in numerous fields. With the uprising of machine models, choosing a satisfactory and fit algorithm for this multi-class (0-9) classification problem became challenging. This paper aims to compare seven machine learning algorithms in terms of their performance metrics in recognizing handwritten digits employing two datasets. The - Nearest Neighbors (kNN), Support Vector Machine (SVM), Logistic Regression, Neural Network, Random Forest (RF), Naive Bayes, and Decision Tree models are accordingly evaluated concerning the Area Under the Curve (AUC), accuracy (ACC), F1-score (F1), precision (PREC), and recall (REC). The widely used Modified National Institute of Standards and Technology database (MNIST) dataset and the Handwritten Digit Classification dataset (HDC) have been the providers of the images on which this research is conducted. The results confirm that the Neural Networks model is a great classifier for this problem; however, it presents similar results to other machine learning classifiers in several cases. Therefore, this paper does not provide an absolute choice of a classifier for the handwritten digit recognition problem but rather explains the reason behind the performance of each model.

1. Introduction

Handwriting recognition, often known as handwriting Optical Character Recognition (OCR) or cursive OCR, is an OCR sub-field that converts handwritten characters into digital text or commands. Moreover, handwritten digit recognition is the computer's ability to recognize human handwritten digits from a variety of sources, such as images, papers, and touch screens, to classify them into ten numerical categories (0-9). The concept of handwritten digit recognition began with the use of pattern matching. Shelia Guberman, who lived in Moscow then, built the first applied pattern recognition program in 1962.

OCR is a worldwide used technology that allows people to view, find, and distinguish text in images and labels in a variety of ways. It is one of the early computer vision challenges, as it does not necessarily require deep learning in some aspects [1]. As a result, various OCR implementations existed even before the deep learning hype in 2012. Modern OCR software uses Artificial intelligence (AI) and machine learning to reach even higher degrees of precision, such as recognizing different languages, reading handwritten text and writing styles, dealing with typical data restrictions, and much more. It is fair to say that deep learning has radically changed the field of OCR. In the last few years, a lot of development has been made in this field; for instance, Convolutional Neural Network (CNN) is currently a powerful tool for solving computer vision problems in many fields [2].

Handwritten digit recognition is subject to several challenges, including:

- A user's handwriting style varies and is inconsistent from time to time.
- Cursive handwriting makes character separation and recognition complex.

- As opposed to printed text which is rotation-invariant, handwritten text may be rotated in various ways.
- Insufficient image quality for the source may result in poor recognition results.

This makes it difficult for various machine-learning techniques to detect the digits.

This paper examines and compares machine learning classifiers in recognizing handwritten digits. It also addresses the methodology employed to detect digits using the MINST [3] and Handwritten Digit Classification datasets. To evaluate them, performance metrics were utilized.

In section 2, similar studies are discussed, and relevant results are presented. In Section 3, the methodology followed in this study is explained. Section 4 discusses the techniques used and the output of each one after testing on all data sets. Section 5 evaluates the results presented in Section 4 [4]. Lastly, the conclusion is conducted in Section 6.

2. Related Work

In [5], the authors compare the performance (accuracy, complexity, execution time, and the number of epochs) of the kKNN, RF, SVM, and various forms of neural network classifiers in recognizing handwritten digits. They have based their work on the MNIST dataset while applying simple image preprocessing (scaling and filtering) techniques. However, throughout their research, they have added extra preprocessing stages for more satisfactory results. The results show an overall accuracy of 98-100% for the test program and 96-98% for the industrial images. The proposed algorithms achieved nearly identical accuracy (with a one percent difference). Despite that, the Convolutional Neural Network (CNN) stands out with the highest accuracy but with a significant computing time.

The approach proposed in [6] is based on the comparison and evaluation of various machine learning algorithms, such as Bayesian networks, Multi-layer perceptrons (MLP), Random forest, SVM, and other Convolutional Neural Network (CNN) models, to train and test a handwritten numbers (0-9) recognition model. The MLP classifiers either stopped at local minimums or overfit. They have experimentally proven that the more data was used in training, the longer it took, but the more accurate the model became. They have also concluded that the SVM presented the best separation margins between a pair of classes. The final results show that the LeNet 4, a CNN method, achieves the highest accuracy 99.3%, with the drawback of a long training time (five weeks) but with a recognition time of 0.05 ms.

