Journal of Big Data

RESEARCH Open Access

Optimizing classification efficiency with machine learning techniques for pattern matching

Belal A. Hamed[1*], Osman Ali Sadek Ibrahim[1] and Tarek Abd El-Hafeez[1,2]

*Correspondence:
Belal A. Hamed
belal.ahmed@mu.edu.eg
[1] Department of Computer Science, Faculty of Science, Minia University, EL-Minia, Egypt
[2] Computer Science Unit, Deraya University, EL-Minia, Egypt

Abstract
The study proposes a novel model for DNA sequence classification that combines machine learning methods and a pattern-matching algorithm. This model aims to effectively categorize DNA sequences based on their features and enhance the accuracy and efficiency of DNA sequence classification. The performance of the proposed model is evaluated using various machine learning algorithms, and the results indicate that the SVM linear classifier achieves the highest accuracy and F1 score among the tested algorithms. This finding suggests that the proposed model can provide better overall performance than other algorithms in DNA sequence classification. In addition, the proposed model is compared to two suggested algorithms, namely FLPM and PAPM, and the results show that the proposed model outperforms these algorithms in terms of accuracy and efficiency. The study further explores the impact of pattern length on the accuracy and time complexity of each algorithm. The results show that as the pattern length increases, the execution time of each algorithm varies. For a pattern length of 5, SVM Linear and EFLPM have the lowest execution time of 0.0035 s. However, at a pattern length of 25, SVM Linear has the lowest execution time of 0.0012 s. The experimental results of the proposed model show that SVM Linear has the highest accuracy and F1 score among the tested algorithms. SVM Linear achieved an accuracy of 0.963 and an F1 score of 0.97, indicating that it can provide the best overall performance in DNA sequence classification. Naive Bayes also performs well with an accuracy of 0.838 and an F1 score of 0.94. The proposed model offers a valuable contribution to the field of DNA sequence analysis by providing a novel approach to pre-processing and feature extraction. The model's potential applications include drug discovery, personalized medicine, and disease diagnosis. The study's findings highlight the importance of considering the impact of pattern length on the accuracy and time complexity of DNA sequence classification algorithms.

Keywords Bioinformatics, Feature extraction, Pattern matching, Machine learning, DNA sequences

Introduction

DNA is a kind of molecule that contains the genetic information needed by an organism to develop, survive, and reproduce. In addition, the sequencing of DNA is a technique used to identify the exact nucleotide sequences in a DNA molecule. The base sequence of DNA transmits the knowledge that a cell needs to assemble RNA and protein components. Additionally, DNA methylation is a genetic alteration important for controlling how the genome functions. It is important for both tumor suppression and carcinogenesis.

The suggested method can therefore be used with any genome or DNA sequence, it has been discovered. The suggested methods may be used with additional data kinds, such as larger datasets. The suggested approach may be used as a conduit element to provide a feeling of raw data that focuses on tiny sets of dimensions and reduces entropy [1]. Additionally, the categorization of biological sequences was one of several tasks in which the convolutional neural network (CNN) performed well. The available executions are frequently best for a certain task. Reusing it is challenging. According to this work, the suggested system can recover structural motifs and known sequences and conduct sequence classification with the highest accuracy compared to standard approaches [2]. Additionally, this work concentrated on effectively categorizing the DNA sequence using machine learning methods. To verify its efficacy, the suggested system is also examined in terms of a few performance measures. The primary contributions of this work are effective feature extraction and pre-processing for locating pertinent DNA data.

Therefore, researching data about DNA methylation may be useful to identify cancer biomarkers. Being able to analyze huge datasets efficiently is important given the abundance of publicly available data on matching methylation of DNA and the genome's large number of methylation regions. As a result, our work has successfully computed a variety of alternative categorization models.

To classify DNA sequences and accurately extract matched sequences using a pattern-matching technique. To assess the performance of the suggested model in terms of DNA sequence occurrence, execution time, F1-score, accuracy, precision, recall, and other metrics.

Related work

DNA patterns have evolved into a significant setting for Sequence Data Analysis (SDA) [3, 4] that helps predict a Sequence Function. (SF). It also investigates how the DNA patterns have evolved together. The model utilized machine learning (ML) techniques and was trained on an actual G4 [5] generation dataset. This approach demonstrates the use of feature engineering techniques to extract relevant information from DNA sequences and classifiers based on ML algorithms to predict the formation of specific DNA patterns.

Touati, R., et al. [6] focused on the classification of helitron families using a combination of machine learning algorithms and feature extraction from DNA sequences. By extracting specific characteristics from DNA sequences, a fresh set of features related to helitrons was obtained. This study showcases the application of feature engineering techniques to capture important properties of DNA sequences and the utilization of machine learning classifiers to automatically classify DNA sequences into different helitron families.

Norlin, S. in [7] explored the categorization of DNA sequences using Nearest Neighbor categorization (NNC) based on Variable Length Markov Chains (VLMC). VLMC information was stored using a Vantage Point Tree (VPT) for efficient retrieval. This demonstrates the use of feature engineering techniques involving VLMC for characterizing DNA sequences and NNC as a classifier to categorize the sequences based on their similarity.

Ryu, C., T. Lecroq, and K. Park in [8] introduced a MAS (Maximal Average Shift), which finds a PSO (Pattern Scan Order) and lengthens the average shift by increasing the length. Two additional expansions were examined in this study—MAS. Through the scan results of the previous frame, it first increases the Scan Speed (SS) of MAS. Using q-grams, the second expansion increases the MAS running duration. As a result, these algorithms demonstrated superior performance compared to traditional algorithms. Further information regarding this project can be found in various details [9–14].

ML methodology for PM from DNA sequences

We opted to build databases on DNA to examine the machine learning algorithms discussed in the following paragraphs. The rationale behind this decision was to sample some of the genes that we had worked on in our previous research endeavors [15], Our objective was to integrate automated learning algorithms and pattern-matching algorithms that are based on specific DNA sequences, in order to create a biological data collection that could be utilized in a classification process. We conducted experiments on a dataset that included DNA sequences, where we compared the effectiveness of searching for a specific pattern with other classification models, such as **Random Forest** [3, 16], **KNN**[16–20], **Naïve Bayes** [21–24], **Decision tree** [23, 25–30], and **Support Vector Machine**[18, 31–36] with **Linear**[37, 38], **RBF**[37, 39], and **sigmoid**[21, 40] classifiers, the results of these classifiers models are calculated by F1 score, recall, precision rate, execution time, and with the accuracy which calculates the most effective pattern-matching classifier. The comparison of DNA sequences is a crucial task in various fields of research, including molecular biology and genetics. To facilitate this task, our study utilizes a machine learning (ML) approach that combines pattern-matching algorithms with ML techniques. This approach enables efficient pattern-matching and comparison of DNA sequences, thereby aiding in the identification of specific query patterns.

Our methodology consists of several different phases, each of which plays a critical role in the overall process. The first phase involves the pre-processing of the DNA sequence data, which includes cleaning and filtering the data to remove any noise or irrelevant information. The pre-processed data is then subjected to feature extraction, which involves identifying and extracting relevant features from the data. In the next phase, the extracted features are used to create a model that can classify the DNA sequences based on their similarity to the query patterns. This step involves the use of various ML algorithms, including supervised and unsupervised learning techniques, to develop an accurate and efficient classification model.

