[1]Gagandeep Kaur,
[2]Satish Saini

# Performance Evaluation of Machine Learning Algorithms- SVM, DT, CNN and KNN

Journal of Electrical Systems

***Abstract: -*** In today's data-rich era, digging into information and pulling out valuable insights has become crucial. Artificial intelligence, powered by smart algorithms that understand data patterns, plays a key role in making this happen. It's not just a tech thing—it's everywhere, from bioinformatics and marketing to gaming and virus detection. To figure out how well different machine learning tools like Support Vector Machines (SVM), Decision Trees (DT), and Convolutional Neural Networks (CNN) work, we did a thorough study. A comparative study was conducted to find out how accurate, precise, and reliable they are, using metrics like recall and F-score.

***Keywords:*** Machine Learning, Artificial Intelligence, SVM, DT, KNN, CNN

## INTRODUCTION

Recent advancements in algorithms & computing capabilities have boosted interest in machine learning (Schmidt, 2019). These algorithms are applied for tasks like classification, regression and clustering, and are particularly effective at handling large high-dimensional data sets (Bojarski et al., n.d.). Machine learning has even been shown to surpass human abilities in areas like self-driving cars and image recognition (He et al., 2015)(Liu & Tian, 2010)(Pazzani, 1997). Consequently, it now impacts many facets of daily life, such as web search, speech recognition and fraud detection (Guzella & Caminhas, 2009)(Huang et al., 2007). While machine learning has a extensive experience of application in fields like biology and chemistry, it has only recently become prominent in the field of science and technology (Baldi & Brunak, 2001)(Noordik, 2004).

Traditionally, experiments have been crucial in the pursuit and identification of new materials. However, these studies can be time-consuming and require significant resources and equipment, limiting the number of materials that can be examined(Martin, 2020). These limitations have often led to breakthrough discoveries being attributed to human intuition or luck. Recently, the field of materials science has experienced a mathematical revolution with the integration of computational methods, including density function theory (DFT), Monte Carlo, and molecular dynamics (Hohenberg & Kohn, 1964)(Oganov, 2011)(Walsh, 2015). These methods have greatly enhanced the efficiency of exploring the properties and structures of materials. Materials science that blends experimental & computational methods saves time

[1]Research Scholar PhD RIMT University, Mandi Gobindgarh, Punjab, India and Assistant Professor, Chandigarh Engineering College, Chandigarh College of Engineering, Jhanjeri, Mohali, Punjab-140307, India
Kaurgagan10deep@gmail.com
[2]Professor, RIMT University, Mandi Gobindgarh, Punjab, India
*Corresponding Author: satishsainiece@gmail.com

& cost. Improved computing & efficient code now enable large-scale simulations & calculations, enhancing results when combined with experiments provide a vast amount of data that can be analyzed using machine learning techniques (Faber et al., 2016)(Wu et al., 2018). This approach allows for the identification of ideal experimental candidates and facilitates the continuous growth in the field of materials and structures (Jalem et al., 2018)(Rosenblatt, 1958).

Towards the impending computer revolution, machine learning algorithms will be incorporated into materials science. With the high number of potential materials available, these algorithms can play a crucial role in bridging the gap between experiment and theory through initiatives such as the Materials Genome Initiative (McCulloch & Pitts, 1943)(Ye et al., n.d.). This shift promotes more intensive and systematic data research, leading to successful applications such as predicting new, stable materials, calculating various material properties, and increasing the speed of calculations based on first principles.

Machine learning application in materials science is in its nascent stage, with most published works being unadorned. They usually involve training models on small datasets or using machine learning techniques for tasks that can be done with conventional methods. Although simple tasks on small datasets can be accomplished using machine learning, this approach does not fully exploit the capabilities of these methods and does not reflect the achievements seen in other domains.

It's crucial to use accurate classification when entering a new scientific field. Misusing terms like "deep learning" for non-deep learning work can mislead newcomers to the field and misrepresent deep learning capabilities (Ye et al., n.d.). Deep learning's success lies in its capacity to learn multi-level data representations without human input, something not achievable with two-layer neural networks.

