Computer Science and Interdisciplinary Research Journal
Volume 2, Issue 1, 2025. ISSN: 3033-1218, DOI: 10.70862/CSIR.2025.0201-12

Analyzing the Efficacy of Base Algorithms: Application of Machine Learning Models in Classification Tasks

Yass Khudheir Salal

General Directorate for Education in Al-Qadisiyah, Ministry of Education, Iraq

yasskhudheirsalal@gmail.com

Abstract – The article presents a comprehensive examination and comparison of several machine learning algorithms, such as Artificial Neural Network, K-Nearest Neighbor, Support Vector Machines, Decision Tree, and Random Forests, with a specific focus on the significance of neural networks in classification tasks. This study employs three distinct datasets: (1) the "Titanic" dataset is about the sinking of the British transatlantic liner, which resulted in the deaths of 1502 of the 2224 people who were on board. On April 15, 1912, the event occurred in the North Atlantic Ocean; (2) the "MNIST" dataset, which consists of 70,000 grayscale images (28×28 pixels) of handwritten digits; (3) and the "Fashion-MNIST" dataset, which showcases images of 10 fashion item classes such as T-shirt/Top, Trouser, Pullover, Dress, Coat, Sandals, Shirt, Sneaker, Bag, Ankle boots which have been classified. The analysis evaluates the strengths and weaknesses of each algorithm based on the research outcomes. The results suggest that for certain classification tasks, the use of machine learning algorithms may be more appropriate, as the accuracy levels may remain relatively unchanged, and vary depending on the size of the dataset used with the features it includes in the classification tasks.

Keywords – machine learning, deep learning, artificial neural networks, classification, Titanic.

DOI: https://doi.org/10.70862/CSIR.2025.0201-12

Corresponding author: Yass Salal,
General Directorate for Education in Al-Qadisiyah, Ministry of Education, Iraq
Email: yasskhudheirsalal@gmail.com

Received: 14 May 2025.
Revised: 3 June 2025.
Accepted: 11 June 2025.
Published: 22 June 2025.

© 2025 Yass Salal. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

The article is published with Open Access at http://csirjournal.com/

1. Introduction

1.1. Objective of the Study

Machine learning and deep learning represent fields within artificial intelligence that apply distinct methods and algorithms to resolve problems associated with classification, regression, image recognition, and the automation of various processes [1-3].

The problem of choosing the best algorithm surfaced during the examination of the Titanic dataset [4], which entails estimating a person's chances of surviving based on particular traits. The objective is to evaluate which algorithms are most appropriate for handling categorization tasks. The objectives of this study are to train neural networks and machine learning algorithms on a variety of classification problems, compare and analyze performance metrics such as Accuracy (ACC), training time, prediction delay, and model memory, and assess whether applying methods to particular issues is feasible.

In this study, the selected algorithms and techniques include the Support Vector Machine (SVM) [5], a machine learning strategy that establishes a separating hyperplane in the parameter space. The K-nearest neighbor (KNN) [6] method classifies an object by examining the classes of its K-nearest neighbor. A Decision Tree (DT) algorithm is utilized to create decision paths based on the evaluation of various features [7]. Random Forest (RF) [8] is another machine learning approach that merges the results from multiple decision trees. Furthermore, the backpropagation method is applied to train the standard neural network model referred to as an artificial neural network (ANN).

In the present investigation, the primary emphasis is placed on the importance of neural networks in

classification endeavors in relation to alternative algorithms, utilizing a range of datasets. This implies that for particular classification tasks, using accuracy as a metric for each algorithm is used to identify the most effective classification approach, as shown below.