The authors of [7] experiment with the CNN method using the Rectified Linear Units (ReLU) activation function on the MNIST dataset of handwritten digits. They also use the Deeplearning4j (DL4J) framework. They have achieved the highest results with two convolutional layers, where the first consists of 32 filters and a window size equal to 5×5, and the second layer is composed of 7x7 window 64 filters. Consequently, they offer a model with an accuracy of 99.21%. In this paper, it is claimed that the high performance of their model concerns both criteria: accuracy and time.

The strategy pursued by the authors in [8] is to compare several classical machine learning algorithms in building handwritten digit classification models. The following algorithms were chosen: Logistic Regression, SVM, Decision Tree, RF, and kNN. The MNIST dataset is employed in the training and testing process of the above algorithms. The algorithms are compared based on factors such as the learning and prediction construction speed and the recognition accuracy. Each algorithm underwent 100 iterations of training and testing phases. According to the performance analysis, the kNN, Decision Tree, Logistic Regression, and Random Forest were the fastest algorithms during the prediction process for the test dataset, with respectively a recognition time of 0.01, 0.03, and 0.36 seconds, respectively. The kNN algorithm was the least computational power intensive during training, as highlighted with a score of (17.43%) of CPU load and memory use of 209.8 MiB. The most accurate algorithms in this study are the SVM, kNN, and RF, with accuracy scores equal to, respectively, 97.93%, 97.21%, and 96.95%.

In this article [9], even though the artificial neural network (ANN) with feature extractors, such as CNN, is a common choice for image classification, the authors claim that the three boosting algorithms discussed performed well in correctly predicting classes, with all three algorithms achieving the accuracy of more than 93%. Algorithms such as Extreme Gradient Boost (XGBoost), AdaBoost, and Gradient Boosting were used on the MNIST dataset to recognize handwritten digits. Confusion matrix, recall, F1 score, and precision were used to compare the performance of the algorithms. It can be concluded from the confusion matrix that the AdaBoost algorithm predicted the classes correctly with an accuracy of 96.86%. With an accuracy of 94.59%, the gradient boosting algorithm came out on top, followed by XGBoost at 93.6%. It was proven that the three algorithms could predict the '1' digit most effectively. Throughout the three algorithms, AdaBoost performed the best in terms of precision at 96.84%, F1 score, and recall at 96.85%.

In this paper [10], a framework for handwritten digit recognition is developed. It is based on feature extraction and algebraic fusion of various classifiers. The feature extraction is based on a CNN model using the MNIST dataset. The results demonstrate that the model's fusion achieves at least 98% accuracy. The authors indicate that the same classifier can perform differently depending on the test set. This remark is due to the various ways of shaping a single digit based on the user's handwriting style. The ensemble learning techniques construct the solution provided by the authors to improve and stabilize the performance of the classifiers. For training base classifiers and primary ensembles, the kNN and RF have been used. The findings show that the KNN method's instance-based learning resulted in over 95.8% accuracy on the two feature sets. While the RF approach may create highly diverse decision tree classifiers using a range of training samples and feature subsets, the accuracy on the two feature sets is above 95.7%. The results also reveal that by utilizing the MNIST data set, their suggested ensemble technique may reach a classification accuracy of more than 98%, indicating that employing ensemble learning to train various classifiers is particularly advantageous to improving overall classification performance.

Several machine learning techniques, such as unsupervised learning algorithms, are investigated in [11]. K-means clustering is used to minimize the size of the data. Neural networks, SVMs, and closest-neighbor methods then follow a linear classifier. The author used the MNIST dataset to compare performance using prediction accuracy. Handwritten digit recognition performance is determined by two factors: (1) the method used to extract features and (2) the algorithm used for classification. Users should choose configuration parameters to improve performance and some computational constraints associated with implementing the algorithms. SVM is a binary classifier that determines the optimal separating hyperplane between two classes and can be utilized to split the feature space into classification regions for multi-class problems. Even with established algorithms, further study is needed to discover the best tuning parameters. Neural networks can be investigated using a variety of hidden layers and nodes. The softmax function is used in the neural network's output layer. Other output functions for those layers that could have an important impact on the performance may be investigated.