The implementation of machine learning algorithms in science has been criticized for their limited ability to produce new insights and knowledge. This is due to their perceived status as a "black box," with intricate models and processes that are hard for humans to comprehend. This discussion will assess the legitimacy of this criticism and explore solutions. We'll delve into machine learning software and algorithms, offering in-depth analysis and examination.

## 1. LITERATURE

Modeling and optimization in machining can be complex tasks, requiring specific approaches to achieve high-quality products at a low cost. In recent years, researchers have turned to computational methods such as support vector machine (SVM), convolutional neural network (CNN), decision tree (DT), and k-nearest neighbors (KNN) and other algorithms to model machining processes. This discussion will focus on these four machine learning algorithms.

### *I. SVM Overview*

The Support Vector Machine (SVM) is a potent computational method for modeling and optimization in machining, founded on statistical learning theory and Vapnik's structural risk minimization (SRM) principle (Deris et al., 2011). This method allows for the generation of

decision-making rules with a small error rate for independent test sets, effectively solving learning problems (Kaur et al., 2022b). In recent years, SVM has been utilized to address issues such as nonlinearity, local minima, and high dimensionality in machining. With its ability to achieve high accuracy in long-term predictions, SVM has become a popular choice in many practical applications compared to other computational approaches.

"The training dataset, D, is composed of pairs of input and output data, represented as (xi, yi) where i ranges from 1 to l, with l being the total number of training data pairs. The output data is often referred to as di, where d represents the desired target value. As such, SVM falls under the category of supervised learning techniques."

SVM has several key advantages, including a small number of parameters to adjust, a unique and globally optimal solution for a specific type of optimization problem, and strong generalization efficiency resulting from application of the structural risk minimization principle. These benefits have led to a significant amount of research on the theory and application of SVM in various fields (Xue & Guodong, 2016)(Fachrurrozi et al., 2017).

Support Vector Machine (SVM) approach relying on decision plane concept, is known for its ability to outperform traditional techniques. This is due to its utilization of the structural risk minimization (SRM) principle as opposed to the conventional empirical risk minimization technique.

The SVM algorithm creates a hyperplane, also known as a decision plane, to define the boundaries between classes. Support vectors, or the closest points to the hyperplane, are used to choose the optimal one (Kaur et al., 2022b)(Kaur et al., 2022a).

In the case of linear separability, the discrimination function needs to be normalized, ensuring that $|g(x)|=1$ for all training samples, even those located away from the classification surface. As a result, the class interval becomes equal to $2 / ||\omega||$, where $\omega$ represents the vector (Zhang, 2012).

For accurate classification of all samples, it is necessary to satisfy the following equation: $yi [(\omega.x_i )+b]-1\geq 0$, where $i=1,2,\ldots.n$. This equation ensures that the minimal classification surface coincides with the ideal classification surface, with the support vectors representing the closest points to the hyperplane.

## *II. DT overview*

A Decision Tree is a common tool for prediction and data classification. It assigns a class to each input in a manner that resembles a flowchart. The flowchart's interior nodes correspond to features, the branches to decisions made on the basis of those features, and the leaf nodes to class labels (Matzavela & Alepis, 2021). Decision trees are particularly useful for understanding the relationships between input variables and the output, and can be used to develop a set of rules for making predictions based on the observed data. The DT algorithm focuses on grouping the data into a set of disordered cases and using those cases to make predictions.

Decision trees are a popular tool for solving classification and prediction problems. They work by dividing the data into subsets using a "divide and conquer" approach, and then building a

tree to represent the categorization process. The tree is then used to classify new data by traversing it and making decisions based on input feature values (Kaur et al., 2022b). The process of building a decision tree involves two main steps: creating the tree structure and assigning data to the tree nodes. Because of its effectiveness and versatility, decision tree techniques are widely studied and used in various applications.

DT represents a combination of attribute value restrictions as a whole, with each root-to-leaf path signifying a conjunction. It can be transformed into simple IF-THEN rules, facilitating prediction and classification of unknown data (Kaur et al., 2022b).