1.2. Literature Review

In recent decades, a variety of classification methods have been introduced [9]. Among the most frequently used techniques are KNN [10], ANNs [11-13], SVM [14,15], and ensembles of classification trees, such as Random Forest [16,17]. Algorithms that utilize Decision Trees are relatively easy to implement, as they require the estimation of fewer parameters; thus, they demonstrate a significant level of automation. However, this practical benefit of DT over ANN can be diminished by a tendency to overfit the data [18]. For these reasons, DT models and, to a lesser extent, ANN have, in some applications, been supplemented or replaced by more advanced and easier-to-train machine learning algorithms (MLAs). Over the last decade, kernel methods like SVM [19,20] and tree ensembles such as RF [21] have emerged as highly effective methodologies for classification tasks. Numerous studies suggest that MLAs often outperform statistical techniques like discriminant analysis or logistic regression, especially when the feature space is complex or when the input datasets are expected to display varying statistical distributions [22]. As computational power has improved, MLAs have garnered more attention, and the performance of pattern recognition systems has correspondingly enhanced [23]. In many studies, RF, SVM, and gradient boosting methods are considered among the top-performing classifiers, while KNN remains a simple and interpretable baseline [24]. The essential steps in deciding which algorithm to use will depend on several factors, including the number of examples in the training set, the dimensions of the feature space, the existence of correlated features, and the risk of overfitting [25]. Once these considerations have been addressed, the suitable algorithm can then be chosen. By applying methods from statistical physics, the generalization performance of SVM, which has historically been considered a robust alternative to ANN for certain classification tasks, was investigated [26].

1.3. Data Preprocessing

Data preprocessing occurs before the training of models. For each classification task, both a training set and a test set are created. The dataset containing various attributes of a selected group of Titanic passengers is referred to as the "Titanic" dataset. The dataset consisting of a collection of handwritten digit samples is known as the "MNIST" dataset [27]. Additionally, the dataset comprising images of clothing items is called the "Fashion-MNIST" dataset [28]. A fundamental principle underpins the processing of all algorithms: to achieve optimal results, hyperparameters must be determined through trial and error following the training of each machine learning algorithm and neural network on the training set. The subsequent step involves evaluating the models' accuracy using the test set after the training phase.

1.4. Evaluating Algorithm Performance

For the algorithms being studied, the prediction accuracy was selected as a crucial criterion. The Kaggle platform was used to gauge the algorithms' accuracy on the Titanic dataset classification task. After submission, the predicted outcomes were evaluated against the ground-truth labels stored on the platform. Test accuracy was also used to evaluate performance on the MNIST and Fashion-MNIST datasets. The accuracy metric is calculated using the following formula:

$$\text{Accuracy} = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \mathbf{1}(\hat{y}_i = y_i). \quad (1)$$

In multi-label classification, this function returns the subset accuracy [29]. If the entire set of predicted labels for a sample exactly matches the true set of labels, then the subset accuracy is 1.0; otherwise, the value is less than one. If the predicted value is the $\hat{y}_i$ of the i sample and y_i is the corresponding true value, then the proportion of correct predictions out of $n_{samples}$ is determined by the above formula.

Accuracy measures the proportion of correctly predicted instances (both positive and negative in binary classification) over the total number of instances.

In addition to “accuracy”, the following parameters are used for comparison: the training time, the prediction time of the result, and the storage space occupied by the model on the storage device. The tasks for each dataset and a comparison of the effectiveness of the algorithms will be described below.

2. Data Exploration

The study examines three specific tasks: "Titanic", "MNIST", and "FashionMNIST", which will be explained as follows:

2.1. Titanic Dataset

For the Titanic task, the objective is to predict the survival outcome of each passenger based on selected features. In processing the data, four representative features selected for the model are listed in Table 1.

Table 1. Key features of the Titanic dataset.

Variables	Definition	Key
Passenger Class	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
Gender	Female, Male	
Sibsp	Siblings / spouses aboard the Titanic	Sibling = brother, sister, stepbrother, stepsister Spouse = husband, wife (excluding non-marital partners and unofficial relationships)
Parch	The number of parents and children (including foster children) aboard.	Parent = mother, father Child = daughter, son, stepdaughter, stepson Some children travelled only with a nanny, therefore parch=0 for them.

