Electronic Supplementary Material (ESI) for Digital Discovery.

Supplementary Information

Artificial Intelligence Aided Recognition and Classification of DNA Nucleotides Using MoS_2 Nanochannel

Sneha Mittal,[†] Souvik Manna,[†] Milan Kumar Jena,[†] Biswarup Pathak*,[†]

[†]Department of Chemistry, Indian Institute of Technology (IIT) Indore, Indore, Madhya Pradesh, 453552, India
*E-mail: biswarup@iiti.ac.in

Contents Pages

1. ML aided DNA recognition S2
2. Hyperparameter tuning of ML regression models S9
3. Stability check of best-fitted XGBR model S10
4. RDKit fingerprints eliminated XGBR prediction S11
5. Pearson's Correlation Matrix S12
6. RDKit fingerprints eliminated feature importance plot S13
7. Ensuring Stability of XGBR models S14
8. ML aided DNA classification S15
9. Hyperparameter tuning of classification models S20
10. Classification Reports S21
11. Stability Check of Best-fitted LR Classification Algorithm S22
12. Single nucleotide identification for rotation dynamics S23
13. Transmission and Current-Voltage Plots with underlying physics S24
14. Adsorption energy and translocation time S26
15. Charge density difference Plots S27
16. Effect of in-plane rotation on transmission function S28

1. ML Aided DNA Recognition:

ML Details

Text S1:

In combination with RDKit fingerprints generated from SMILES string, 8 key input features derived from the molecular, chemical, and electronic properties of the DNA nucleotides have been considered in the input training dataset. A total of six input features, including average atomic radius, average ionic radius, average covalent radius, average Pauling electronegativity, average number of valance electrons, and average polarizability were manually extracted utilizing composition-based feature vectors ("CBFVs")[1] and two features HOMO and LUMO were extracted from the gaussian optimized geometry of DNA nucleotides.

Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique for analyzing large datasets that have a high number of dimensions or features per observation.[2,3] The primary goal of PCA is to reduce the dimensionality of the data while retaining the maximum amount of information and making the data more interpretable. Dimensionality reduction is achieved by transforming the original dataset into a new coordinate system, also known as principal components, where the majority of the variation in the data can be described using fewer dimensions than the original dataset. The transformation is linear and is performed by computing the eigenvectors and eigenvalues of the covariance matrix of the data.

XGBoost Regression (XGBR)

In 2016, Tianqi Chen and his co-workers introduced a machine learning algorithm called XGBoost (eXtreme Gradient Boosting) in their paper titled "XGBoost: A Scalable Tree Boosting System".[4] XGBoost is a type of ensemble learning algorithm that utilizes multiple tree learners to enhance prediction accuracy using the principle of gradient boosting. This approach involves combining a weak model with several other weak models (decision trees) to create a more powerful and accurate model.[5] XGBoost has gained widespread popularity due to its efficiency, scalability, and ability to handle a variety of data types.

Random Forest Regression (RFR)

The random forest ML algorithm was first introduced by Leo Breiman and his colleagues in 2001.[6] The algorithm utilizes an ensemble learning approach that combines multiple decision trees, using a technique known as bootstrap and aggregation (bagging), to generate a powerful predictive model. Rather than relying on a single decision tree, the random forest algorithm aggregates the predictions of many decision trees to make a final prediction. The final output is calculated as the average of the individual predictions made by each decision tree. By using different subsets of data and features to train each decision tree, the random forest helps to reduce overfitting and improve the generalization performance of the model.

Extra Tree Regression (ETR)

The extra tree ML algorithm was first introduced by Pierre Geurts et al. in 2006 in their paper titled "Extremely randomized trees."[7] It is an ensemble learning method that combines multiple decision trees to make a more accurate and stable prediction. The basic idea behind ETR is to create a large number of decision trees, each trained on a random subset of the training data and using a random subset of the available features at each split. The final prediction is then made by aggregating the predictions of all the individual trees.

Light Gradient Boosted Machine Regressor (LGBR):

The light gradient boosted machine regressor (LGBR) algorithm was first introduced by Guolin Ke et al. in 2017 in the article entitled "LightGBM: A Highly Efficient Gradient Boosting Decision Tree."[8] The LGBR algorithm uses a gradient boosting framework that builds an ensemble of decision trees, where each tree is trained to correct the errors made by the previous trees. LGBMR is optimized for speed and efficiency, and it uses two novel optimization techniques: gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) to improve its performance. The model can handle categorical variables, unlike other algorithms, which could be advantageous in avoiding overfitting.

Mean Absolute Error (MAE):

Mean Absolute Error (MAE) is a commonly used performance metric in ML to measure the average absolute difference between the predicted values and the actual values. It provides a straightforward measure of the model's accuracy by quantifying the average magnitude of errors. MAE is defined as,

$$\text{MAE} = \frac{1}{N}\sum_{i}^{n}|Y_i - y_i|$$

Here Y_i and y_i denotes the DFT calculated, and predicted transmission function, respectively, and N represents the total number of data points.

Coefficient of Determination (R^2):

The coefficient of determination (R^2) is a statistical measure of how well a regression model performs in predicting the output variable on the test dataset. It quantifies the proportion of the variance in the actual output values that is captured by the predicted values from the model. The coefficient of determination normally ranges from 0 to 1, where 0 indicates that the model fails to capture any meaningful relationship between the input features and the target, and 1 indicates that the model perfectly predicts the output variable on the test data, capturing all the variance in the actual values. R^2 is defined as,

$$R^2 = \frac{\sum_{i}^{n} (Y_i - y_i)^2}{\sum_{i}^{n} (Y_i - \bar{y}_i)^2}$$

Here Y_i and y_i denotes the DFT determined and predicted transmission function, respectively; n represents the total number of transmission data points, and $\bar{y}_i$ represents an average of transmission values.

10-fold cross-validation:

The cross-validation method is a statistical method of ML model validation. The method assesses how well a trained model will generalize to an independent dataset. In the 10-fold cross-validation, the input dataset is randomly partitioned into 10 equal parts or 'folds.' The data of 9 of these folds is used for the training of the model, and the data of the remaining fold is used for validation of the model. The process is repeated 10 times so that each group can be utilized as validation data. The results of the 10 evaluations are then averaged to produce a single performance metric, which is used to assess the model's performance.