Once the classification model is developed, the pattern-matching algorithm is applied to the DNA sequences to identify any matches with the query patterns. This step involves the efficient comparison of DNA sequences based on their similarity to the query patterns, thereby enabling the identification of specific patterns of interest.

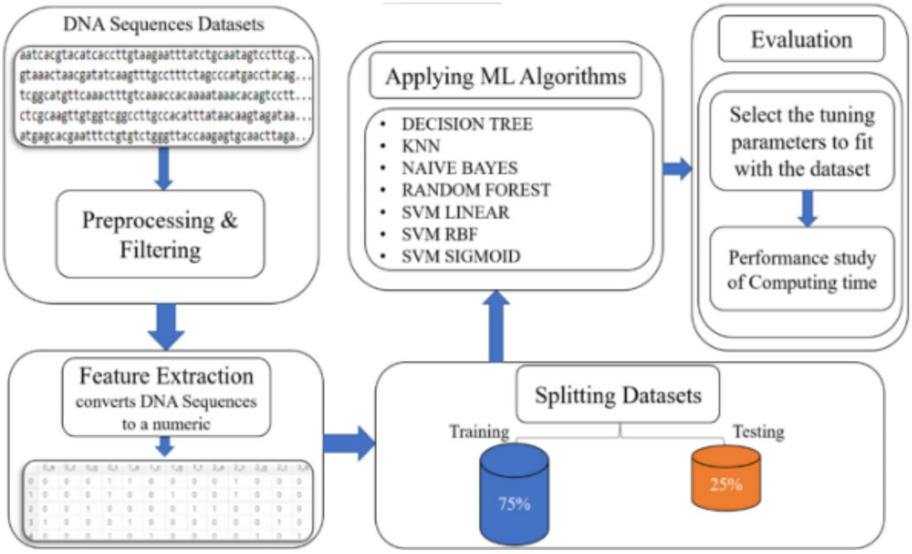
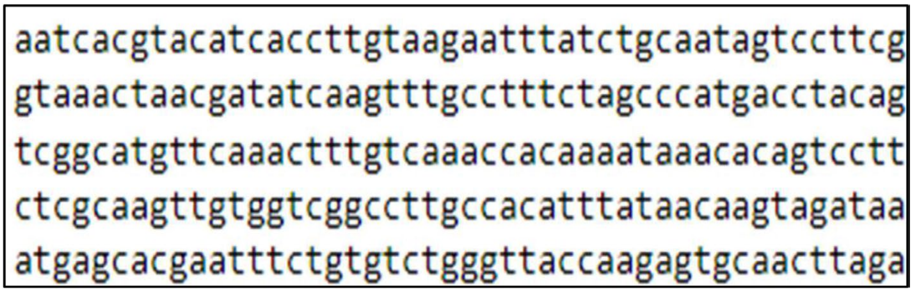

DNA Sequences Datasets
aatcacgtacatcaccttgtaagaatttatctgcaatagtccttcg...
gtaaactaacgatatcaagtttgcctttctagcccatgacctacag...
tcggcatgttcaaactttgtcaaaccacaaaataaacacagtcctt...
ctcgcaagttgtggtcggccttgccacatttataacaagtagataa...
atgagcacgaatttctgtgtctgggttaccaagagtgcaacttaga...
Preprocessing & Filtering
Feature Extraction
converts DNA Sequences to a numeric
Applying ML Algorithms
DECISION TREE
KNN
NAIVE BAYES
RANDOM FOREST
SVM LINEAR
SVM RBF
SVM SIGMOID
Evaluation
Select the tuning parameters to fit with the dataset
Performance study of Computing time
Splitting Datasets
Training
Testing
75%
25%

Fig. 1 Framework of the proposed work



```
aatcacgtacatcaccttgtaagaatttatctgcaatagtccttcg
gtaaactaacgatatcaagtttgcctttctagcccatgacctacag
tcggcatgttcaaactttgtcaaaccacaaaataaacacagtcctt
ctcgcaagttgtggtcggccttgccacatttataacaagtagataa
atgagcacgaatttctgtgtctgggttaccaagagtgcaacttaga
```

Fig. 2 Sample of FASTA Dataset

The overall framework is illustrated in Fig. 1, which provides an overview of the different phases involved in our ML approach for pattern-matching in DNA sequences. By combining both pattern-matching algorithms and ML techniques, our approach enables the efficient search of DNA sequences for specific query patterns, thereby facilitating the identification of critical information for various applications in molecular biology and genetics.

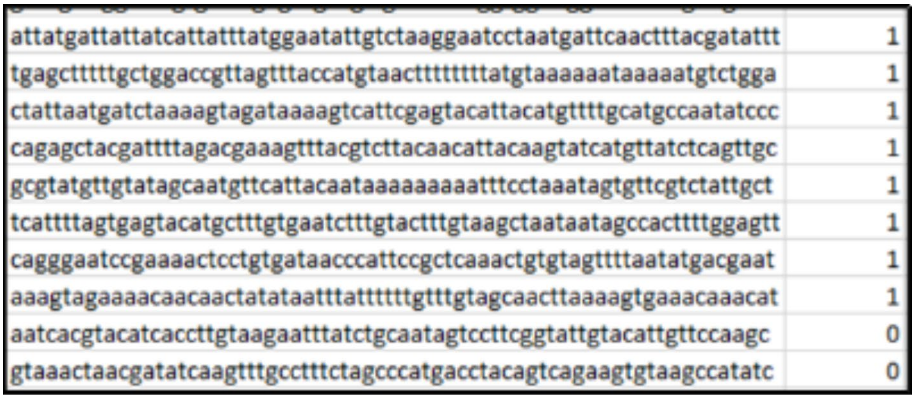
Pre-processing step

Gathering the dataset

In this study, we acquired a set of biological DNA sequences from "The National Centre for Biotechnology Information (NCBI)" (https://www.ncbi.nlm.nih.gov) [41]. The DNA data is stored in the (FASTA) format and is comprised of genomic sequences. Upon analyzing the dataset, we noticed that there was an unbalanced dataset problem, which needed to be addressed during the pre-processing stage. Additionally, the genomic sequence of the DNA dataset is categorical, which presents unique challenges for analysis. The DNA Sequences in FASTA files are illustrated in Fig. 2, and they feature sequence sizes of over 12 million characters of ATCG. This large dataset requires careful handling during the pre-processing stage to ensure that the data is clean, relevant, and appropriately prepared for analysis.