The target variable for this analysis is survival. The training dataset comprises 891 instances, while the testing dataset includes 418 instances. In the neural network model, the Adam optimizer is employed to enhance the learning process [30], alongside the "binary_crossentropy" loss function and the accuracy metric for performance evaluation. The architecture consists of four neurons in the input layer, 18 neurons in the hidden layer utilizing the Relu activation function, and a single neuron in the output layer with a Sigmoid activation function. The model is trained over 40 epochs.

2.2. The MNIST Dataset

The MNIST dataset contains labeled images of handwritten digits in the training data and unlabeled images of handwritten digits in the testing data. The testing data is only used when submitting to the competition and these images are unlabeled because if the testing data labels were provided, then it would be possible to cheat and to get a perfect score in the competition.

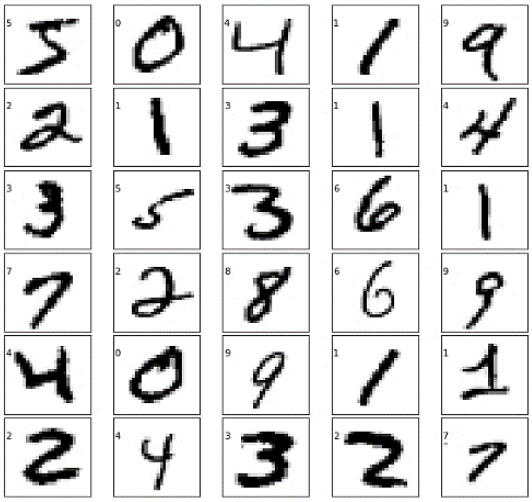


Figure 1. MNIST dataset examples.

In order to evaluate classifiers, scikit-learn's cross_val_score or train_test_split functions are used to set a portion of labeled training data for model evaluation.

The MNIST dataset comprises a total of 70,000 images, each measuring 28×28 pixels, representing handwritten digits from 0 to 9. This dataset is divided into a training set containing 60,000 images and a test set with 10,000 images. The primary features for analysis are the individual pixels of these images as shown in Figure 1.

2.3. Fashion-MNIST Dataset

The objective is to identify the type of garment depicted in a 28×28-pixel image. I will be using the Fashion-MNIST dataset, which is a collection of 70,000 images (28×28 pixels each). The collection contains ten types of clothing to be classified. The training sample contains 60,000 images, and the test sample contains 10,000 images. The input features are the pixel values of the images (see Figure 2).



Figure 2. Fashion-MNIST dataset examples.

3. Result and Discussion

3.1. Titanic Classification

The default hyperparameters were utilized for the Decision Tree model. The tasks analyzed in this study include "Titanic", "MNIST", and "FashionMNIST".

In the Random Forest model, the number of trees utilized (n_estimators) is set to 50, the k-neighbors parameter is configured to k=4. In the SVM model, all hyperparameters were applied with their default settings. The predicted outcomes were submitted to the Kaggle platform for the Titanic machine learning competition to assess prediction accuracy on the test dataset. The results were documented by evaluating the developed model on the test dataset, employing the "accuracy" metric as shown in Table 2.

Table 2. Results of classification accuracy on the Titanic dataset.

Metric / Classifier	ACC	Training time/sec.	Time to predict test sample /sec.	Storage space/KB
ANN	0.772	15.8800	0.2640	100.00
KNN	0.777	0.0029	0.0400	72.80
SVM	0.772	0.0030	0.0100	22.30
DT	0.764	0.0020	0.0003	9.55
RF	0.774	0.1920	0.0270	367.00

3.2. MNIST Classification

The following configurations were used to adapt the neural network: the "Adam" optimizer was utilized alongside the "categorical_crossentropy" loss function. The network architecture included 784 input neurons, a first hidden layer with 128 neurons activated by the "relu" function, a second hidden layer also consisting of 128 neurons with the "relu" activation, and an output layer featuring ten neurons with the "softmax" activation function. The training process spanned 30 epochs.