Population Stability Index (PSI)

The population stability index (PSI) is a statistic that measures how much a variable has shifted over time and is used to monitor applicability of a statistical model to the current population. A PSI value below 0.1 implies a stable model with no significant change in the population distribution. A PSI value between 0.1 and 0.2 suggests moderate changes, requiring cautious consideration before retraining the model. When the PSI value exceeds 0.2, it indicates a significant change in the population distribution, highlighting an unstable model.

Learning Curve:

The learning curve is a visualization technique that demonstrates the impact of increasing training data on the model's performance toward output prediction. Learning curves help in understanding the trade-off between bias and variance. Bias refers to the error introduced when a simplified model is used to approximate a complex problem. Variance, on the other hand, refers to the model's sensitivity to fluctuations in the training data. By observing the convergence of training and testing performance, one can check whether the model is stable or not. If the scores exhibit high variance or large fluctuations, it suggests an unstable model that is highly sensitive to changes in the training data.

Most-stable Configuration

Text S2:

To determine the most-stable configuration of DNA nucleotides, we first relaxed the atomic structures of DNA nucleotides (dAMP, dGMP, dCMP, dTMP) by utilizing the B3LYP/6-31+G* level of theory, as available in the Gaussian09 code.[9] Subsequently, we focused on obtaining the most stable geometry of DNA nucleotides adsorbed on the MoS_2 nanochannel surface. To get the most stable configuration of MoS_2 nanochannel+nucleotide systems, we have considered all possible rotations from 0° to 180° around the x-axis in the yz-plane for all four DNA nucleotides, as shown in **Figure S1.**

Scheme S1: Rotation of dAMP over the MoS_2 nanochannel: We have considered all possible rotations from 0° to 180° (in the steps of 30°) around the x-axis in the yz-plane for each individual DNA nucleotide adsorbed on MoS_2 nanochannel surface, as shown in **Figure S1.**

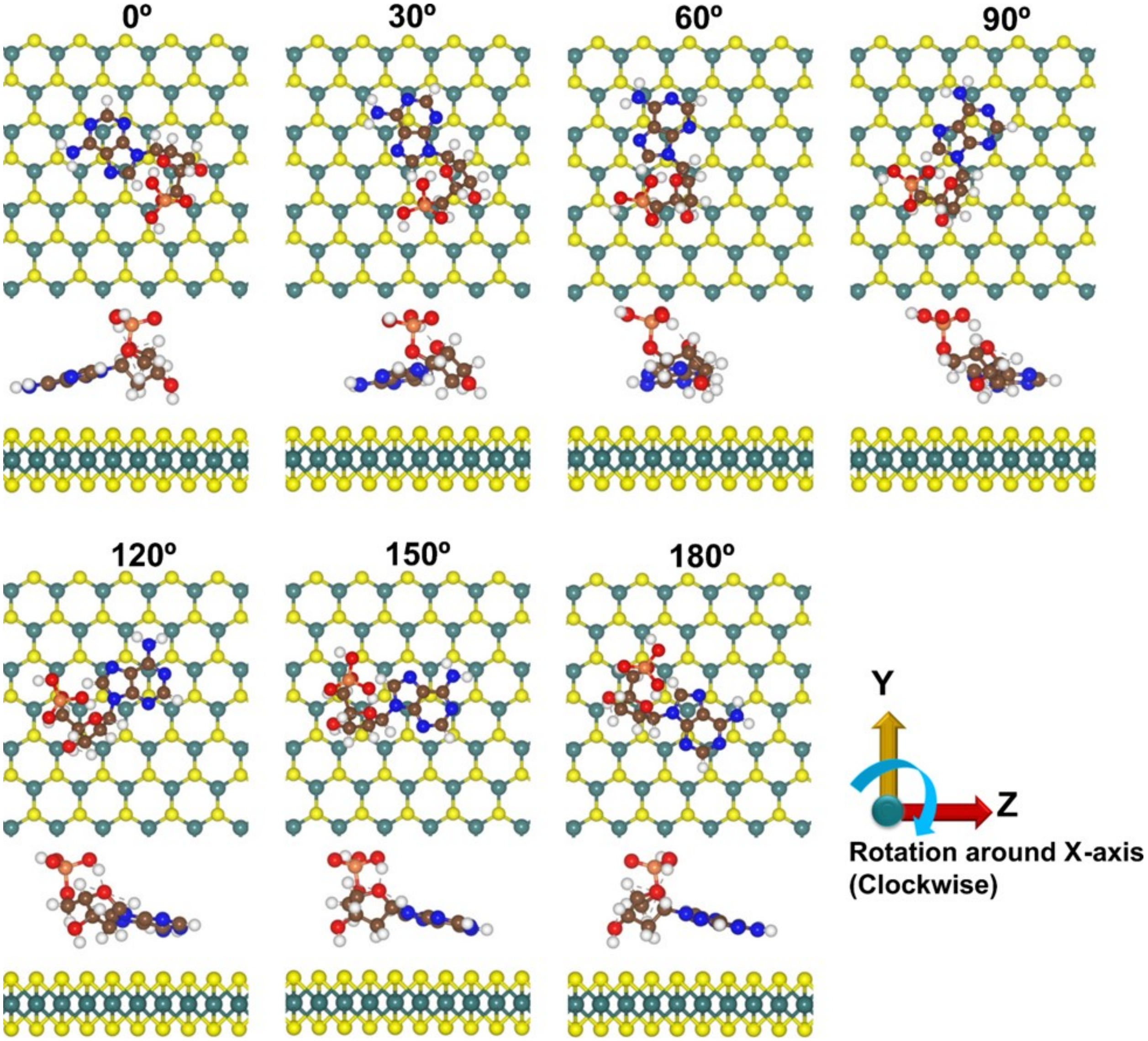


Figure S1: Representative orientations of dAMP (both top and side views) over the MoS_2 nanochannel are illustrated, corresponding to in-plane rotations from 0° to 180° in the steps of 30° around the x-axis in the yz-plane.