Table 1 Sample of the converted DNA Sequences from the FASTA file to a CSV file

DNA Sequence	label
aatcacgtacatcaccttgtaagaatttatctgcaatagtccttcggtattgtacattgttccaagcatag	1
gtaaactaacgatatcaagtttgcctttctagcccatgacctacagtcagaagtgtaagccatatcactg	1
tcggcatgttcaaactttgtcaaaccacaaaataaacacagtccttgaaatcgaatacgtagtttacatt	1
ctcgcaagttgtggtcggccttgccacatttataacaagtagataagcgtacggggcatgctttcccagt	1
atgagcacgaatttctgtgtctgggttaccaagagtgcaacttagacattcatctttatacactcgaaag	1
tgctttggaaggaagatctggccatataaatttactgcatgctcttactggtcagtttgctacaagcttt	0
gtgcggaggtatggcattttaatgttgagcaacgttcagtcgttcgtcgttggcaagttcaagatggtgt	0



attatgattattatcattatttatggaatattgtctaaggaatcctaatgattcaactttacgatattt	1
tgagctttttgctggaccgttagtttaccatgtaacttttttttatgtaaaaaataaaaatgtctgga	1
ctattaatgatctaaaagtagataaaagtcattcgagtacattacatgttttgcatgccaatatccc	1
cagagctacgattttagacgaaagtttacgtcttacaacattacaagtatcatgttatctcagttgc	1
gcgtatgttgtatagcaatgttcattacaataaaaaaaaatttcctaaatagtgttcgtctattgct	1
tcattttagtgagtacatgctttgtgaatctttgtactttgtaagctaataatagccacttttggagtt	1
cagggaatccgaaaactcctgtgataacccattccgctcaaactgtgtagttttaatatgacgaat	1
aaagtagaaaacaacaactatataatttatttttttgtttgtagcaacttaaaagtgaaacaaacat	1
aatcacgtacatcaccttgtaagaatttatctgcaatagtccttcggtattgtacattgttccaagc	0
gtaaactaacgatatcaagtttgcctttctagcccatgacctacagtcagaagtgtaagccatatc	0

Fig. 3 Sample of CSV Dataset

Transforming the data

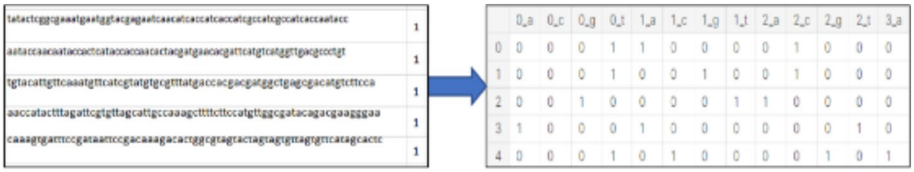
After consolidating all of the DNA sequence data, the next step involved transforming the data to make it suitable for machine learning model training. There are various methods for data transformation, depending on the type of data and business requirements. In our study, we converted the DNA Sequences in the FASTA file to a CSV file. We collected data for a group of biological DNA sequences by categorizing each group and assigning them a label (e.g., 1). We also added some sequences that were not related to DNA and assigned them to a separate category (e.g., 0). We created this dataset based on the use of FASTA DNA sequence files for specific genes. We converted the FASTA files for a particular gene to a CSV file and selected some of the sequences from it, labeling them to indicate that they belonged to this gene (e.g., 1). We also added another sequence that was not associated with this gene and labeled it accordingly (e.g., 0). Table 1 displays a sample of the conversion process from DNA Sequences from the FASTA file to a CSV file.

Cleaning and labeling the dataset

After completing the conversion process from DNA Sequences in the FASTA file to a CSV file, we ensured that the resulting dataset was valid and that each sample had a corresponding class label (1 or 0). Figure 3 provides a sample of the cleaned CSV dataset, demonstrating the successful completion of the data pre-processing stage.

Feature extraction

The extraction of important features is a critical phase in our analysis as irrelevant features can negatively impact the efficiency of the ML classifier. By selecting features



tatactcggcgaaatgaatggtacgagaatcaacatcaccatcaccatcgccatcgccatcaccaatacc	1
aataccaacaataccactcataccaccaacactacgatgaacacgattcatgtcatggttgacgccctgt	1
tgtacattgttcaaatgttcatcgtatgtgcgtttatgaccacgacgatggctgagcgacatgtcttcca	1
aaccatactttagattcgtgttagcattgccaaagcttttcttccatgttggcgatacagacgaagggaa	1
caaagtgatttccgataattccgacaaagacactggcgtagtactagtagtgttagtgttcatagcactc	1

	0_a	0_c	0_g	0_t	1_a	1_c	1_g	1_t	2_a	2_c	2_g	2_t	3_a
0	0	0	0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	0	0	1	0	0	0
2	0	0	1	0	0	0	0	1	1	0	0	0	0
3	1	0	0	0	1	0	0	0	0	0	0	1	0
4	0	0	0	1	0	1	0	0	0	0	1	0	1

Fig. 4 Conversion Datasets to numeric variables

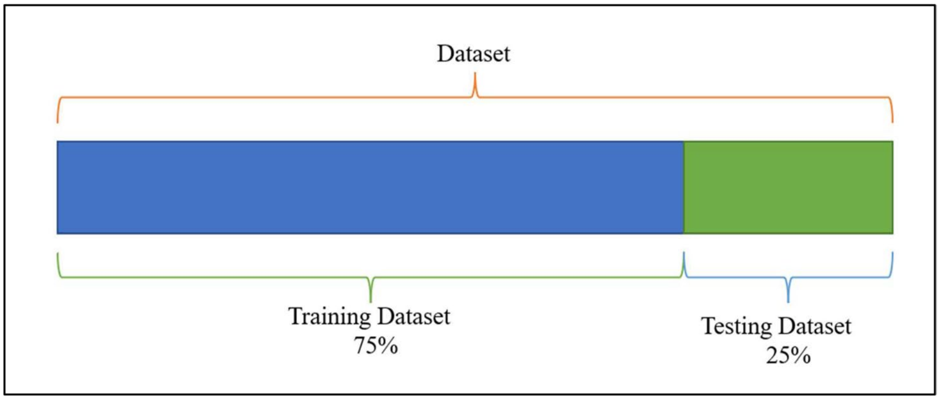


Fig. 5 Splitting Datasets

appropriately, we can enhance classification accuracy and reduce the training time for the model. Feature extraction involves breaking down vast amounts of raw data into smaller groupings for processing. Due to the vast number of variables in these massive datasets, processing them requires significant computer resources.

However, we cannot run machine learning algorithms on data in 'Sequence' (text) formats. Therefore, we must preprocess the data to transform it into a usable format for our algorithms. In this case, we convert the DNA sequence data into numerical data. There are several methods for calculating the numerical value for each feature. To accomplish this, we utilize the **GET_DUMMIES** function from the **Pandas Library**[42–44]. This function converts DNA Sequences to a numeric variable that encodes categorical information. Dummy variables have two possible values: 0 or 1. Once we have transformed the data into numerical format, we can use it in ML models for classification. The conversion of datasets to numeric variables is illustrated in

Train/test splitting

To ensure an unbiased evaluation of the machine learning model, it is essential to use data that was not previously used in the training process. Therefore, we need to split the collected dataset into separate training and testing datasets. In this step, after converting the data into a usable format for our ML model, we divided the dataset into a testing set and a training set. The training set was used to build and train our models using various classification algorithms, while the test set was used to evaluate the trained models on unseen data. The train_test_split function was utilized to split the data into a 25% testing set and a 75% training set. Additionally, stratify splitting was applied to ensure that the same split percentage was applied for each class in our data. The process of splitting the dataset into training and testing sets is depicted in Fig. 5.

Defining algorithms

After completing the data pre-processing and splitting the dataset into training and testing sets, we can begin deploying various classification algorithms. To ensure a comprehensive evaluation of the models, we will compare the performance of seven different algorithms. The training data comprises 75% of the dataset, and these classification methods are used to train the models. The remaining 25% of the data is reserved for testing, and various metrics are used to evaluate the performance of the machine learning algorithms in conducting DNA sequence pattern-matching and retrieving matched sequences. To create the algorithms, we need to import each algorithm we intend to use and various performance measures from the SKLEARN library [44], such as accuracy_score and classification_report, for each ML algorithm. The machine learning methods employed include (K-Nearest Neighbors) KNN, (Decision Tree) DT, (Random Forest) RF, (Naive Bayes) NB, Support Vector Machines (SVM) SIGMOID, SVM LINEAR, and SVM RBF, as depicted in Fig. 6.