For the KNN algorithm, the number of neighbors (k) was set to five. In the case of the SVM and Decision Tree models, all hyperparameters were maintained at their default settings. The Random Forest model was constructed using 130 trees.

The performance of the models was evaluated by testing them on the dataset's test sample, with results measured using the "accuracy" metric, as shown in Table 3.

Table 3. Results of classification accuracy on the MNIST dataset.

Metric / Classifier	ACC	Training time/sec.	Time to predict test sample /sec.	Storage space/KB
ANN	0.897	204.015	0.71700	1.5300
KNN	0.793	0.08970	47.4330	373.00
SVM	0.769	302.024	106.432	84.600
DT	0.867	21.6012	0.18200	1.0030
RF	0.878	59.7898	0.61200	158.00

The best prediction accuracy was achieved using a neural network; however, this approach required more time for training compared to other algorithms. KNN required minimal training time since it stores the training data without explicit learning, but it requires substantial storage capacity. It was found that using an ANN for this specific task is beneficial, as long as the time needed for training the model isn't a major concern.

Random Forest is a viable alternative for reducing training time. It was anticipated that neural networks would outperform traditional machine learning algorithms in the task of digit recognition, given that each pixel in the image serves as a parameter. Neural networks are generally recommended for scenarios involving a high number of parameters, where conventional machine learning methods might fail to identify essential correlations. However, in practice, the performance of the algorithms was found to be roughly equivalent. This may be attributed to the simplicity of the digit images, which allows machine learning algorithms to effectively classify the data [31].

3.3. Fashion-MNIST Classification

The value of *k* for the KNN classifier was set to 5, representing the number of neighbors. All hyperparameters for both SVM and Decision Trees were used with their default values. The Random Forest model consisted of 130 trees. The results were reported by computing the accuracy metric from the developed model over the utilized dataset's test set, as shown in Table 4.

Table 4. Results of classification accuracy on the Fashion-MNIST dataset.

Metric / Classifier	ACC	Training time/sec.	Time to predict test sample /sec.	Storage space/KB
ANN	0.830	140.00	0.640	1.36
KNN	0.8388	0.17	42.560	363.00
SVM	0.8828	392.98	170.680	125.00
DT	0.7919	77.28	0.029	2.00
RF	0.8751	195.12	1.718	266.00

The best prediction accuracy was achieved by the SVM algorithm, although the neural network followed closely. The Decision Tree model achieved relatively low accuracy. However, it trained quickly and required minimal memory. Compared with other machine learning algorithms, the accuracy of the SVM algorithm was high, although it required significantly more time for both training and prediction.

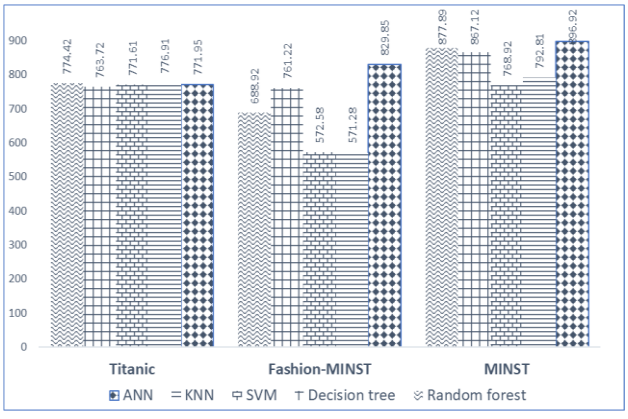


Figure 3. General diagram.

It was anticipated that the neural network would outperform traditional machine learning algorithms in clothing recognition. Although it was anticipated that the neural network would achieve the highest performance, the SVM algorithm slightly

outperformed it in terms of accuracy. Subsequently, we will summarize the study's results (see Figure 3).