We have relaxed each individual nucleotide over the MoS_2 nanochannel surface in each considered orientation. For geometry relaxation, we perform density functional theory (DFT) computations by using the SIESTA (Spanish Initiative for Electronic Simulations with Thousands of Atoms) code.[10] GGA-PBE (generalized gradient approximation with Perdew-Burke-Ernzerhof) approximation,[11] 400 Ry mesh cut-off, Troullier-Martins norm-conserved pseudopotentials,[12] and conjugate gradient (CG) algorithm are the parameters used in the DFT-assisted geometry optimization.

Table S1. Relative energies (in eV) of the MoS_2 nanochannel+nucleotide systems when DNA nucleotides (dAMP, dGMP, dCMP, dTMP) are adsorbed on the nanochannel surface in seven different orientations (0°, 30°, 60°, 90°, 120°, 150°, 180°) as shown in **Figure S1**.

Nucleotides	**0^0**	**30^0**	**60^0**	**90^0**	**120^0**	**150^0**	**180^0**
dAMP	0.04	**0.00**	0.01	0.01	0.03	0.02	0.04
dGMP	0.02	0.01	0.10	0.05	0.01	**0.00**	0.02
dCMP	0.03	0.03	0.02	0.01	0.03	**0.00**	0.01
dTMP	0.04	**0.00**	0.02	0.01	0.02	0.01	0.02

To this end, we have a total of seven different orientations for each DNA nucleotide. To determine the most stable configuration, we have computed the relative energy values, as given in **Table S1**. The most stable configuration of each MoS_2 nanochannel+nucleotide system is given in **Figure S2**.

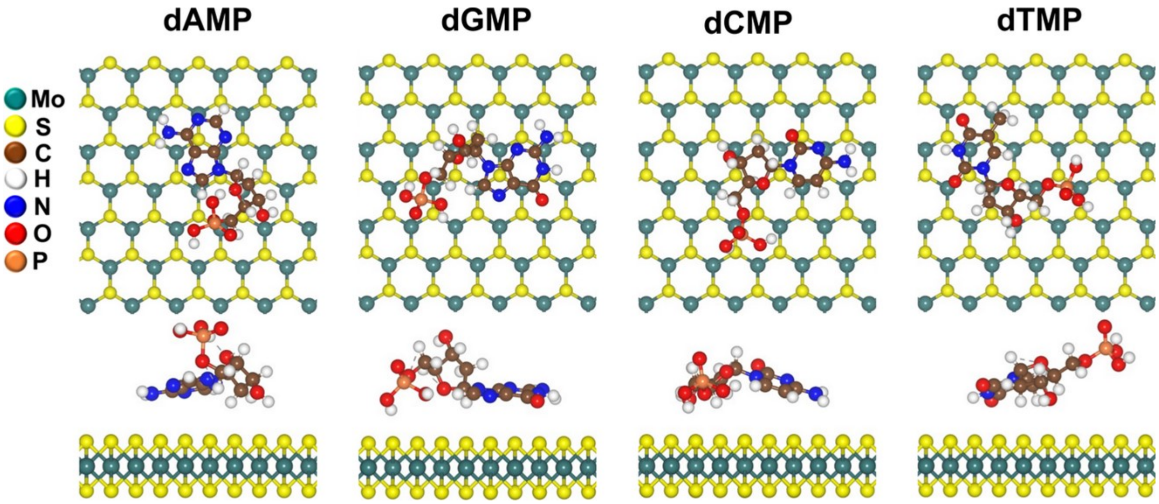


Figure S2. The most stable configurations (top and side views) of MoS_2 nanochannel+nucleotide systems.

2. Details of Optimized Hyperparameters and Test Mean Absolute Error (MAE):

Table S2. Tuned hyperparameters with their corresponding test MAE values.

S. No.	ML Regression Models	Optimized Hyperparameters	Test MAE
1.	XGBoost Regressor (XGBR)	verbose= 5, n_estimators=2500, min_child_weight=1, max_depth= 4,learning_rate= 0.15,cv=2,booster= 'gbtre', base_score= 0.5	0.15
2.	Random Forest Regressor (RFR)	bootstrap: True, ccp_alpha: 0.0, criterion: squared_error, max_depth: None, max_leaf_nodes: None, max_samples: None, min_samples_split: 2, n_estimators: 100, n_jobs: None, random_state: 100, verbose: 0	0.25
3.	Extra Tree Regressor (ETR)	bootstrap: True, ccp_alpha: 0.0, criterion: squared_error, max_depth: None, max_leaf_nodes: None, max_samples: None, min_samples_split: 2, n_estimators: 100, n_jobs: None, random_state: 1, verbose: 0	0.38
4.	Light Gradient Boosted Machine Regressor (LGBR)	subsample= 1, reg_lambda= 10, reg_alpha= 0, num_leaves=10, n_estimators=577,min_child_samples= 20,max_depth= 10,learning_rate= 0.5, colsample_bytree= 0.5)	0.47