In [12], the author uses Convolutional neural networks (CNN) as an example of image classification and, more precisely, Keras sequential models as a classifier. The dataset used for training and testing is the MNIST dataset. The most important phase was picture preprocessing, which was accomplished with OpenCV and Scipy. This study's classifier is a sequential model with a four-layer CNN. Different CNN layers, such as two, three, and four-layered CNN, were tested for accuracy on the test data. The four-layered CNN exhibited the best accuracy of 99.25%, with the least loss in the case of 15 epochs among all the layers evaluated. Comparative studies have been conducted over five epochs. 4-layer CNN had the highest accuracy 98.89%, but the accuracy was lower than the one in the 15 epochs. This is because the algorithm can train itself more efficiently when 15 epochs are employed, resulting in improved accuracy and smaller losses. Also, API can show the outcome for every digit the user enters. This allows the online mechanism to recognize numbers with high accuracy accurately. Users can insert a number into the terminal window by hand, and the recognized digit will be displayed in the next window. The entire process may be enhanced by using other methods, such as a multi-layer perceptron (MLP).

In [13], the paper compares three classification algorithms using the National Institute of Standards & Technology (NIST) handwritten dataset: K Star, MLP, and Naive Bayes (NB) algorithms based on correlation features selection (CFS). This comparison aims to determine which of the three classifiers can provide a suitable accuracy rate while using the fewest number of features possible. Precision, recall, and F-measure is the accuracy measurement measures used to evaluate each classifier's performance independently. The results reveal that the K star algorithm performs better than NB and MLP algorithms, with an accuracy of (82.36%). A comparison of three classification techniques was performed to recognize NIST handwritten digits. To get a suitable recognition rate, only 37 features out of 256 were picked using (CFS). After evaluating each classifier on 46080 samples using 10-fold cross-validation, K Star performed better than the other classifiers with an accuracy of 82.36%, compared to NB 67.04% and MLP 78.35%. In this comparative approach, the K Star algorithm outperforms the two other classifiers (NB and MLP) in handwritten digit recognition.

In [14], the authors tested five classical machine learning classifiers and compared them using performance metrics such as recall and F1-Score. They used only 1 dataset: the MNIST dataset. The paper discusses the preprocessing steps the data went through before training, including line localization and thresholding techniques. Feature extraction then took place, and the data was ready for training. The five classifiers were SVM, Decision Tree, RF, ANN, kNN, and K-Means Algorithm. The accuracies achieved were 90%, 87%, 97%, 98%, and 98%. The paper tested four different kernels for the SVM, and the best accuracy was achieved with a polynomial kernel of 90%. However, ANN achieved the highest precision, 0.97. The SVM took the longest time in training, while KNN took the longest time in testing.

The authors in [15] propose different feature extraction methods to be applied to the MNIST dataset before considering which classifiers to use. They tried four feature extraction methods: Cavity, Hu moments, Zernike Moments, and Hog transformed and combined them with different classifiers. The machine learning techniques were KNN, SVM one versus one, SVM one versus all, Decision Tree, and MLP. The authors tested 17 combinations, and the best results were by training a KNN model on HOG Features, resulting in a recognition rate of 96.57%. However, they faced a limitation due to the size of the feature vector (1296) generated by the HOG extraction method. The paper couldn't showcase the performance of MLP, SVOMO, and SMOVA models with HOG Features. Also, other related work in [16-25] has been proposed in recent years to address machine learning and its application in different fields.

3. Proposed Approach

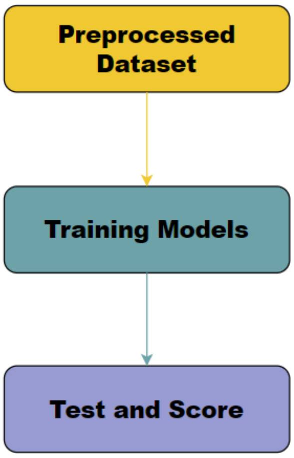

FIGURE 1. Methodology Diagram

3.1. Datasets

1) *MNIST (Modified National Institute of Standards and Technology database)[3]* is a common dataset of handwritten digits that carries 60,000 handwritten digits for training and 10,000 handwritten digits for a machine learning model to be tested and verified that they work. A sample of this dataset is shown in figure 2. This is a subset of the larger set available from NIST. NIST's black-and-white images were also normalized in size, centered and smoothed to fit within a 28 x 28-pixel border box, and introduced with gray shading. The class distribution is illustrated in figure 3.