Fitting models

Model fitting is the process of assessing how well a classification machine learning model generalizes to a dataset that is similar to the one on which it was trained. A well-fitted model produces more accurate results, whereas an overfitted model closely fits the training data and may not perform well on new data. An underfitted model, on the other hand, does not fit the training data closely enough and may not capture the underlying patterns and relationships in the data.

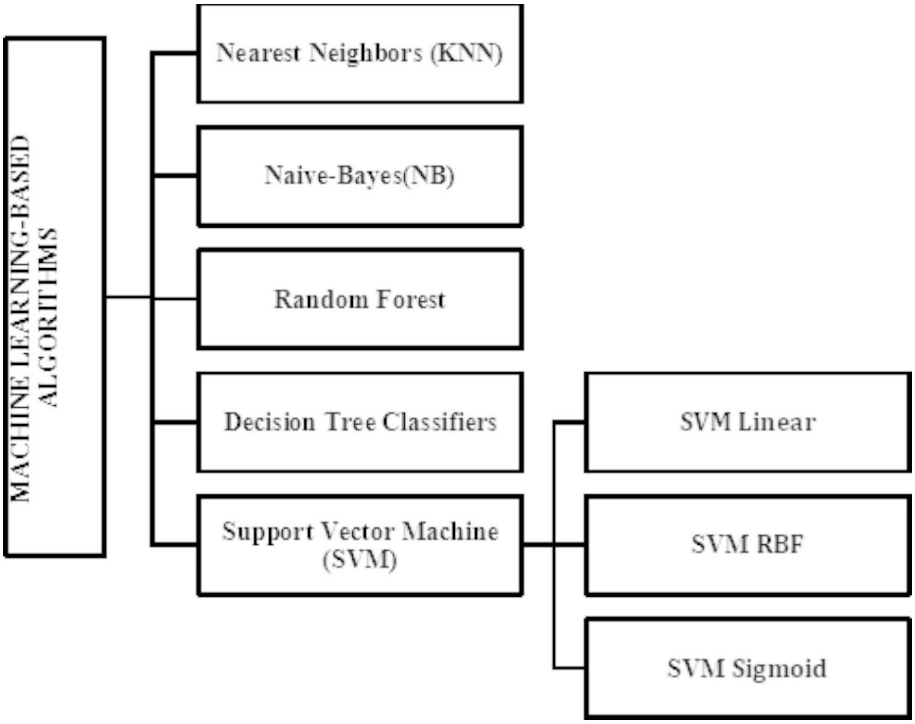
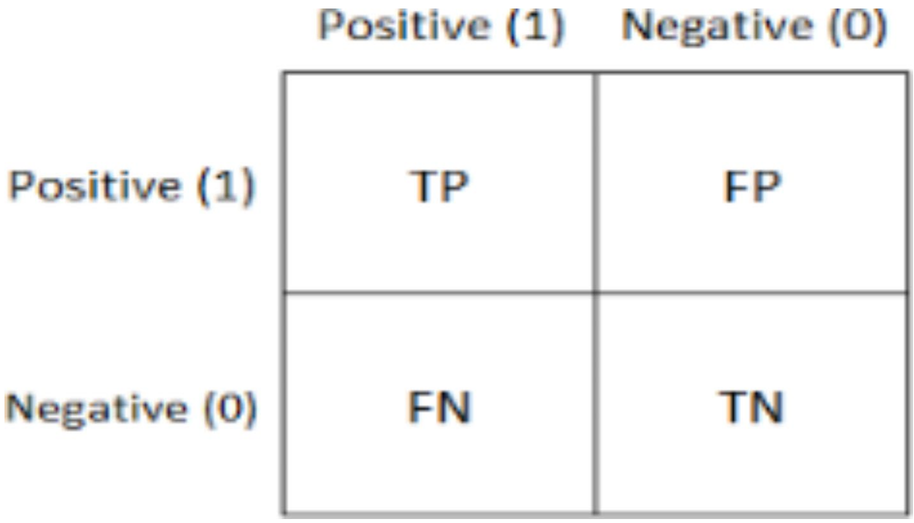

MACHINE LEARNING-BASED ALGORITHMS
Nearest Neighbors (KNN)
Naive-Bayes(NB)
Random Forest
Decision Tree Classifiers
Support Vector Machine (SVM)
SVM Linear
SVM RBF
SVM Sigmoid

Fig. 6 Classification Algorithms Used



	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

Fig. 7 The Confusion Matrix

Tuning parameters

Modifying parameters before executing a training operation can help regulate the behavior of our ML algorithm. These parameter settings can significantly impact model training in terms of training duration, model accuracy, and model convergence. In this stage, we experiment with varying the algorithm's parameters to find the best classification model solutions. For instance, when using the KNN algorithm, we set K=3 to achieve the highest accuracy in identifying the DNA dataset. In the Decision Tree algorithm, we use max depth=5 while classifying the dataset, which provides an optimal solution. For the RF model, we found that setting max depth=5 and n estimators=10 yielded the best classification results. We also employ several kernels to ensure the accuracy of SVM Classification, including SVM RBF, SVM SIGMOID, and SVM LINEAR, while using the SVM method. By experimenting with different parameter settings, we can optimize the behavior of our ML algorithm and achieve better classification performance.

Table 2 provides a comparison of different classification algorithms based on their time complexity, advantages, and disadvantages. The table lists the algorithms in rows, with columns indicating the time complexity, advantages, and disadvantages of each algorithm. The time complexity column provides an estimate of the time required by each algorithm to complete the classification task. The advantages column lists the strengths and benefits of each algorithm, such as high accuracy, robustness, and interpretability. The disadvantages column highlights the limitations and weaknesses of each algorithm, such as high computational complexity, overfitting, and poor performance on imbalanced datasets.

Some of the algorithms listed in Table 2 include Naive Bayes, Random Forest, K-Nearest Neighbors, Decision Trees, and Support Vector Machines (SVMs). For example, Naive Bayes is known for its simplicity and efficiency, making it a popular choice for text classification tasks. However, its main disadvantage is its assumption of independence between features, which can result in poor performance when dealing with highly correlated features.

Table 2 Classification algorithms Comparisons

Algorithms	Time Complexity	Advantages	Disadvantages
KNN	O (n * d) Where: n: the number of instances, d: dimensions	1. There is no training period- KNN. 2. Simple Implementation	1. It does not perform well with huge datasets. 2. Does not function properly with several dimensions. 3. Sensitive to missing and noisy data 4. Scaling of Features
SVM	O(s*d) Where: s: number of SV, d: data dimensionality	1. In higher dimensions, it performs effectively. 2. When classes can be separated, the best algorithm is used. 3. Outliers have less influence. 4. SVM is well-suited for binary classification in extreme cases.	1. Slower with bigger datasets 2. Overlapped classes perform poorly. 3. It is critical to choose proper hyperparameters. 4. Choosing the right kernel function might be difficult.
Decision Tree	O(k) Where: k: depth of tree	1. No data normalization or scaling is required. 2. Missing value handling 3. Feature selection that is automatic	1. Susceptible to overfitting. 2. Data sensitivity. When data changes little, the consequences might alter dramatically. 3. It takes more time to train decision trees.
Random Forest	O(k*m) Where: k: depth of tree, m: decision trees	1. Error reduction 2. Excellent performance on unbalanced datasets 3. Dealing with massive amounts of data 4. Effective handling of missing data 5. Outliers have little influence	1. Features must have some predictive power, or they will not operate. 2. The tree predictions must be uncorrelated.
Naive Bayes	O(n*d)	1. Scalable when dealing with large datasets. 2. Insensitive to unimportant characteristics. 3. Effective multi-class prediction 4. High dimensional performance with good performance	1. The independence of characteristics is not valid. 2. Training data should accurately represent the population.