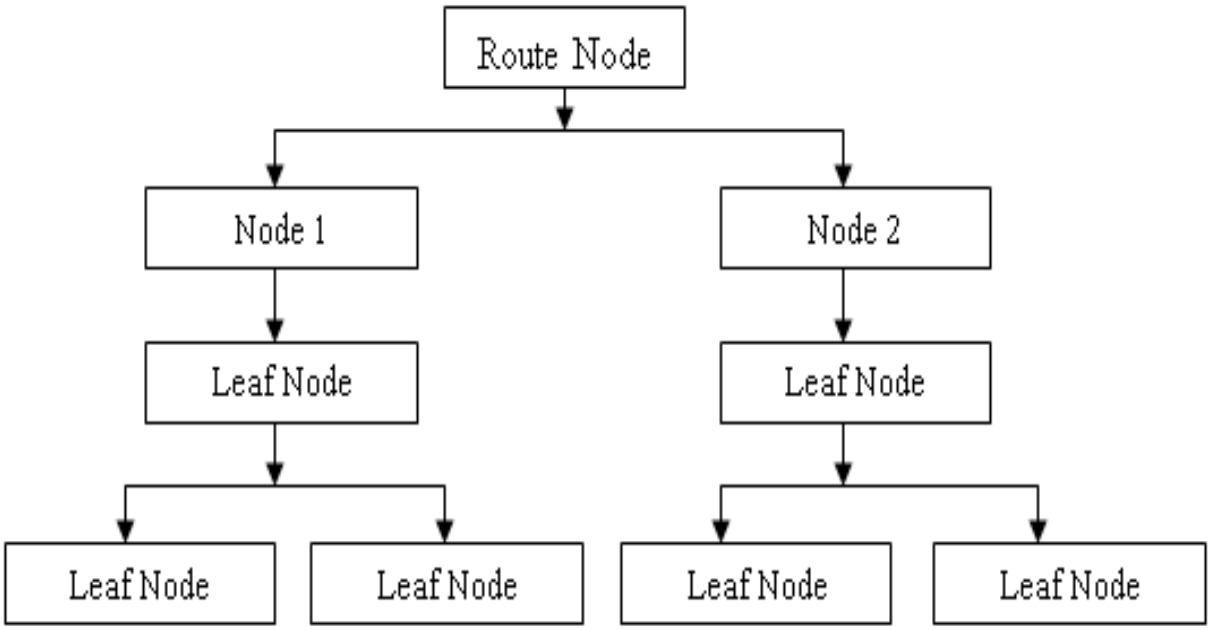


**Figure 1.** depicts the fundamental structure of a Decision Tree.

First, we need to define decision trees: Based on a data set D ($t_1$, $t_2$…., $t_n$), a decision tree [A1, A2, A3] contains the attributes of the dataset D (Kaur et al., 2022b)(Kaur et al., 2022a). Whereas, $t_i =< t_{i1},\ldots\ldots\ t_{in} >$.

Also, a given category C is composed of these entries: [ $C_1$, …,$C_m$].

The decision tree must should encompass the following attributes:

a. Each internal node is associated with Attribute A.
b. Predicates are assigned to each arc to apply to the attributes of the parent node.
c. Each leaf node is designated with Class C.

Classifying and predicting data with a Decision Tree involves two steps:

i. Building the tree through analysis of training data
ii. Applying the tree to classify all tuples in $t_D$

Decision Tree is a widely used classification technique in ML and data science. It creates a simple, interpretable, graphical model for classification and prediction, suitable for both numerical and categorical data (Kaur et al., 2022b)(Kaur et al., 2022a).

Decision Trees in Data Mining merge mathematical and computational methods to describe, categorize, and generalize data.

*III. KNN overview*

K-means is a type of unsupervised learning algorithm that solves a common clustering problem. Datasets requiring classification into multiple clusters can be grouped with a simple and straightforward method (assuming k clusters). Each cluster should have a k-center (Kaur & Saini, 2023) (Dharani & Aroquiaraj, 2014). To solve k-means clustering iterations:

Step1: Select k number of clusters

Step2: Designate centroids by selecting k random points from the dataset

Step3: Assign dataset points to their closest centroid

Step4: Designate new centroids for the formed clusters.