FIGURE 2. MNIST Dataset

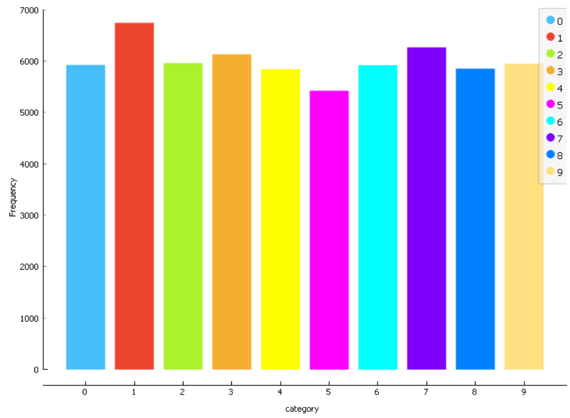

FIGURE 3. MNIST Class Distribution

2) *MHDC (Handwritten Digit Classification dataset) [26]* is a dataset of 30,010 binary images of handwritten digits, as shown in the representative piece below (figure 4). Each digit (0-9) is assigned a folder that holds 3001 enhanced and binarized images of that digit. All of the images are 128 x 128 pixels.

FIGURE 4. HDC Dataset

3.2. Training

The datasets described above are used to train the chosen classifiers. The chosen classifiers are the following:

1) *KNN is a simple and straightforward algorithm that maintains all existing cases and classifies new ones based on a majority vote of its k neighbors [27]. KNN (k- Nearest Neighbors) applies to both classification and regression problems. However, in practice, it is more commonly used in classification problems.*

2) *SVM (Support Vector Machine)* is a supervised machine learning model. Each data point is plotted in n-dimensional space (n being the number of features), with every feature being the value of a specific coordinate. The kernel trick allows the performance of nonlinear classification [28].

3) *Logistic Regression* is a classification algorithm. It is generally used to calculate discrete values depending on variables [29]. In other words, it fits data to a logistic function to forecast the likelihood of an event occurring. Hence, it's also called logistic Regression as it predicts probabilities. Its output values range from 0 to 1.

4) *Neural Network:* In deep learning techniques, artificial neural networks (ANNs) and simulated neural networks (SNNs) are two forms of neural networks. They get their name and structure from the human brain and work similarly to how neurons communicate. Neural networks can support computers in making accurate decisions while requiring little human involvement [30]. This depends on its ability to learn and model nonlinear and complex relationships between input and output data.

5) *Random Forest:* is a machine learning ensemble technique used in classification and Regression. Its algorithm is responsible for determining the output based on the assumptions of the decision trees [31]. It makes predictions by averaging the results of several trees. As the number of trees grows, so does the precision of the outcome. Decision nodes, root nodes, and leaf nodes are the three parts of a decision tree. A decision tree approach divides a dataset into branches, which are then divided further into branches. This procedure is repeated until a leaf node is reached and cannot isolate the leaf node anymore. To forecast the outcome, the qualities represented by the nodes in the decision tree are utilized.

6) *Naive Bayes* is a probabilistic classifier that uses Bayes' theorem and assumes features independence. According to a Naive Bayes classifier, the presence of one feature in a class is unaffected by the presence of any other feature. It is simple to implement and performs well when a class's probability depends on casual factors' probability [32].

7) *Decision Tree:* is a supervised learning algorithm commonly used for classification issues. It works with both continuous and categorical dependent variables. The population is divided into two or more homogeneous groups using this algorithm. To make as many unique groups as possible, this is done using the most significant attributes and independent variables [33].

3.3. Test and Score

The proposed classifiers are compared against each other in each dataset using various performance metrics.

4. Experimental Results

In this section, the Orange data mining platform is used to evaluate the mentioned algorithms' performance in recognizing handwritten digits with the MNIST and HDC datasets.

4.1. Performance Metrics

The proposed algorithms are evaluated in terms of Area Under the Curve (AUC), accuracy (ACC), F1-score (F1), precision (PREC), and recall (REC). Four variables contribute to calculating most of these metrics: true positive (TP), which represents the number of instances that have been correctly classified to the selected class; false positive (FP), true negative (TN), and false negative (FN). The calculation formulas are presented in table 1.

TABLE 1: Evaluation measures

Measures	Formulas
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F-measure	$2 \times \frac{Precision \times Recall}{Precision + Recall}$

4.2. Algorithms parameters

The recognition of digits (0-9) is a multi-class classification problem; therefore, the algorithm's parameter tuning was adapted accordingly. Orange data mining tool provides several customizations to each model before training and testing it.

In the kNN model, it was set to 5 nearest neighbors with the Euclidean metric (distance parameter) and uniform weights for all points in the same neighborhood.