Random Forest, on the other hand, is known for its high accuracy and robustness to noise and outliers. However, its training time can be significant, and it may suffer from overfitting when dealing with highly complex or imbalanced datasets.

K-Nearest Neighbors is a simple and effective algorithm that can be used for both classification and regression tasks. However, its main disadvantage is its high computational complexity, which can make it impractical for large datasets.

Decision Trees are easy to understand and interpret, making them a popular choice for applications where interpretability is important. However, they can suffer from overfitting and poor performance on imbalanced datasets.

Finally, SVMs are known for their high accuracy and ability to handle complex datasets. However, their training time can be significant, and they may require careful selection and tuning of hyperparameters to achieve optimal performance.

Overall, Table 2 provides a useful summary of the advantages and disadvantages of different classification algorithms and can help guide researchers in selecting the most appropriate algorithm for their specific task and dataset.

Evaluation

In this phase, several classification measurements were applied to report the performance of each model. These measurements are accuracy, recall, precision, and F1-score. All those measurements cannot be calculated without a confusion matrix [45, 46].

The Confusion Matrix is a performance statistic for a machine learning classification task where the output might be two or more classes. It is a table with four alternative combinations of projected and actual values, as shown below.

Accuracy

is a measure of how close a measurement is to the truth, represented as the percentage of correctly classified instances. It is calculated using the equation:

$$\text{Accuracy} = (\text{TP} + \text{TN}) \ / \ (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

where TN refers to the correct number of classifications of negative instances, TP refers to the correct number of classifications of positive instances, FP refers to the incorrect number of classifications of negative instances, and FN refers to the incorrect number of classifications of positive instances.

Precision

refers to the metric that measures how many of the predicted outputs were predicted correctly, calculated using the equation:

$$\text{Precision} = \text{True positives} / \ (\text{True positives} + \text{False positives})$$

Recall

is the percentage of correct positive predictions that have been made, based on all the correct positives in the dataset. It is calculated using the equation:

$$\text{Recall} = \text{True positives} / \ (\text{True positives} + \text{False negatives})$$

F1-Score

is defined as the harmonic mean of both recall and precision of a model, scaled appropriately, calculated using the equation:

$$\text{F1 Score} = 2 * \ (\text{Precision} * \text{Recall}) \ / \ (\text{Precision} + \text{Recall})$$

The ROC AUC

can also be defined in terms of precision and recall, which are two other common metrics used in binary classification. Precision measures the fraction of positive predictions that are correct, while recall measures the fraction of positive examples that are correctly identified by the model. The relationship between precision, recall, and the true positive rate (TPR) and false positive rate (FPR) is as follows:

$$\mathrm{TPR} = \mathrm{recall} = \mathrm{true\ positives} / \ (\mathrm{true\ positives} + \mathrm{false\ negatives})$$
$$\mathrm{FPR} = \mathrm{false\ positives} / \ (\mathrm{false\ positives} + \mathrm{true\ negatives})$$

The ROC curve can be constructed by varying the classification threshold of the model and plotting the TPR against the FPR. The ROC AUC is then calculated by integrating the ROC curve as follows:

$$\mathrm{ROC\ AUC} = \mathrm{integral}\ (\mathrm{recall}\ (\mathrm{FPR}))\ \ \mathrm{dFPR}$$

where recall(FPR) is the recall as a function of the false positive rate, and the integral is taken over the range of FPR. The ROC AUC can be interpreted as the probability that a randomly chosen positive example is ranked higher than a randomly chosen negative example by the model's predicted scores, when the threshold for positive classification is varied. A perfect classifier has an ROC AUC of 1, while a random classifier has an ROC AUC of 0.5.

Computation time

is the learning time of the model in the DNA classification model using different Machine Learning Algorithms, as well as the time spent in the model testing process. The time function is used to record the time it takes to train the data during the classification process for all the methods used in the model. The computation time for each algorithm is shown in Table 2.

The experimental results

Table 3 presents the results of the DNA sequence classification algorithms, along with their time complexity. The table includes seven algorithms: K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Naive Bayes, Support Vector Machine with Radial Basis Function kernel (SVM RBF), Support Vector Machine with Sigmoid kernel (SVM Sigmoid), and Support Vector Machine with Linear kernel (SVM Linear).

The performance of the algorithms is evaluated using several metrics, including accuracy, precision, ROC_AUC, recall, and F1 score. The execution time of each algorithm is also reported.

Table 3 shows that SVM Linear has the highest accuracy (0.963) and F1 score (0.97), indicating that it has the best overall performance among the algorithms. Naive Bayes also performs well with an accuracy of 0.838 and an F1 score of 0.94. KNN has the lowest accuracy (0.778) and F1 score (0.79) among the algorithms, while Random Forest performs poorly with an accuracy of 0.609 and an F1 score of 0.55. In terms of time

Table 3 Results of the DNA sequence Classification Algorithms and Time Complexity

	Algorithms	Accuracy	Precision	ROC_AUC	Recall	F1 score	Execution time
1	KNN	0.778	0.62	0.701	0.65	0.79	14.448
2	Decision Tree	0.815	0.92	0.891	0.71	0.8	13.271
3	Random Forest	0.609	0.67	0.623	0.47	0.55	12.983
4	Naive Bayes	0.838	0.83	0.855	0.88	0.94	12.606
5	SVM RBF	0.925	0.83	0.937	0.88	0.94	13.173
6	SVM Sigmoid	0.925	0.83	0.952	0.88	0.94	14.189
7	SVM Linear	0.963	0.91	0.963	0.94	0.97	10.059