3. Stability Check of Best-Fitted XGBR Model:

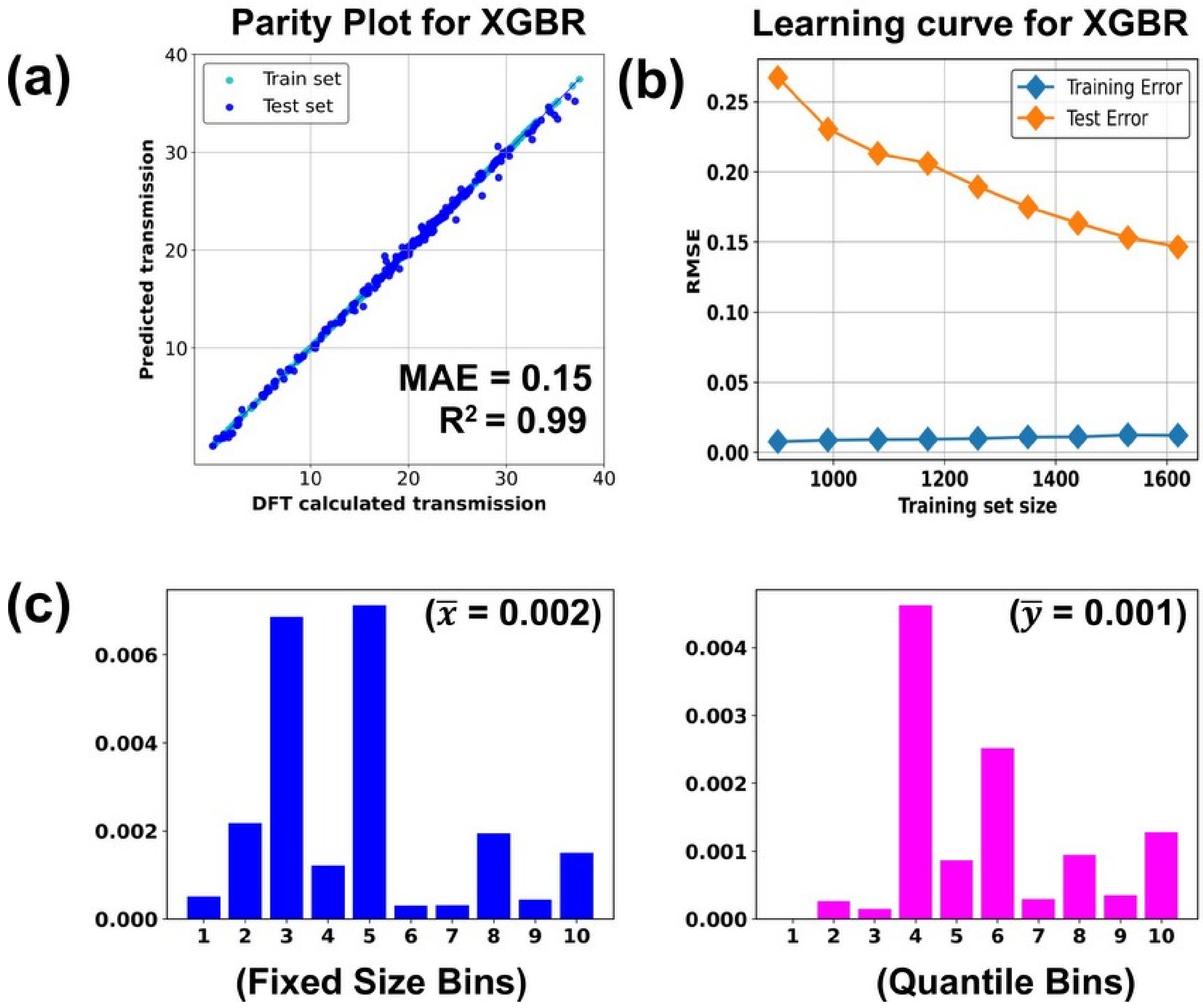


Figure S3. (a) Scatter plot of ML predicted vs. DFT calculated transmission function for train and test datasets with the best-fitted XGBR model, **(b)** learning curve for the best-fitted XGBR model. The considered training data size range is 0.50-0.90 (in the step of 0.05). For each considered train-test split, 10-fold cross-validation is performed, and the given MAE values are the mean of MAE of each fold cross-validation, and **(c)** population stability index (PSI) analysis with both fixed size bins and quantile bins for the best-fitted XGBR model. Here $\bar{x}$ and $\bar{y}$ are the mean PSI scores for fixed-size bins and quantile bins, respectively.

4. RDKit fingerprints eliminated XGBR prediction:

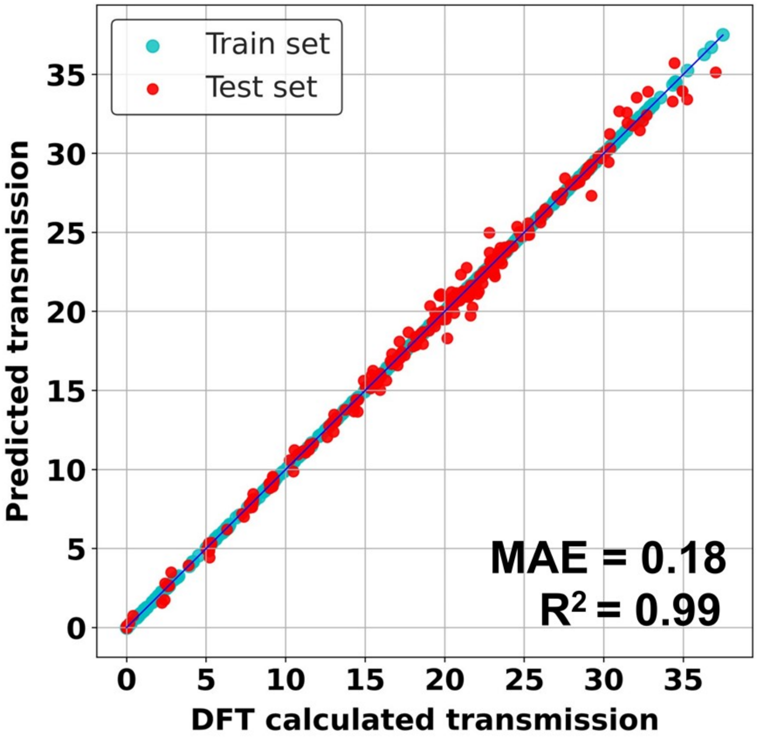


Figure S4. Scatter plot of ML predicted vs. DFT calculated transmission function for train and test datasets with XGBR model trained with RDKit fingerprints eliminated input dataset. The tight cluster of points aligned with the diagonal line signifies that the model consistently performs well on both the training and test datasets, indicating a good generalization capability.