The SVM's kernel function of choice was the Radial Basis Function (RBF). The SVM Linear is the traditional SVM with a linear kernel function.

The Logistic Regression model used a Tikhonov regularization (L2) type with a cost strength (C) equal to 1.

The Neural Network model was accorded the Rectified Linear Unit (ReLu) activation function for the hidden layer along with the Adam stochastic gradient-based optimizer and a maximal number of iterations equal to 100 iterations. The hidden layers were also adjusted to 100 neurons per hidden layer.

The Random Forest model was calibrated to include ten decision trees in the forest. The growth control attributes were regulated so that the smallest subset that can be split has to have five instances.

No customization was needed for the Naive Bayes model.

Regarding the tree model, the limit of the maximal tree depth was set to 100 node levels, nodes with less than five instances cannot be split, and splits would never produce branches with less than two training examples.

Most of Orange's models apply a few default preprocessing techniques, such as one-hot encoding for categorical variables, removing empty columns, or substituting missing values with mean values. They also eliminate instances with missing target values.

4.3. MNIST dataset results

As shown in table 2, the Neural Network model outperformed the other experimented models in all the chosen performance metrics with an AUC of 1.0, ACC of 0.978, an F1-score of 0.978, and a recall of 0.978. This experiment suggests that the Neural Network offers more satisfactory results than Trees, SVM, kNN, and Logistic regression models in recognizing handwritten digits, as illustrated by the confusion matrix in figure 5.

TABLE 2: Performance measures using the mentioned algorithms on the MNIST dataset

Algorithms	AUC	ACC	F1	PREC	REC
kNN	0.988	0.928	0.928	0.929	0.928
SVM w/RBF Kernel	0.997	0.924	0.924	0.928	0.924
SVM w/Linear Kernel	0.995	0.906	0.905	0.907	0.906
Logistic Regression	0.999	0.976	0.976	0.976	0.976
Neural Network	1.000	0.978	0.978	0.978	0.978
Random Forest	0.978	0.852	0.852	0.853	0.852
Naïve Bayes		0.767	0.769	0.774	0.767
Tree	0.832	0.748	0.747	0.747	0.748

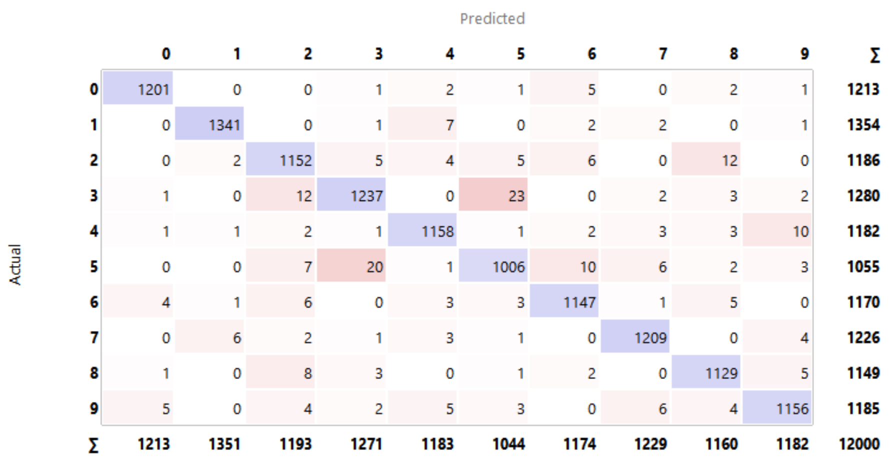

FIGURE 5. NN Confusion Matrix on MNIST Dataset

4.4. HDC dataset results

Table 3 demonstrates the result of applying the studied algorithms to the HDC dataset. The metrics values of the SVM, Logistic Regression, and Neural Network models are similar and represent the highest values among the other algorithms as they have values not less than 0.954 that reach 0.999. The Tree model, however, presents significantly low-performance measures compared to the rest of the models.