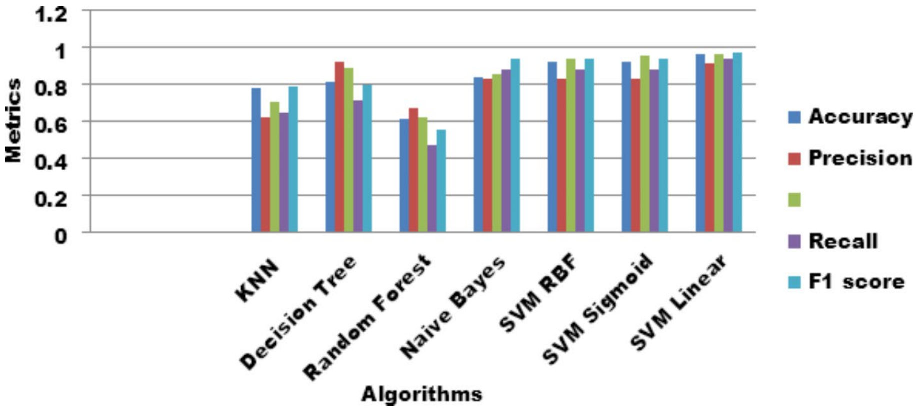


Fig. 8 Metrics of each algorithm on the same DNA sequence

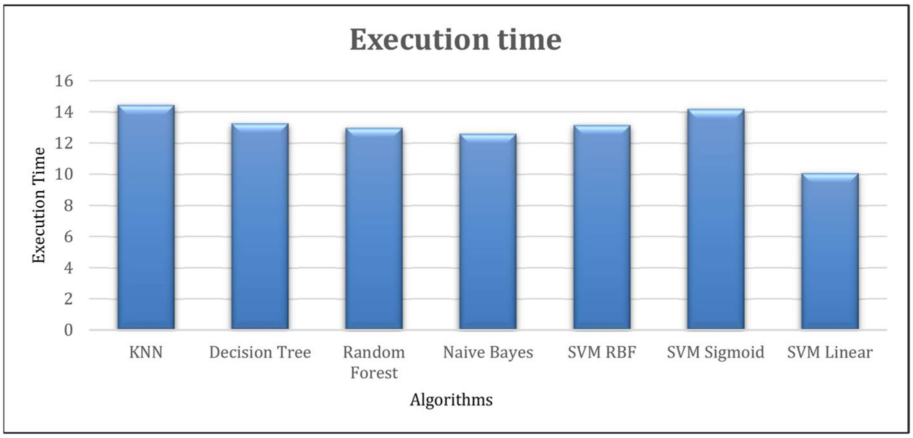


Fig. 9 DNA sequence execution time for each technique

complexity, SVM Linear has the lowest execution time (10.059 seconds), followed by Decision Tree (13.271 seconds) and SVM RBF (13.173 seconds). KNN has the highest execution time (14.448 seconds), while SVM Sigmoid has the second-highest execution time (14.189 seconds).

Overall, the results suggest that SVM Linear and Naive Bayes are the top-performing algorithms for DNA sequence classification, while KNN and Random Forest are less effective. The time complexity results indicate that SVM Linear, Decision Tree, and SVM RBF are the most efficient algorithms in terms of execution time.

Figure 8 displays the metrics of each algorithm on the same DNA sequence, while Fig. 9 shows the execution time for each technique in the DNA sequence classification process.

Table 4 provides a comparison of the machine learning techniques discussed above with traditional techniques, including our two proposed methods (EFLPM and EPAPM) [47] based on their execution time for different pattern lengths in DNA sequences.

Table 4 summarizes the results of experiments conducted on different pattern matching and classification algorithms for different pattern lengths in DNA sequences. The table shows the pattern lengths in columns and the algorithms in rows, with corresponding values indicating execution time achieved by each algorithm for a particular pattern length.

Table 4 A comparison of machine learning techniques with traditional techniques based on their execution time for different pattern lengths in DNA sequences

No	Algorithms	Pattern Length				
		5	**10**	**15**	**20**	**25**
1	FLPM	0.0045	0.0025	0.0025	0.0026	0.0018
2	PAPM	0.0054	0.0018	0.0022	0.0025	0.0024
3	EFLPM	0.0035	0.0018	0.0019	0.0020	0.0012
4	EPAPM	0.0044	0.0011	0.0017	0.0013	0.0015
5	SVM Linear	0.0035	0.0020	0.0019	0.0015	0.0012

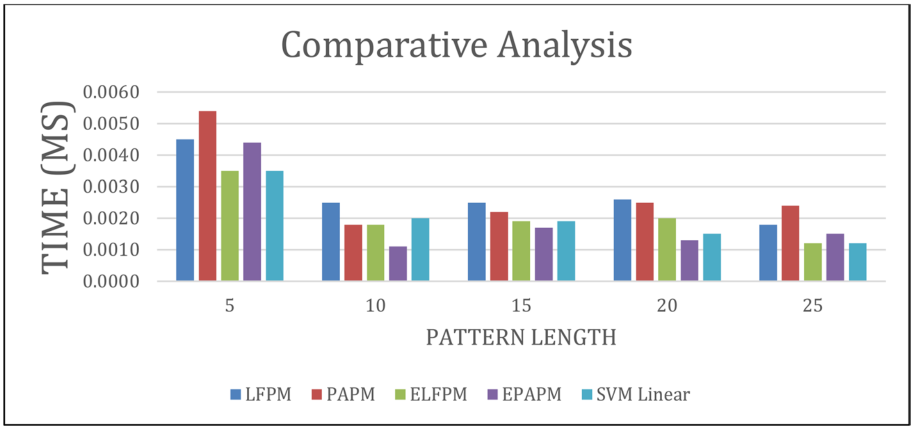


Fig. 10 Summary of the experiments results

The experiments involved the use of FLPM, PAPM, EFLPM, EPAPM, and SVM Linear algorithms for DNA sequence classification. At a pattern length of 5, FLPM achieved timeof 0.0045, PAPM achieved time of 0.0054, EFLPM achieved time of 0.0035, EPAPM achieved time of 0.0044, and SVM Linear achieved time of 0.0035.

As the pattern length increased, the accuracy of each algorithm varied. For instance, at a pattern length of 10, both EFLPM and EPAPM achieved the highest accuracy of 0.0018, while PAPM achieved time of 0.0018. SVM Linear performed the best at a pattern length of 25, achieving time of 0.0012.

In addition, the table indicates that our proposed methods, EFLPM and EPAPM, outperformed traditional techniques such as FLPM and PAPM in terms of classification accuracy for certain pattern lengths. Furthermore, SVM Linear consistently performed well across all pattern lengths.

Overall, the results suggest that the accuracy of each algorithm is dependent on the pattern length, and machine learning techniques, specifically SVM Linear and our proposed methods, are better suited for DNA sequence classification than traditional techniques like FLPM and PAPM.The visualization of the summary of experimental results is shown in Fig. 10.

As shown in Fig. 10, SVM Linear outperforms other methods in terms of execution time, indicating that our model is efficient in classifying DNA sequences and matching patterns. Moreover, we compared the proposed and existing systems based on their runtime, and the results demonstrate that both our proposed methods (EFLPM and EPAPM) and SVM Linear with a linear kernel have similar execution times, minimizing time complexity. Therefore, these methods are effective and practical solutions for DNA sequence classification tasks that require fast execution times.

Discussion

The use of machine learning algorithms in pattern matching has gained significant attention in recent years due to their ability to accurately classify and identify patterns in large datasets. In this discussion section, we will explore the advantages and limitations of using machine learning algorithms in pattern matching and their potential applications.

One of the main advantages of using machine learning algorithms in pattern matching is their ability to process large amounts of data quickly and accurately. Machine learning algorithms can identify complex patterns and relationships in data sets that may not be immediately apparent to human analysts. Furthermore, these algorithms can learn and adapt to new patterns as they are discovered, making them a powerful tool for identifying and classifying new patterns and trends.

Another advantage of using machine learning algorithms in pattern matching is their ability to automate the process, reducing the need for human intervention. This can significantly reduce the time and resources required to analyze large datasets, enabling researchers to focus on other aspects of their research.

However, there are also limitations to using machine learning algorithms in pattern matching. One of the main challenges is the need for large amounts of high-quality data to train the algorithms effectively. In many cases, obtaining high-quality data can be difficult, particularly when working with complex data sets such as DNA sequences.