5. Pearson's Correlation Matrix:

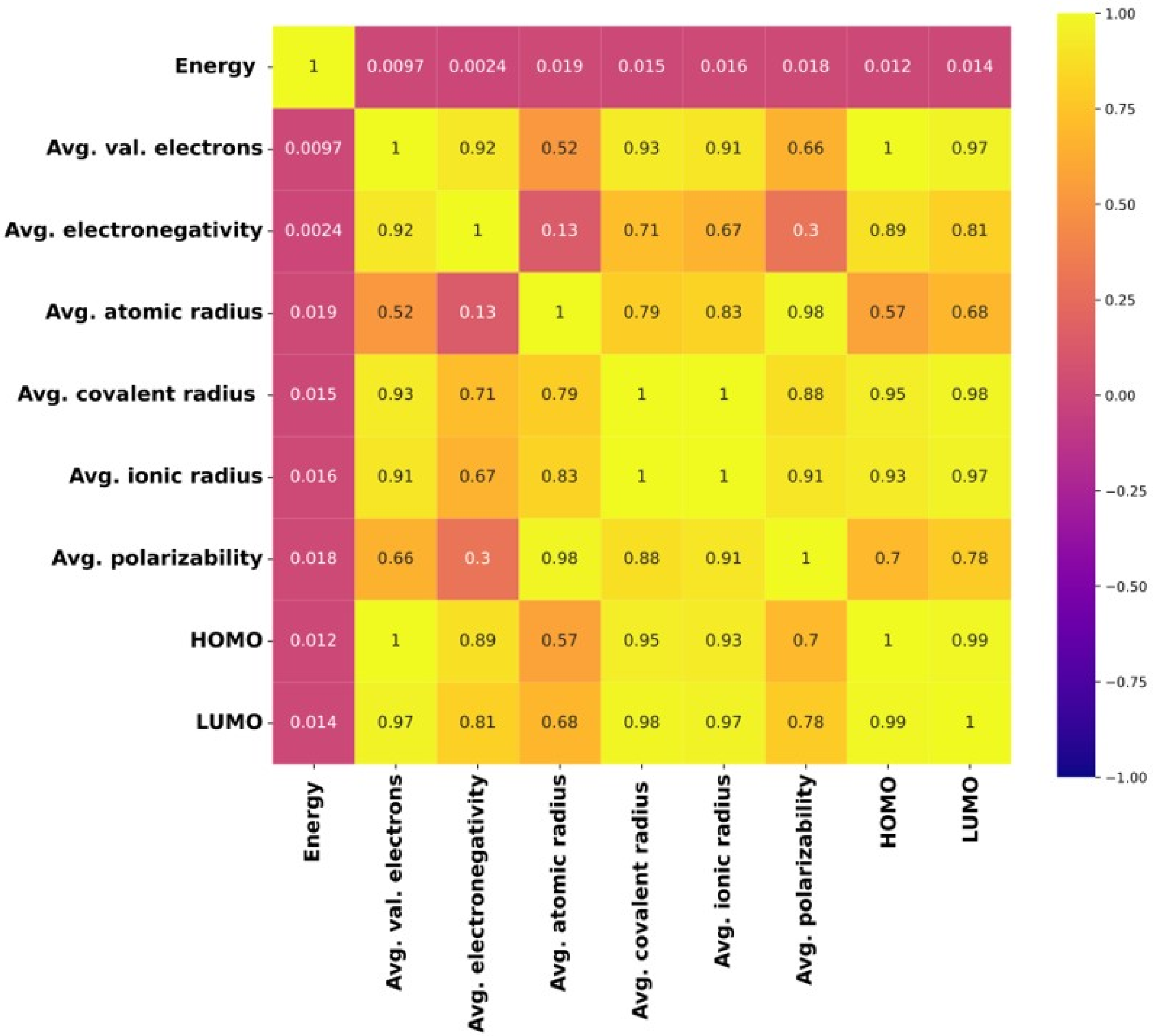


Figure S5. Pearson's correlation matrix illustrating the correlation among the RDKit fingerprints eliminated input features. The plot shows a high positive correlation among the features avg. covalent radius and avg. ionic radius as well as HOMO and LUMO.

6. RDKit fingerprints eliminated Feature Importance Plot:

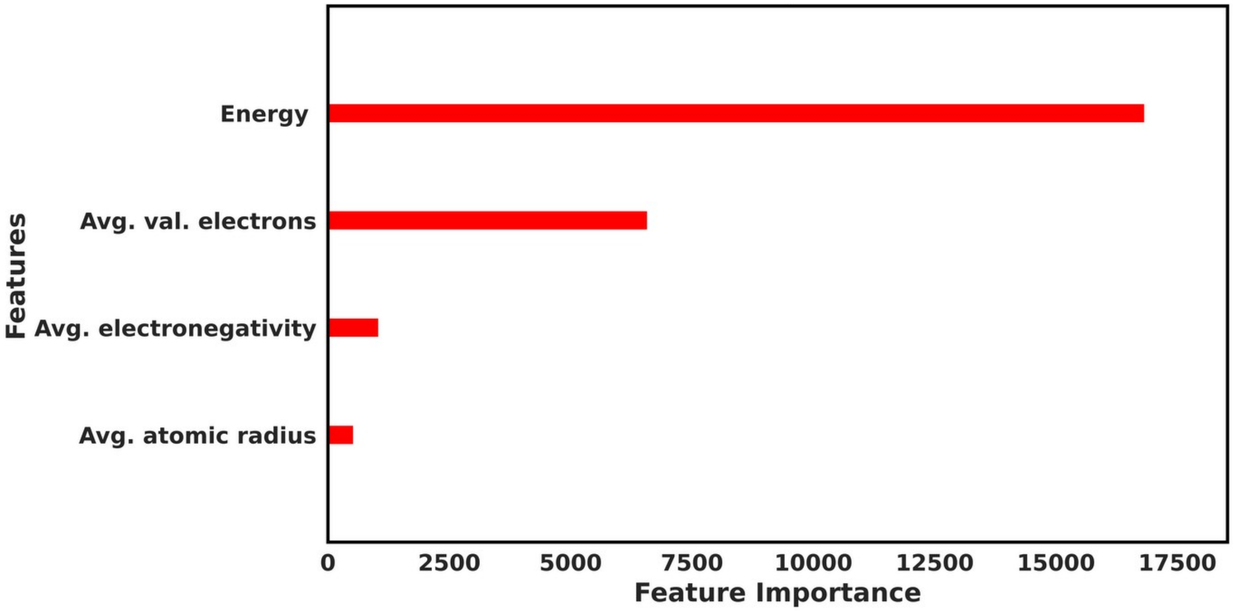


Figure S6. Feature importance plot of RDKit fingerprints eliminated input features for the best-fitted model XGBR toward prediction of transmission function. The plot shows the highest feature importance of the energy feature, which is obvious because it is the energy at which the transmission function is calculated.