Step5: Repeat step3 and step4 until

i. Newly formed centroids do not change location
ii. Data points tend to remain in the same cluster i.e., data points do not change cluster
iii. Iteration limit has reached to its maximum

There must be the same number of clusters as data points or fewer. Minimizing the squared error function, J(v), is the goal as stated in the equation:

$$J(v) = \sum_{i=1}^{C} \quad \sum_{j=1}^{Ci} \quad \left(\left||xi - vj\right|\right)^2$$

In Euclidean geometry, ||xi-vj|| is the distance between two points (Kaur et al., 2022b)(Xue & Guodong, 2016).

Cluster i has ci data points and there are c cluster centers.

X={x1, x2, ……xn} denotes the set of data or observation points

V={v1, v2,…….vn} denotes the centers of clusters

*IV. CNN overview*

Cellular Neural Networks in a LAN environment consist of numerous cells, denoted by c (i, j), (i, j 1,2,3,….). They are nonlinear neural networks. Each cell connects to its eight neighboring cells through template parameters, affecting the intensity of each neighboring cell.

$N_r(i, j) = c(k, l): \max\{|k - i|, |i - j| \leq r\} \ldots\ldots 1 \leq k \leq m, 1 \leq l \leq n$

In the equation of a LAN-based Cellular Neural Network (CNN), r, k, and l are positive integers. The term c(k, l) refers to a single cell c(i, j) and its surroundings within a radius of r [37]. The equation assumes uniformity in the defined surroundings of the CNN, leading to the conclusion that symmetry exists if c(i, j) is in Nr(k, l) and c(k, l) is in Nr(i, j). This holds true for all c(i, j) and c(k, l) (Xue & Guodong, 2016)(Kaur & Saini, 2023).

CNNs maintain temporal and spatial displacement invariance by combining a convolutional and a pooling layer in feature extraction. They have four main layers: convolutional, pooling, fully connected, and output as shown in figure 2. It's common to stack multiple convolutional

and pooling layers. In the convolutional layer, each neuron's output feature is locally linked to its input, and the weight of each connection is used to weight the sum with the input and an offset value to determine the neuron's input, giving CNN its name, as it's equivalent to convolution (Lecun et al., 1998)(Bhatia & Author, 2010).

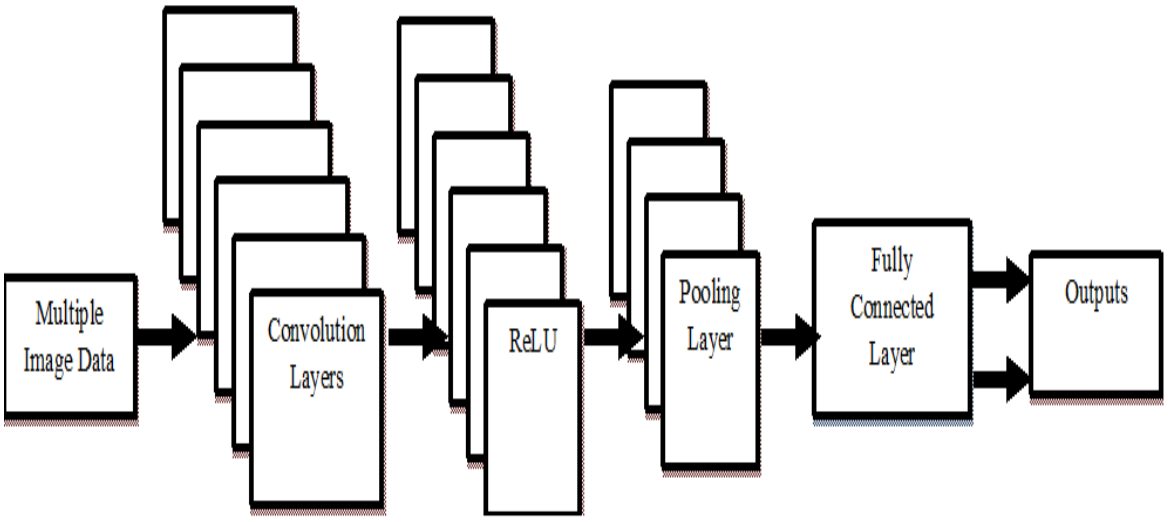


**Figure 2.** shows CNN Algorithm layers

## 2. PROPOSED SYSTEM

The suggested study's major goal is to undertake a thorough evaluation of four key machine learning algorithms: Support Vector Machines (SVM), Decision Trees (DT), Convolutional Neural Networks (CNN), and k-Nearest Neighbors (KNN). Several critical steps are included in the strategy. To begin, varied datasets covering a wide range of genres, sizes, and complexities will be chosen to ensure a full evaluation of algorithm performance. Extensive data preprocessing, which includes tasks such as managing missing values and normalizing data, attempts to improve dataset quality and reliability. The algorithms are then implemented using well-known libraries such as Scikit-learn and TensorFlow, with careful tuning of appropriate hyperparameters.