Another limitation is the potential for overfitting, where the algorithm becomes too specialized in recognizing specific patterns in the training data and performs poorly when presented with new data. To address this challenge, researchers must carefully select and preprocess the data used to train the algorithms and use appropriate techniques such as cross-validation to evaluate their performance.

Despite these limitations, machine learning algorithms have many potential applications in pattern matching, including DNA sequence classification, image recognition, and natural language processing. For example, in DNA sequence classification, machine learning algorithms can be used to identify specific patterns associated with various diseases, enabling researchers to develop more targeted treatments.

Overall, the use of machine learning algorithms in pattern matching has the potential to revolutionize many fields and disciplines, enabling researchers to identify and analyze patterns in large datasets quickly and accurately. However, it is important to carefully evaluate the strengths and limitations of these algorithms to ensure they are used effectively and appropriately.

Limitations

The limitations for Pattern Matching classification can be summarized as follows:

- **Algorithm comparison**: The study compares the proposed model with only two other algorithms, FLPM and PAPM. While the results show that the proposed model outperforms these algorithms, it would be valuable to compare the Deep Learning models with a wider range of algorithms to further validate its effectiveness.
- **Pattern length evaluation**: The study examines the impact of pattern length on the accuracy and time complexity of each algorithm, but only for a limited range of pattern lengths. It would be valuable to investigate the performance of the algorithms for longer or more complex patterns.

- **Feature extraction**: More complex feature extraction methods could potentially improve the model's performance.
- **Scope of applications**: The study focuses primarily on DNA sequence classification for drug discovery, personalized medicine, and disease diagnosis. While these are important applications, the model's potential for other applications or fields is not explored in depth.
- **Imbalanced dataset**: The dataset used in the study may be imbalanced, meaning that there are more examples of one class than the other. This could affect the model's performance and lead to biased results.
- **hyperparameter tuning**: The study uses a limited range of hyperparameters for each algorithm, which may not be optimal for all datasets or applications.

Conclusion and future work

The proposed model for DNA sequence classification offers a valuable contribution to the field and has significant potential for practical applications. Further research and development in this area could lead to improved accuracy and efficiency in DNA sequence classification, with important implications for drug discovery, personalized medicine, and disease diagnosis.

This paper focuses on using a pattern-matching method to retrieve matched DNA sequences. The study covers the following steps:

- Building a DNA Sequences dataset from DNA FASTA files and converting it to a CSV file.
- Importing data from the CSV file.
- Converting text inputs to numerical data.
- Building and training classification algorithms.
- Comparing and contrasting classification algorithms based on execution time, recall, precision, F1-score, ROC_AUC, occurrences, and accuracy.

The performance of the proposed model is evaluated using various machine learning algorithms, and the results indicate that the SVM linear classifier achieves the highest accuracy and F1 score among the tested algorithms. This finding suggests that the proposed model can provide better overall performance than other algorithms in DNA sequence classification. In addition, the proposed model is compared to two suggested algorithms, namely FLPM and PAPM, and the results show that the proposed model outperforms these algorithms in terms of accuracy and efficiency. The study further explores the impact of pattern length on the accuracy and time complexity of each algorithm. The results show that as the pattern length increases, the execution time of each algorithm varies. For a pattern length of 5, SVM Linear and EFLPM have the lowest execution time of 0.0035 s. However, at a pattern length of 25, SVM Linear has the lowest execution time of 0.0012 s. The experimental results of the proposed model show that SVM Linear has the highest accuracy and F1 score among the tested algorithms. SVM Linear achieved an accuracy of 0.963 and an F1 score of 0.97, indicating that it can provide the best overall performance in DNA sequence classification. Naive Bayes also performs well with an accuracy of 0.838 and an F1 score of 0.94.

Future work The proposed model for DNA sequence classification is a promising development that can enhance the accuracy and efficiency of DNA sequence classification. However, there are several future directions that could be pursued to further improve the model's performance and expand its potential applications.

One possible future direction is to investigate the performance of the proposed model on larger datasets. The current study used a relatively small dataset, and it would be interesting to see how the model performs on larger-scale datasets with more diverse sequences. This would help to validate the model's effectiveness in real-world scenarios and enhance its potential applications.

Another possible future direction is to explore the use of deep learning techniques for DNA sequence classification. Deep learning models, such as convolutional neural networks (CNNs,MLP and LSTM) and recurrent neural networks (RNNs), have shown promising results in various domains, including natural language processing and computer vision. It would be interesting to see how these techniques could be adapted to DNA sequence classification and whether they could provide improved performance compared to the proposed model.

Furthermore, it would be valuable to investigate the model's performance on different types of DNA sequences, such as those from different organisms or with different functional roles. This would help to identify any potential limitations of the model and areas where it could be further improved.

Acknowledgements
Authors sincerely acknowledge Computer Science Department in Faculty of Science, Minia University for the facilities and support.

Authors' contributions
This work was carried out in collaboration among all authors. Author BAH designed the study, performed the statistical analysis and wrote the protocol. Authors OASI and TAEH managed the analyses of the study, managed the literature searches and wrote the first draft of the manuscript. All authors read and approved the final manuscript.

Funding
Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Data availability
The data that support the findings of this study are available in https://github.com/belalahmedhamed/Bioinformatics-Dataset. These data were derived from the following resources available in the public domain of "The National Center for Biotechnology Information advances science and health by providing access to biomedical and genomic information" https://www.ncbi.nlm.nih.gov/guide/dna-rna/.

Declarations

Competing interests
The authors declare that there is no conflict of interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical statement
All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Consent statement
Informed consent was obtained from all individual participants included in the study.

Received: 30 March 2023 / Accepted: 16 July 2023
Published online: 25 July 2023

References

1. Marczyk VR, Recamonde-Mendoza M, Maia AL, Goemann IMJT. *Classification of Thyroid Tumors Based on DNA Methylation Patterns* 2023(ja).