7. Ensuring Stability of XGBR Models:

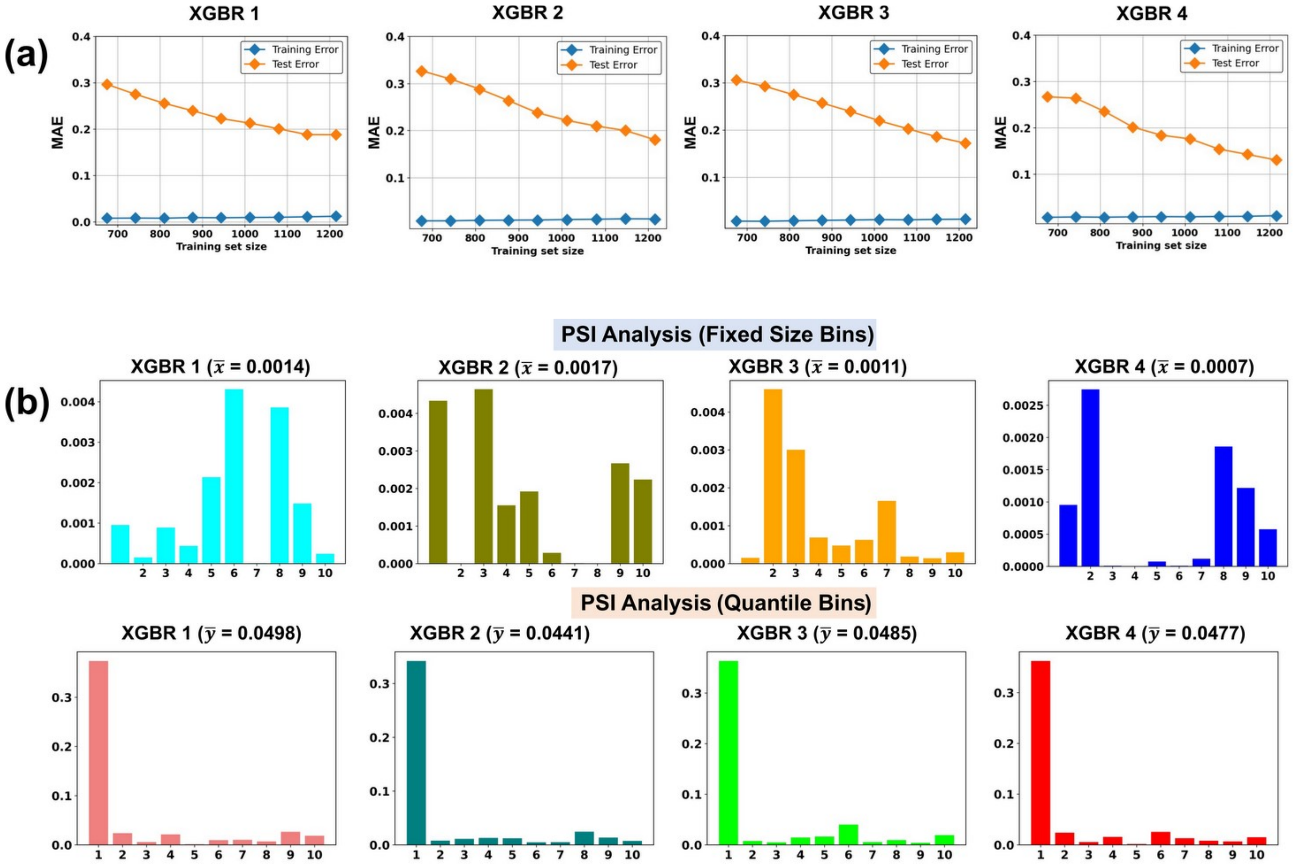


Figure S7. **(a)** Learning curves for the models **XGBR 1**, **XGBR 2**, **XGBR 3**, and **XGBR 4** used in the prediction of completely unknown nucleotides dAMP, dGMP, dCMP, and dTMP, respectively. The considered training data size range is 0.50-0.90 (in the step of 0.05). For each considered train-test split, 10-fold cross-validation is performed, and the given MAE values are the mean of MAE of each fold of 10-fold cross-validation and **(b)** The population stability index (PSI) analysis for the models **XGBR 1**, **XGBR 2**, **XGBR 3**, **XGBR 4**. Here $\bar{x}$ and $\bar{y}$ are the mean PSI of the models for fixed-size bins and quantile bins, respectively.

8. ML Aided DNA Classification:

Table S3. Features description used in the ML aided classification of DNA nucleotides.

Feature	Description
T	Transmission in the energy range of -2.5 to -1.7 eV
MAX	Maxima normalized transmission (T/Tmax); Tmax is the maximum value of transmission
MIN	Minima normalized transmission (T/Tmin); Tmin is the minimum value of transmission
AVG	Average normalized transmission (T/Tavg); Tavg is the average value of transmission

Classification Details

Text S3:

Logistic Regression (LR)

Logistic regression is a supervised machine learning algorithm mainly used for classification. It is a kind of statistical algorithm which analyze the relationship between a set of independent variables and the dependent variables.[13] In binary classification, the model uses a sigmoid function (also known as a logistic function) to map real-valued inputs to probabilities between 0 and 1. In multiclass classification, a softmax function is used to map inputs to probabilities across multiple classes. This allows the algorithm to effectively categorize data based on the given set of independent variables.

Random Forest Classification (RFC)

Random forest classification is an ensemble learning algorithm that combines multiple decision trees to make predictions.[6] The algorithm works by constructing a multitude of decision trees during the training phase. Each tree is trained on a different subset of the training data, randomly sampled with replacement (known as bootstrapping). Additionally, at each node of the tree, only a random subset of features is considered for splitting. During prediction, each tree in the random forest independently generates a prediction, and the final prediction is determined through a majority voting process. The class label that receives the most votes across all trees is chosen as the predicted class.