Each approach is assigned a designated number of points according to a formula that involves multiplying each measurement by specific coefficients after all measurement data has been collected. The coefficient for accuracy is set at 1000, while the coefficients for training time, prediction time, and occupied space are -0.4, -0.6, and -0.4, respectively.

A substantial positive coefficient is selected for accuracy, as it is the most critical parameter and is evaluated in relative terms. In contrast, the coefficients for time and space utilization are assigned negative values, reflecting their inherently negative dimensions. A diagram is created for the Z-final scoring, illustrating that although certain models may exhibit high accuracy, they can receive low scores in other criteria, which can considerably influence the overall evaluation.

Consequently, it cannot be definitively asserted that neural networks possess a substantial advantage over machine learning algorithms. The findings indicate that for certain classification tasks, employing machine learning algorithms may be more appropriate, as the accuracy levels may remain relatively unchanged.

4. Conclusion

In this paper, we argue that the most effective method for assessing the strengths and weaknesses of various algorithms, including ANN, kNN, SVM, Decision Trees, and Random Forests, is to conduct a comprehensive evaluation across multiple datasets. Furthermore, the article examines the applicability of these algorithms to specific issues, highlighting that the ANN classifier demonstrated the highest accuracy in its results.

The researchers will persist in their efforts to evaluate the effectiveness of machine learning algorithms in comparison to neural networks across various tasks. They are particularly focused on areas such as traffic management, resource distribution, and the prediction of signal levels in 5G and 6G mobile networks.

Acknowledgments

I would like to express my sincere gratitude to my former supervisor Sanjar Abdoulaev for his invaluable support.

References

[1] Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics: A review. *Cognitive Robotics, 3*, 54–70. https://doi.org/10.1016/j.cogr.2023.04.001

[2] Salal, Y. K., Hussain, M., & Theodorou, P. (2021). Student next assignment submission prediction using a machine learning approach. In A. A. Radionov & V. R. Gasiyarov (Eds.), *Advances in Automation II. RusAutoCon 2020* (Lecture Notes in Electrical Engineering, Vol. 729). Springer. https://doi.org/10.1007/978-3-030-71119-1_38

[3] Hussain, S., Muhsion, Z. F., Salal, Y. K., Theodorou, P., Kurtoğlu, F., & Hazarika, G. C. (2019). Prediction model on student performance based on internal assessment using deep learning. *International Journal of Emerging Technologies in Learning, 14*(8), 4. https://doi.org/10.3991/ijet.v14i08.10001

[4] Kaggle. (n.d.). *Titanic – Machine learning from disaster*. Retrieved February 3, 2025, from https://www.kaggle.com/c/titanic/data

[5] Scikit-learn. (n.d.). *SVC*. Retrieved February 6, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[6] Scikit-learn. (n.d.). *KNeighborsClassifier*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[7] Scikit-learn. (n.d.). *DecisionTreeClassifier*. Retrieved February 5, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[8] Scikit-learn. (n.d.). *RandomForestClassifier*. Retrieved February 3, 2025, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[9] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Publ. Springer New York. https://doi.org/10.1007/978-0-387-84858-7

[10] Breiman, L., & Ihaka, R. (1984). *Nonlinear discriminant analysis via scaling and ACE* (Technical Report No. 40). Department of Statistics, University of California, Berkeley. https://digitalassets.lib.berkeley.edu/sdtr/ucb/text/40.pdf

[11] McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation, 24*, 109–165. https://doi.org/10.1016/S0079-7421(08)60536-8

[12] Brown, W. M., Gedeon, T. D., Groves, D. I., & Barnes, R. G. (2000). Artificial neural networks: A new method for mineral prospectivity mapping. *Australian Journal of Earth Sciences, 47*, 757–770. https://doi.org/10.1046/j.1440-0952.2000.00807.x

[13] Rigol-Sanchez, J. P., Chica-Olmo, M., & Abarca-Hernandez, F. (2003). Artificial neural networks as a tool for mineral potential mapping with GIS. *International Journal of Remote Sensing, 24*, 1151–1156. https://doi.org/10.1080/0143116021000031791