To ensure a fair comparison, all algorithms will use the same settings. To measure generalization performance and avoid overfitting concerns, cross-validation techniques, notably k-fold cross-validation, will be used. To assess algorithm performance, criteria such as accuracy, precision, recall, F1 score, and computational efficiency will be used. Understanding algorithm behavior and identifying strengths and shortcomings will be aided by visualizations such as confusion matrices and ROC curves. To establish the significance of observed performance differences, statistical analysis, including paired t-tests, will be performed.

A comparison study will be undertaken across datasets to find patterns and trends, providing insights into the adaptability and resilience of the algorithm.

## 3. RESULTS AND DISCUSSION

SVM, DT, KNN and CNN, we have implemented all the four algorithms (SVM, DT, KNN and CNN) in Python for evaluating their performance. The models were trained as well as tested before implementation. After successful implementation on labelled data, the results were calculated from confusion matrix and are shown in Table 1.

**Table 1.** comparison of SVM, DT, KNN and CNN

| Approaches | Accuracy | Precision | Recall | F-score |
|---|---|---|---|---|
| **KNN** | 93.23 | 93.45 | 94.32 | 93 |
| **SVM** | 95.23 | 94.34 | 95.34 | 94 |
| **Decision Tree** | 86.34 | 86.23 | 85.34 | 86.22 |
| **CNN** | 74.23 | 54.34 | 67 | 53.22 |

Here, Table 1 clearly shows that SVM is the superior in terms of precision, accuracy, f-score and recall. Whereas, CNN is inferior amongst all the four algorithms. All the parameters with original values are clearly depicted in the following graphs.