Decision Tree Classification (DTC)

Decision tree classification is a supervised learning classification algorithm that constructs a tree-like structure resembling a flowchart, where each internal node represents a test on a specific attribute, each branch represents a possible outcome of the test, and each leaf node holds a class label or a predicted value.[14] The decision tree is built iteratively by recursively partitioning the training data into subsets based on attribute values. This partitioning process continues until a termination criterion, such as reaching a maximum tree depth or a minimum number of samples, is met. During training, the decision tree algorithm determines the best attribute to split the data based on a metric like entropy or Gini impurity. These metrics measure the impurity or disorder in the subsets, and the algorithm aims to find the attribute that maximizes the information gain or minimizes the impurity after the split.

K-Nearest Neighbors Classification (KNC)

K-nearest neighbors classification (KNC) is a simple supervised machine learning classification algorithm operating on the principle of similarity, leveraging the proximity of instances in the feature space.[15] By identifying the K closest neighbors to a test instance from the training dataset, KNC determines its class or value through a majority voting process. This allows KNC to make reliable predictions based on the collective behavior of its nearest neighbors.

Confusion Matrix

A confusion matrix is a table that summarizes the performance of a classification model (binary or multiclass), which aims to predict a categorical label for each input instance. It displays the counts of true positives, true negatives, false positives, and false negatives. For binary classification, the matrix will be of a 2 × 2 table, where the rows represent the actual classes or labels, and the columns represent the predicted classes. For multiclass classification, the matrix shape will be equal to the number of classes, i.e., for n classes, it will be n × n. By examining the values in the confusion matrix, various performance metrics (accuracy, precision, recall, and F1 score) can be calculated.

Accuracy Score

Accuracy is a metric that measures the overall correctness or accuracy of a classification model. It calculates the proportion of correctly predicted instances out of the total number of instances. Accuracy ranges from 0 to 1, with 1 indicating perfect accuracy, meaning that all predictions are correct.

The formula for accuracy is as follows:

$$Accuracy = \frac{(True\ Positives\ +\ True\ Negatives)}{(True\ Positives\ +\ True\ Negatives\ +\ False\ Positives\ +\ False\ Negatives)}$$

In this formula, True Positives represent the number of instances correctly predicted as positive, True Negatives represent the number of instances correctly predicted as negative, False Positives represent the number of instances incorrectly predicted as positive, and False Negatives represent the number of instances incorrectly predicted as negative.

Precision

Precision is a metric that measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It quantifies the model's ability to avoid false positives. The formula for precision is as follows:

$$\text{Precision}=\frac{True\ Positives}{True\ Positives\ +\ False\ Positives}$$

Recall

A recall is a metric that measures the proportion of correctly predicted positive instances out of all actual positive instances. The formula for the recall is as follows:

$$Recall=\frac{True\ Positives}{True\ Positives\ +\ False\ Negatives}$$

F1-score

The F1 score is a metric used to evaluate the performance of a ML classification model. The F1 score is calculated using the following formula:

$$F1\ score = 2\times\frac{precision\times recall}{precision\ +\ recall}$$

It takes into account both precision and recall, providing a harmonic mean of the two values. A higher F1 score indicates better model performance, with a maximum value of 1 representing perfect precision and recall.

Permutation Feature Importance

Permutation feature importance is a technique used in machine learning to measure the importance of each feature in a predictive model.[16] It works by systematically permuting the values of a single feature and observing the resulting impact on the model's performance. By quantifying the impact of each feature on the model's performance, permutation feature importance offers insights into the global interpretation of relative importance of different features. This information helps in understanding the underlying relationships between features and the target variable.

SHAP Summary Bar Plot

The SHAP (SHapley Additive exPlanations) summary bar plot provides the visual understanding of contribution of each feature towards every prediction of the ML model.[17] This plot is particularly useful for understanding the relative importance of features and gaining insights into the model's decision-making process.

9. Details of Optimized Hyperparameters and Test Accuracy for Classification Algorithms:

Table S4. Tuned hyperparameters for selected ML classification algorithms with their corresponding test accuracy values.

S. No.	Classification Models	Optimized Hyperparameters	Test Accuracy
1.	Logistic Regression (LR)	C=0.01, penalty= 'none', solver= 'newton-cg'	100%
2.	Random Forest Classification (RFC)	'bootstrap': True, 'criterion': ‘entropy,' 'max_depth’: 32, 'max_features': 'sqrt,' 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100, 'random_state’: 95,	90%
3.	Decision Tree Classification (DTC)	'criterion’: ‘gini,' 'max_depth’: 45, 'max_features': 'sqrt,' 'min_samples_leaf': 1, 'min_samples_split': 2, ‘splitter’: best, 'random_state’: 35,	86%
4.	K-Nearest Neighbors Classification (KNC)	'metric': 'manhattan,' 'n_neighbors': 1, "weights': 'distance'	80%