2. Liu PJFiG. Pan-cancer DNA methylation analysis and tumor origin identification of carcinoma of unknown primary site based on multi-omics. 2022;12:798748.
3. Zhao F, Li L, Lin P, Chen Y, Xing S, Du H, Wang Z, Yang J, Huan T, Long C, Zhang L, Wang B, Fang M. HExpPredict: In Vivo Exposure Prediction of Human Blood Exposome Using a Random Forest Model and Its Application in Chemical Risk Prioritization. 2023;131(3):037009.
4. Suyama Y, Hirota SK, Matsuo A, Tsunamoto Y, Mitsuyuki C, Shimura A, Okano K. Complementary combination of multiplex high-throughput DNA sequencing for molecular phylogeny. Wiley Online Library; 2022.
5. Zhong H-S, Dong M-J, F.J.I.S.C.L S, Gao. *G4Bank: A database of experimentally identified DNA G-quadruplex sequences* 2023: p. 1–9.
6. Touati R, Messaoudi I, Oueslati AE, Lachiri Z, Kharrat M. New Intraclass Helitrons classification using DNA-Image sequences and machine learning approaches. IRBM. 2021;42(3):154–64.
7. Norlin S. "DNA Seq Classif Using Variable Length Markov Models" 2020.
8. Ryu C, Lecroq T, Park K. Fast string matching for DNA sequences. Theor Comput Sci. 2020;812:137–48.
9. Xu G, Li H, Ren H, Lin X, X.J.I.T.o.C C, Shen. DNA similarity search with access control over encrypted cloud data. 2020;10(2):1233–52.
10. Yang A, Zhang W, Wang J, Yang K, Han Y. L.J.F.i.B. Zhang, and Biotechnology. Rev application Mach Learn algorithms Seq data Min DNA. 2020;8:1032.
11. Ravikumar M, Prashanth MJC, Cognition. and M.L.A.P.o. ICCCMLA, *Analysis of DNA sequence pattern matching: a brief survey* 2021: p. 221–229.
12. Millán Arias P, Alipour F, Hill KA, Kari LJPo. DeLUCS: Deep learning for unsupervised clustering of DNA sequences. 2022;17(1):e0261531.
13. Rossi F, Paiardini AJCB. *A machine learning perspective on DNA and RNA G-quadruplexes*. 2022. 17(4): p. 305–9.
14. Xiong X, Zhu T, Zhu Y, Cao M, Xiao J, Li L, Wang F, Fan C, Pei HJNMI. Mol convolutional neural networks DNA Regul circuits. 2022;4(7):625–35.
15. Ibrahim OAS, Hamed BA, El-Hafeez TAbd. *A new fast technique for pattern matching in biological sequences* 2022: p. 1–22.
16. Jukic S, Saracevic M, Subasi A, Kevric JJM. *Comparison of ensemble machine learning methods for automated classification of focal and non-focal epileptic EEG signals*. 2020. 8(9): p. 1481.
17. Hassan SU, Ahamed J, Ahmad KJSO, Computers. Analytics of machine learning-based algorithms for text classification. 2022;3:238–48.
18. Kurani A, Doshi P, Vakharia A, J.A.o.D M. *A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting*. 2023. 10(1): p. 183–208.
19. Mondal HS, Ahmed KA, Birbilis N, Hossain MZJSR. Mach Learn detecting DNA attachment SPR Biosens. 2023;13(1):3742.
20. Alshayeji MH, S.C.J.E.S.w.A., Sindhu. Viral genome prediction from raw human DNA sequence samples by combining natural language processing and machine learning techniques. 2023;218:119641.
21. Sarkar S, Mridha K, Ghosh A, Shaw RN. *Machine Learning in Bioinformatics: New Technique for DNA Sequencing Classification*, in *Advanced Computing and Intelligent Technologies: Proceedings of ICACIT 2022*. 2022, Springer. p. 335–355.
22. Karr AF, Bowen Z. and A.A.J.a.p.a. Porter, *Structure of Classifier Boundaries: Case Study for a Naive Bayes Classifier* 2022.
23. Habib MA, Manik MMH, Khulna B. *Classification of DNA Sequence Using Machine Learning Techniques*. 2022, EasyChair.
24. Khatun ME, Rabeya T. *A Machine Learning Approach for Sentiment Analysis of Book Reviews in Bangla Language*. in 2022 *6th International Conference on Trends in Electronics and Informatics (ICOEI)*. 2022. IEEE.
25. Kushwah JS, Kumar A, Patel S, Soni R, Gawande A, Gupta SJMTP. Comp study regressor classifier Decis tree using Mod tools. 2022;56:3571–6.
26. Rivera-Lopez R, Canul-Reich J, Mezura-Montes E, Cruz-Chávez MAJS, Computation E. Induction of decision trees as classification models through metaheuristics. 2022;69:101006.
27. Costa VG, C.E.J.A.I R, Pedreira. *Recent advances in decision trees: An updated survey* 2023. 56(5): p. 4765–4800.
28. Lee CS, Cheang PYS, J.A.i.D M. Predictive analytics in business analytics: decision tree. 2022;26(1):1–29.
29. Bansal M, Goyal A, A.J.D.A.J., Choudhary. *A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning*. 2022. 3: p. 100071.
30. Shorabeh SN, Samany NN, Minaei F, Firozjaei HK, Homaee M, Boloorani ADJRE. Decis model based Decis tree Part swarm Optim algorithms identify optimal locations solar power plants Constr Iran. 2022;187:56–67.
31. Ravikumar M, Prashanth M, Guru D. Matching pattern in DNA sequences using machine learning Approach based on K-Mer function, Modern approaches in machine Learning & Cognitive Science: a Walkthrough. 2022, Springer. 159–71.
32. Koul N, Manvi SS, Gardiner B. *Method for Classification of Cancers with Partial Least Squares Regression as Feature Selector with Kernel SVM*. in *2022 International Conference for Advancement in Technology (ICONAT)*. 2022. IEEE.
33. Manoharan A, Begam K, Aparow VR, J.J.o.E D. *Artificial neural networks, gradient boosting and support Vector Machines for electric vehicle battery state estimation: a review*. 2022. 55: p. 105384.
34. Zhang H, Zou Q, Ju Y, Song C, Chen DJCB. *Distance-based support vector machine to predict DNA N6-methyladenine modification*. 2022. 17(5): p. 473–82.
35. Roy A, Chakraborty SJRE, Safety S. *Support vector machine in structural reliability analysis: A review* 2023: p. 109126.
36. Jäger J, Krems RVJNC. *Universal expressiveness of variational quantum classifiers and quantum kernels for support vector machines*. 2023. 14(1): p. 576.
37. Dragomir MP, Calina TG, Perez E, Schallenberg S, Chen M, Albrecht T, Koch I, Wolkenstein P, Goeppert B, Roessler SJE. DNA methylation-based classifier differentiates intrahepatic pancreato-biliary tumours 2023. 93.
38. Chadha A, Dara R, Pearl DL, Sharif S, Poljak ZJPVM. *Predictive analysis for pathogenicity classification of H5Nx avian influenza strains using machine learning techniques* 2023. 216: p. 105924.
39. Mangkunegara IS, Purwono P. *Analysis of DNA Sequence Classification Using SVM Model with Hyperparameter Tuning Grid Search CV*. in 2022 *IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*. 2022. IEEE.
40. Andrade-Girón D, Carreño-Cisneros E, Mejía-Dominguez C, Velásquez-Gamarra J, Marín-Rodriguez W, Villarreal-Torres H. R.J.E.E.T.o.P.H. Meleán-Romero, and Technology, *support vector machine with optimized parameters for the classification of patients with COVID-19*. 2023. 9: p. e8–e8.
41. NCBI., *National Center for Biotechnology Information* 2020.
42. Borjigin C. *Data analysis with Python*, in *Python Data Science*. Springer; 2023. pp. 295–342.

43. Rajamani SK, Iyer RS. *Machine Learning-Based Mobile Applications Using Python and Scikit-Learn*, in *Designing and Developing Innovative Mobile Applications*. 2023, IGI Global. p. 282–306.
44. Lavanya A, Gaurav L, Sindhuja S, Seam H, Joydeep M, Uppalapati V, Ali W. Assessing the performance of Python Data visualization libraries: a review. and V.S. SD; 2023.
45. Valero-Carreras D, Alcaraz J, Landete MJC, Research O. Comparing two SVM models through different metrics based on the confusion matrix. 2023;152:106131.
46. Li J, Sun H, Li JJML. *Beyond confusion matrix: learning from multiple annotators with awareness of instance features*. 2023. 112(3): p. 1053–75.
47. Ibrahim OAS, Hamed BA, El-Hafeez TAbd. A new fast technique for pattern matching in biological sequences. 2023;79(1):367–88.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.