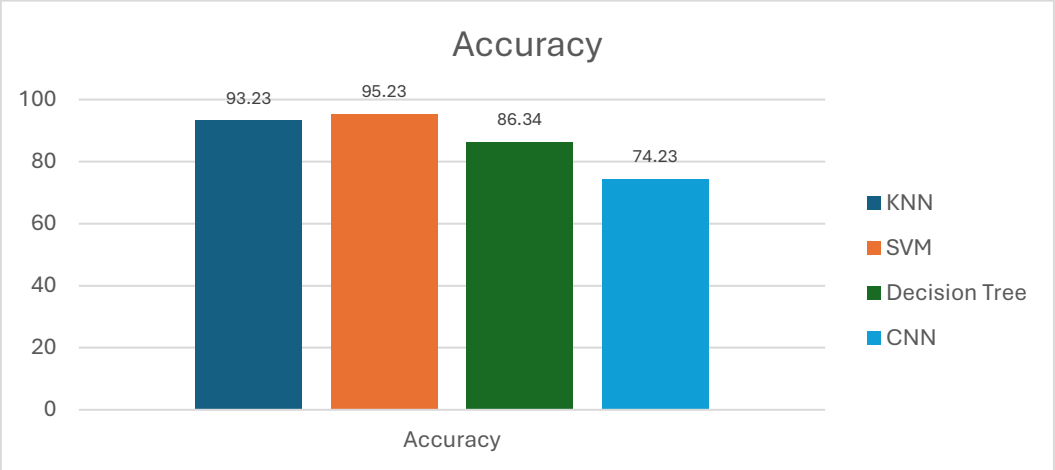


**Figure 3.** Depicts accuracy comparison of KNN, SVM, DT and CNN

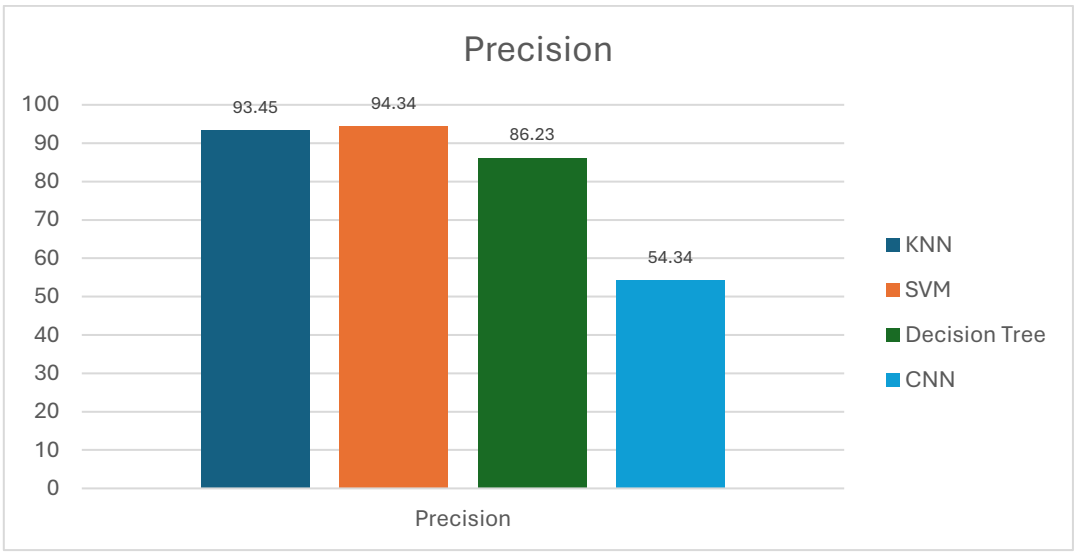


**Figure 4.** Depicts precision comparison of KNN, SVM, DT and CNN


**Figure 5.** Depicts recall comparison of KNN, SVM, DT and CNN

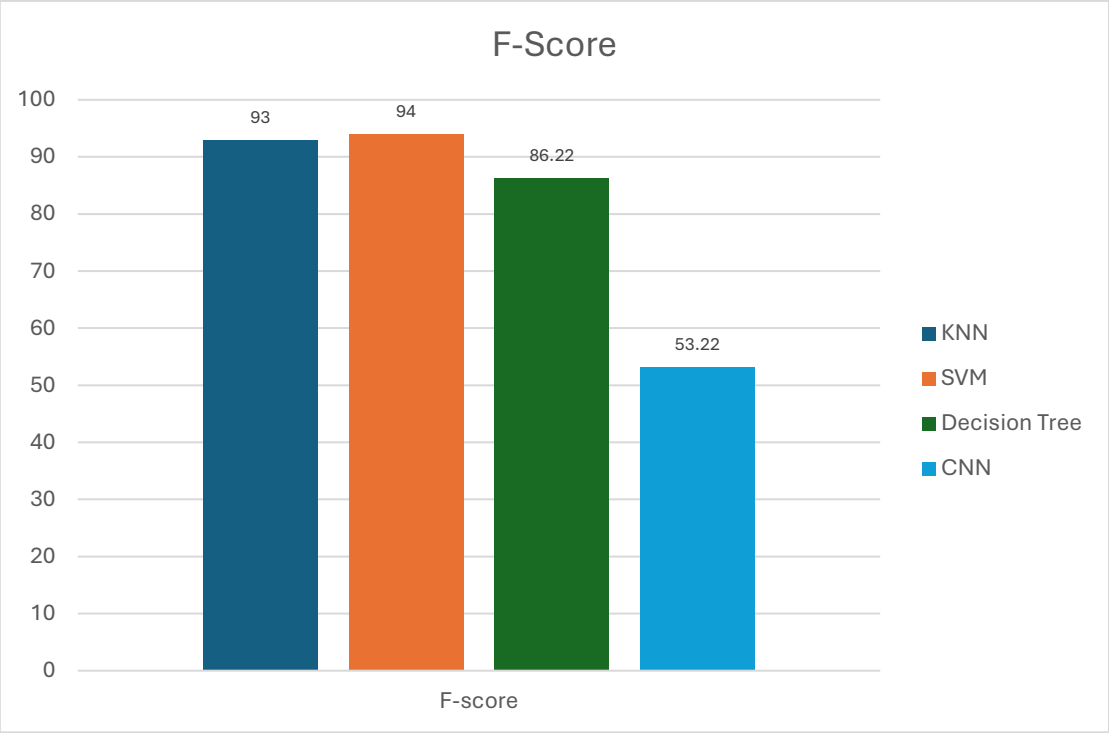


**Figure 6.** Depicts f-score comparison of KNN, SVM, DT and CNN

ML algorithms such as SVM, DT, KNN, and CNN are all employed for regression and classification problems. They are still distinct in their strategy and performance. Table 2. summarizes some important considerations based on algorithm research to consider before selecting the optimal algorithm for your task.

**Table 2.** Pros and cons of SVM, DT, KNN and CNN algorithms

| Algorithm | Type | Pros | Cons |
|---|---|---|---|
| **SVM** | Linear/non-linear classification and regression model | Can handle high-dimensional and non-linear data using kernel functions, computationally efficient, effective in handling noisy or overlapping data | Sensitive to kernel function and regularization parameter selection, challenging to interpret |
| **DT** | Decision-making algorithm for classification and regression | Simple and easy to comprehend, it is capable of handling categorical and numerical data, suitable for small datasets | Can suffer from overfitting, may require pruning or ensemble methods, not suitable for large or complex datasets |

| | | | |
|---|---|---|---|
| **KNN** | Non-parametric classification and regression model | Simple and effective, capable of handle numerical and categorial data, easy to interpret | Computationally expensive and requires large amounts of memory, appropriate for smaller databases with a big number of characteristics and just a handful of classes |
| **CNN** | Deep learning model for image and video classification and recognition | Highly effective for image and video classification, can automatically learn and extract features from the input data | Requires large amounts of data to train effectively, computationally expensive, can suffer from overfitting |