TABLE 3: Performance measures using the mentioned algorithms on the HDC dataset

Algorithms	AUC	ACC	F1	PREC	REC
kNN	0.985	0.913	0.912	0.914	0.913
SVM w/RBF Kernel	0.998	0.954	0.954	0.954	0.954
SVM w/Linear Kernel	0.997	0.935	0.935	0.936	0.935
Logistic Regression	0.999	0.973	0.973	0.973	0.973
Neural Network	0.999	0.974	0.974	0.974	0.978
Random Forest	0.977	0.850	0.850	0.850	0.850
Naïve Bayes		0.789	0.789	0.790	0.789
Tree	0.837	0.730	0.730	0.730	0.730

5. Discussion

All classifiers performed comparably well, as shown in Table 2. However, both ANN and SVM and Logistic Regression performed exceptionally well. The first two models mentioned convert data into higher dimensional spaces, which is the responsibility of the kernel in SVM, and the hidden layers in neural networks. This allows cleaner separation between classes. However, increasing the dimensional space of the dataset also increases the risk of overfitting the model, which can be noticed slightly in the recall score of the SVM model, which used an RBF Kernel. This is where algorithms such as Logistic Regression may outperform others. Logistic Regression does not change the dimensional spaces of the data but rather by predicting the likelihood of classes using the sigmoid function based on some input parameters. Other models, such as the Decision Tree and Naive Bayes, performed poorly compared to the abovementioned models. kNN is a rather simple model; it does not require training, nor does it have a loss function that requires minimization by training. When making a prediction, it searches the data and computes to find the nearest k data points during runtime. This makes kNN vulnerable to outliers and noise in datasets. Both kNN and decision trees are non-parametric. Decision trees have a high chance of growing complexity and overfitting when applied to large datasets, especially image datasets. Both datasets were used to indicate very similar conclusions. Overall, the difference in performance between the top classifiers was fairly small to make definitive conclusions about the preferred model.

6. Conclusion

The field of OCR is very broad; even though significant work has been accomplished throughout the years, there still is room for more to be done. The main purpose of this paper was to showcase the performance of various machine learning models in handwritten digit recognition. The models, in comparison, were trained on two fairly medium-sized datasets, 1 of them being the popular MNIST dataset. After thoroughly analyzing the performance metrics, it was proven that neural networks tend to outperform classical Machine Learning algorithms. However, the performance margin between them is insufficient to conclude that classical machine learning algorithms are not ideal.

References

[1] B. V. Dasarathy, "Nearest neighbor (NN) norms: NN pattern classification techniques," *IEEE Computer Society Tutorial,* 1991.

[2] E. Hancer, B. Xue, and M. Zhang, "Differential evolution for filter feature selection based on information theory and feature ranking," *Knowledge-Based Systems,* vol. 140, pp. 103-119, 2018.

[3] Y. LeCun, "The MNIST database of handwritten digits," *http://yann.lecun.com/exdb/mnist/ ,* 1998.

[4] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence,* vol. 97, no. 1-2, pp. 273-324, 1997.

[5] Y. Chychkarov, A. Serhiienko, I. Syrmamiikh, and A. Kargin, "Handwritten Digits Recognition Using SVM, KNN, RF and Deep Learning Neural Networks," in *CMIS*, 2021, pp. 496-509.

[6] O. M. Khanday and S. Dadvandipour, "Analysis of machine learning algorithms for character recognition: a case study on handwritten digit recognition," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 21, no. 1, pp. 574-581, 2021.

[7] S. Ali, Z. Shaukat, M. Azeem, Z. Sakhawat, and T. Mahmood, "An efficient and improved scheme for handwritten digit recognition based on convolutional neural network," *SN Applied Sciences,* vol. 1, no. 9, pp. 1-9, 2019.

[8] O. Voloshchenko and M. Plechawska-Wójcik, "Comparison of classical machine learning algorithms in the task of handwritten digits classification," *Journal of Computer Sciences Institute,* vol. 21, 2021.

[9] S. Ghosh, "Comparative Analysis of Boosting Algorithms Over MNIST Handwritten Digit Dataset," in *Evolutionary Computing and Mobile Sustainable Networks*: Springer, 2022, pp. 985-995.

[10] H.-h. Zhao and H. Liu, "Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition," *Granular Computing,* vol. 5, no. 3, pp. 411-418, 2020.

[11] A. O. Otiko, J. A. Odey, and G. A. Inyang, "HANDWRITTEN DIGIT RECOGNITION: A PERFORMANCE STUDY OF MACHINE LEARNING TOOLS."

[12] K. Senthil Kumar, S. Kumar, and A. Tiwari, "Realtime Handwritten Digit Recognition Using Keras Sequential Model and Pygame," in *Proceedings of 6th International Conference on Recent Trends in Computing*, 2021: Springer, pp. 251-260.