[14] Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In

Proceedings of the fifth annual workshop on Computational Learning Theory (pp. 144–152). https://doi.org/10.1145/130385.130401

[15] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297. https://doi.org/10.1007/BF00994018

[16] Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32. https://doi.org/10.1023/A:1010933404324

[17] Rodriguez-Galiano, V. F., & Chica-Rivas, M. (2014). Evaluation of different machine learning methods for land cover mapping of a Mediterranean area using multi-seasonal Landsat images and digital terrain models. *International Journal of Digital Earth, 7*, 492–509. https://doi.org/10.1080/17538947.2012.748848

[18] Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees* (Vol. 37). Wadsworth. https://doi.org/10.1201/9781315139470

[19] Al-Anazi, A., & Gates, I. D. (2010). A support vector machine algorithm to classify lithofacies and model permeability in heterogeneous reservoirs. *Engineering Geology, 114*, 267–277. https://doi.org/10.1016/j.enggeo.2010.05.005

[20] Ben-Hur, A., Horn, D., Siegelmann, H., & Vapnik, V. N. (2001). Support vector clustering. *Journal of Machine Learning Research, 2*, 125–137.

[21] Chen, W., Xie, X. S., Wang, J. L., Pradhan, B., Hong, H. Y., Bui, D. T., Duan, Z., & Ma, J. Q. (2017). A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility. *Catena, 151*, 147–160. https://doi.org/10.1016/j.catena.2016.11.032

[22] Gutiérrez, S. L. M., Rivero, M. H., Ramírez, N. C., Hernández, E., & Aranda-Abreu, G. E. (2014). Decision trees for the analysis of genes involved in Alzheimer's disease pathology. *Journal of Theoretical Biology, 357*, 21–25. https://doi.org/10.1016/j.jtbi.2014.05.002

[23] Ghimire, B., Rogan, J., Galiano, V. R., Panday, P., & Neeti, N. (2012). An evaluation of bagging, boosting, and random forests for land-cover classification in Cape Cod, Massachusetts, USA. *GIScience & Remote Sensing, 49*, 5, 623–643. https://doi.org/10.2747/1548-1603.49.5.623

[24] Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J. P. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing, 67*, 93–104. https://doi.org/10.1016/j.isprsjprs.2011.11.002

[25] Pierdicca, R., Malinverni, E. S., Piccinini, F., Paolanti, M., Felicetti, A., & Zingaretti, P. (2018). Deep convolutional neural network for automatic detection of damaged photovoltaic cells. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42*, 893–900. https://doi.org/10.5194/isprs-archives-XLII-2-893-2018

[26] Funaya, H., & Ikeda, K. (2012). A statistical analysis of soft-margin support vector machines for non-separable problems. In *2012 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–7). IEEE. https://doi.org/10.1109/IJCNN.2012.6252443

[27] Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017). EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 2921–2926). IEEE. https://doi.org/10.1109/IJCNN.2017.7966217

[28] Meshkini, K., Platos, J., & Ghassemain, H. (2020). An analysis of convolutional neural network for fashion images classification (fashion-MNIST). In *Proceedings of the Fourth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'19)*, *4*, (pp. 85–95). Springer International Publishing. https://doi.org/10.1007/978-3-030-50097-9_10

[29] Bi, W., & Kwok, J. (2013). Efficient multi-label classification with many labels. In *Proceedings of the 30th International Conference on Machine Learning, 28* (pp. 405–413). PMLR.

[30] Zhang, Z. (2018). Improved Adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)* (pp. 1–2). IEEE. https://doi.org/10.1109/IWQoS.2018.8624183

[31] Chauhan, N. K., & Singh, K. (2018, September). A review on conventional machine learning vs deep learning. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)* (pp. 347–352). IEEE. https://doi.org/10.1109/GUCON.2018.8675097