## **4.** CONCLUSIONS

This study conducted a thorough performance evaluation of four machine learning algorithms, namely SVM, DT, KNN, and CNN, by analyzing parameters derived from the confusion matrix. The evaluation was carried out using a comprehensive image dataset. Labelled data was used to evaluate the performances. Data set containing 4738 Images was used. The results of the implementation revealed that SVM outperformed the other algorithms in terms of precision, accuracy, f-score and recall. On the other hand, CNN demonstrated comparatively lower performance among the four algorithms.

Furthermore, the authors provided extensive information about the techniques and algorithms employed, the datasets utilized, and the outcomes pertaining to the algorithm parameters such as accuracy, precision, recall, and f-score for each of the four algorithms. The choice of algorithm depends on several factors, including the particular problem under consideration, the size and complexity of the dataset, and the desired performance metrics. SVM proves to be an excellent option for high-dimensional or non-linear data, DT is well-suited for small datasets containing categorical and numerical data, KNN is suitable for small datasets with a large number of features, and CNN is particularly effective for image and video classification tasks.

## **5.** ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to RIMT University for providing a suitable research environment and resources that aided in the execution of this work. The university's assistance has been critical to the success of this research project.

**Ethical Considerations:** The authors declare no conflicts of interests.

REFERENCES

[1] Baldi, P., & Brunak, S. (2001). *Bioinformatics: the machine learning approach*. MIT press.

[2] Bhatia, N., & Author, C. (2010). *Survey of Nearest Neighbor Techniques*. *8*(2), 302–305.

[3] Bojarski, M., Testa, D. Del, Goyal, P., Zhang, J., Dworakowski, D., Jackel, L. D., Firner, B., Monfort,

M., Zhao, J., & Zieba, K. (n.d.). *End to End Learning for Self-Driving Cars*. 1–9.

[4] Deris, A. M., Zain, A. M., & Sallehuddin, R. (2011). Overview of support vector machine in modeling machining performances. *Procedia Engineering*, *24*, 308–312.

[5] Dharani, T., & Aroquiaraj, I. L. (2014). *CONTENT BASED IMAGE RETRIEVAL SYSTEM with MODIFIED KNN ALGORITHM*.