[13] M. B. Abdulrazzaq and J. N. Saeed, "A comparison of three classification algorithms for handwritten digit recognition," in *2019 International Conference on Advanced Science and Engineering (ICOASE)*, 2019: IEEE, pp. 58-63.

[14] R. KARAKAYA and S. KAZAN, "Handwritten digit recognition using machine learning," *Sakarya University Journal of Science,* vol. 25, no. 1, pp. 65-71, 2021.

[15] K. Derdour, H. Mouss, and R. Bensaadi, "Multiple Features Extraction and Classifiers Combination Based Handwriting Digit Recognition," *International Journal on Electrical Engineering and Informatics,* vol. 13, no. 1, pp. 163-178, 2021.

[16] Abd Elminaam, D.S., El Tanany, A., Salam, M.A. and Abd El Fattah, M., 2022, May. CPSMP_ML: Closing price Prediction of Stock Market using Machine Learning Models. In 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 251-255). IEEE.

[17] Ai, M.A., Shanmugam, A., Muthusamy, S., Viswanathan, C., Panchal, H., Krishnamoorthy, M., Elminaam, D.S.A. and Orban, R., 2022. Real-time facemask detection for preventing COVID-19 spread using transfer learning based deep neural network. Electronics, 11(14), p.2250.

[18] Neggaz, N. and AbdElminaam, D.S., 2021, May. Automatic sport video mining using a novel fusion of handcrafted descriptors. In 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 387-394). IEEE.

[19] Salam, M.A., Ibrahim, L. and Abdelminaam, D.S., 2021. Earthquake Prediction using Hybrid Machine Learning Techniques. International Journal of Advanced Computer Science and Applications, 12(5), pp.654-6652021.

[20] Mahmoud, E., Kader, H.A. and Minaam, D.A., 2019, October. Fuzzy knowledge base system for floating car data on SUMO. In 2019 29th International Conference on Computer Theory and Applications (ICCTA) (pp. 38-42). IEEE.

[21] AbdElminaam, D.S., ElMasry, N., Talaat, Y., Adel, M., Hisham, A., Atef, K., Mohamed, A. and Akram, M., 2021, May. HR-chat bot: Designing and building effective interview chat-bots for fake CV detection. In 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 403-408). IEEE.

[22] AbdElminaam, D.S., Fahmy, A.G., Ali, Y.M., El-Din, O.A.D. and Heidar, M., 2022, May. DeepECG: Building an Efficient Framework for Automatic Arrhythmia classification model. In 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 203-209). IEEE.

[23] AbdElminaam, D.S., Fahmy, A.G., Ali, Y.M., El-Din, O.A.D., Aly, A.R. and Heidar, M., 2022, May. ESEEG: An Efficient Epileptic Seizure Detection using EEG signals based on Machine Learning Algorithms. In 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 210-215). IEEE.

[24] AbdElminaam, D.S., Ahmed, N., Yasser, M., Ahmed, R., George, P. and Sahhar, M., 2022, May. DeepCorrect: Building an Efficient Framework for Auto Correction for Subjective Questions Using GRU_LSTM Deep Learning. In 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) (pp. 33-40). IEEE.

[25] Ali, M.A., Orban, R., Rajammal Ramasamy, R., Muthusamy, S., Subramani, S., Sekar, K., Rajeena PP, F., Gomaa, I.A.E., Abulaigh, L. and Elminaam, D.S.A., 2022. A Novel Method for Survival Prediction of Hepatocellular Carcinoma Using Feature-Selection Techniques. Applied Sciences, 12(13), p.6427.
[26] Pradheep. *Handwritten Digit Classification dataset*.
[27] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine learning: algorithms and applications*. Crc Press, 2016.
[28] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet],* vol. 9, pp. 381-386, 2020.
[29] T. G. Nick and K. M. Campbell, "Logistic regression," *Topics in biostatistics,* pp. 273-301, 2007.
[30] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. P. Campbell, "Introduction to machine learning, neural networks, and deep learning," *Translational Vision Science & Technology,* vol. 9, no. 2, pp. 14-14, 2020.
[31] Y. Liu, Y. Wang, and J. Zhang, "New machine learning algorithm: Random forest," in *International Conference on Information Computing and Applications*, 2012: Springer, pp. 246-252.
[32] G. Bonaccorso, *Machine learning algorithms*. Packt Publishing Ltd, 2017.
[33] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b,* vol. 4, pp. 51-62, 2017.