[6] Faber, F. A., Lindmaa, A., Lilienfeld, O. A. Von, & Armiento, R. (2016). *Machine Learning Energies of 2 Million Elpasolite ð ABC 2 D 6 Þ Crystals*. *135502*(September), 2–7. https://doi.org/10.1103/PhysRevLett.117.135502

[7] Fachrurrozi, M., Fiqih, A., Saputra, B. R., & Algani, R. (2017). *Content Based Image Retrieval for Multi-Objects Fruits Recognition using k-Means and k-Nearest Neighbor*. 1–6.

[8] Guzella, T. S., & Caminhas, W. M. (2009). A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, *36*(7), 10206–10222.

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*, 1026–1034.

[10] Hohenberg, P., & Kohn, W. (1964). Inhomogeneous electron gas. *Physical Review*, *136*(3B), B864.

[11] Huang, C.-L., Chen, M.-C., & Wang, C.-J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications*, *33*(4), 847–856.

[12] Jalem, R., Kanamori, K., Takeuchi, I., & Nakayama, M. (2018). Bayesian-Driven First-Principles Calculations for Accelerating Exploration of Fast Ion Conductors for Rechargeable Battery Application. *Scientific Reports*, *March*, 1–10. https://doi.org/10.1038/s41598-018-23852-y

[13] Kaur, G., & Saini, S. (2023). Comparison of State Vector Machine and Decision Tree-Content Based Image Retrieval Algorithms to Perceive Accuracy. *2023 1st International Conference on Innovations in High Speed Communication and Signal Processing (IHCSP)*, 11–15.

[14] Kaur, G., Saini, S., & Sehgal, A. (2022a). Applications of Machine Learning and Deep Learning. In *Artificial Intelligence* (pp. 55–70). Chapman and Hall/CRC.

[15] Kaur, G., Saini, S., & Sehgal, A. (2022b). Machine Learning–Principles and Algorithms. In *Artificial Intelligence* (pp. 21–54). Chapman and Hall/CRC.

[16] Lecun, Y., Bottou, L., Bengio, Y., & Ha, P. (1998). *Gradient-Based Learning Applied to Document Recognition*. *November*, 1–46.

[17] Liu, S., & Tian, Y. (2010). *Facial Expression Recognition Method Based on Gabor Wavelet Features and Fractional Power Polynomial Kernel PCA *. 20071152*, 144–151.

[18] Martin, R. M. (2020). *Electronic structure: basic theory and practical methods*. Cambridge university press.

[19] Matzavela, V., & Alepis, E. (2021). Decision tree learning through a predictive model for student academic performance in intelligent m-learning environments. *Computers and Education: Artificial Intelligence*, *2*, 100035.

[20] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*, 115–133.

[21] Noordik, J. H. (2004). *Cheminformatics developments: History, reviews and current research*.

[22] Oganov, A. R. (2011). *Modern methods of crystal structure prediction*. John Wiley & Sons.

[23] Pazzani, M. (1997). *Learning and Revising User Profiles : The Identification of Interesting Web Sites*. *331*, 313–331.

[24] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386.

[25] Schmidt, J. (2019). Recent advances and applications of machine learning in solid- state materials science. *Npj Computational Materials*, *February*. https://doi.org/10.1038/s41524-019-0221-0

[26] Walsh, A. (2015). The quest for new functionality. *Nature Chemistry*, *7*(4), 274–275.

[27] Wu, Y., Sasaki, M., Goto, M., Fang, L., & Xu, Y. (2018). *Electrically Conductive Thermally Insulating Bi − Si Nanocomposites by Interface Design for Thermal Management*. https://doi.org/10.1021/acsanm.8b00575

[28] Xue, W., & Guodong, L. (2016). *Image Edge Detection Algorithm Research Based on the CNN ' s Neighborhood Radius Equals 2*. 115–119. https://doi.org/10.1109/ICSGEA.2016.38

[29] Ye, W., Chen, C., Wang, Z., Chu, I., & Ong, S. P. (n.d.). Deep neural networks for accurate predictions of crystal stability. *Nature Communications*, *2018*, 1–6. https://doi.org/10.1038/s41467-018-06322-x
[30] Zhang, Y. (2012). *Support Vector Machine Classification Algorithm and Its Application*. 179–180.