10. Classification Reports:

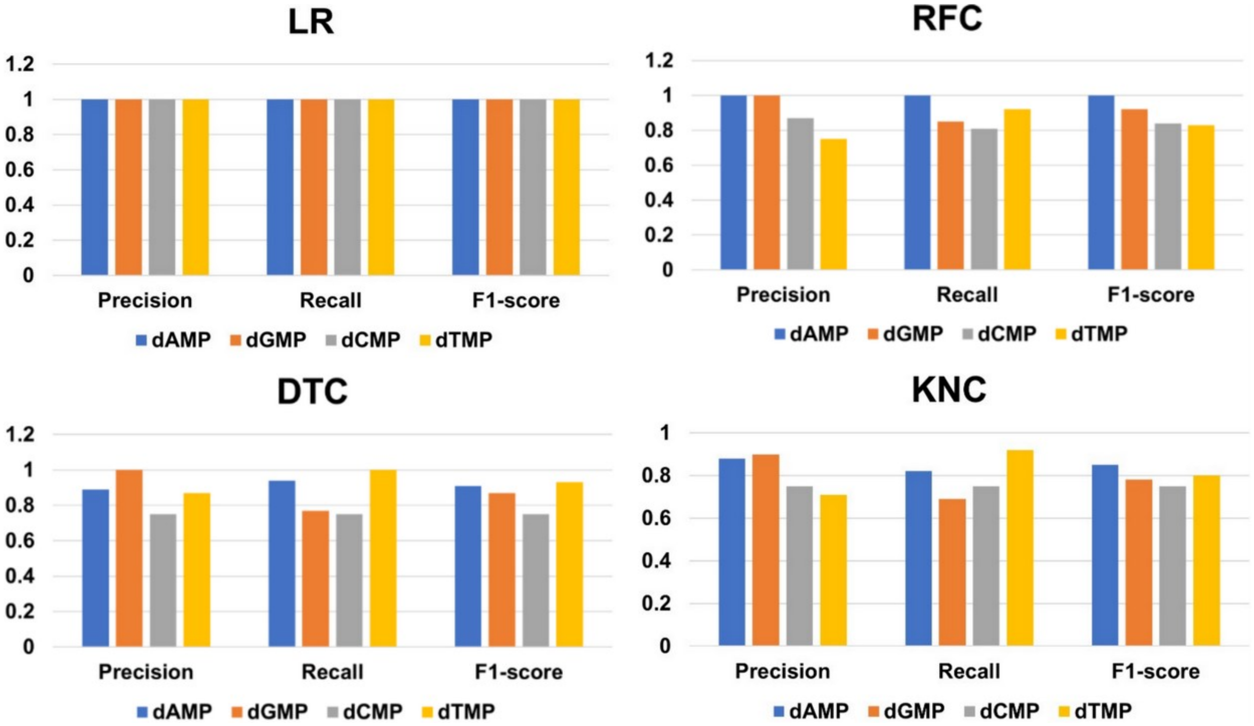
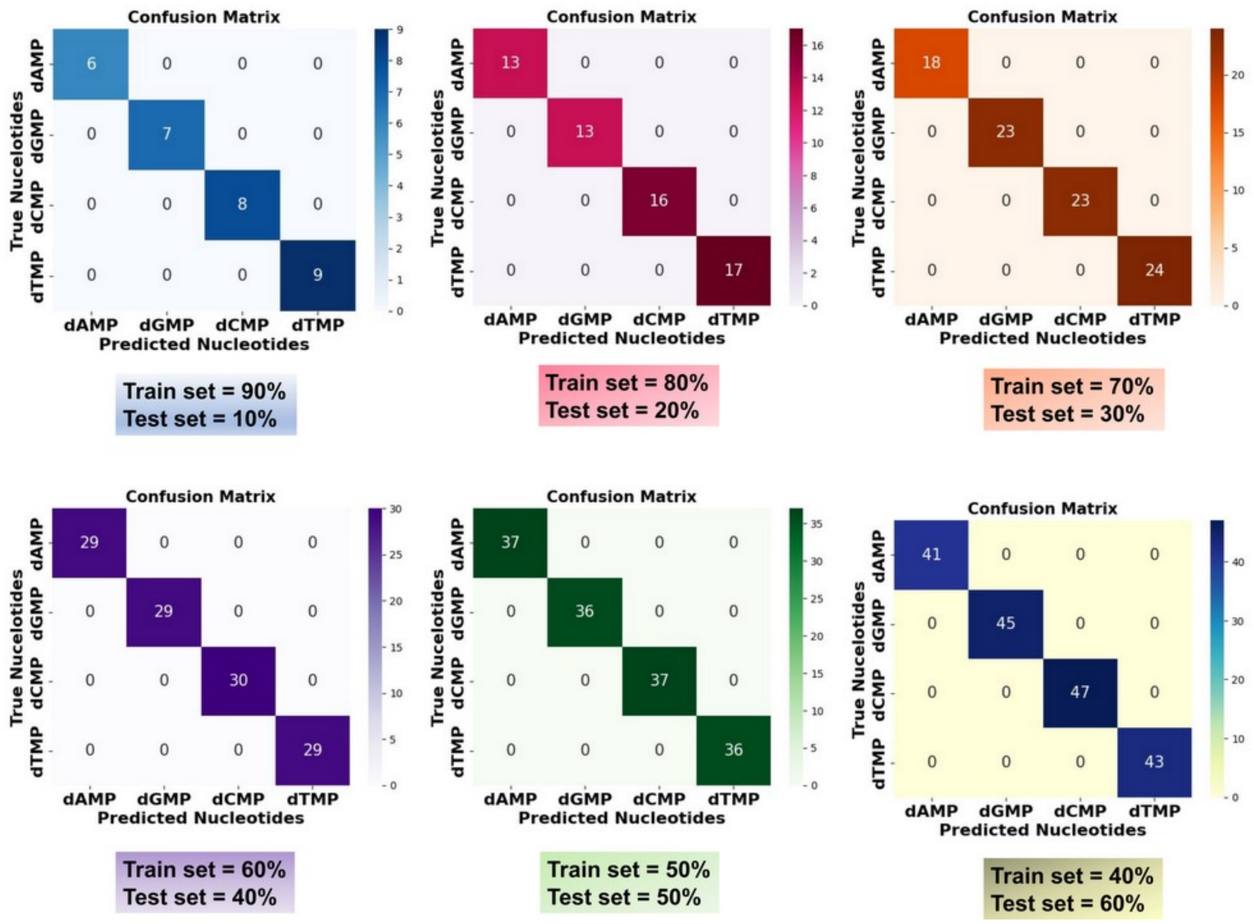


Figure S8. Classification reports of the models LR, RFC, DTC, and KNC in the quaternary classification of DNA nucleotides. The model LR is found to be the best-fitted with perfect precision, recall, and F1-score equal to 1.

11. Stability Check of Best-fitted LR Classification Algorithm:



Train set = 90%
Test set = 10%

Train set = 80%
Test set = 20%

Train set = 70%
Test set = 30%

Train set = 60%
Test set = 40%

Train set = 50%
Test set = 50%

Train set = 40%
Test set = 60%

Figure S9. Confusion matrixes for LR algorithm with different train-test split ratios.

12. Single Nucleotide Identification for Rotation Dynamics:

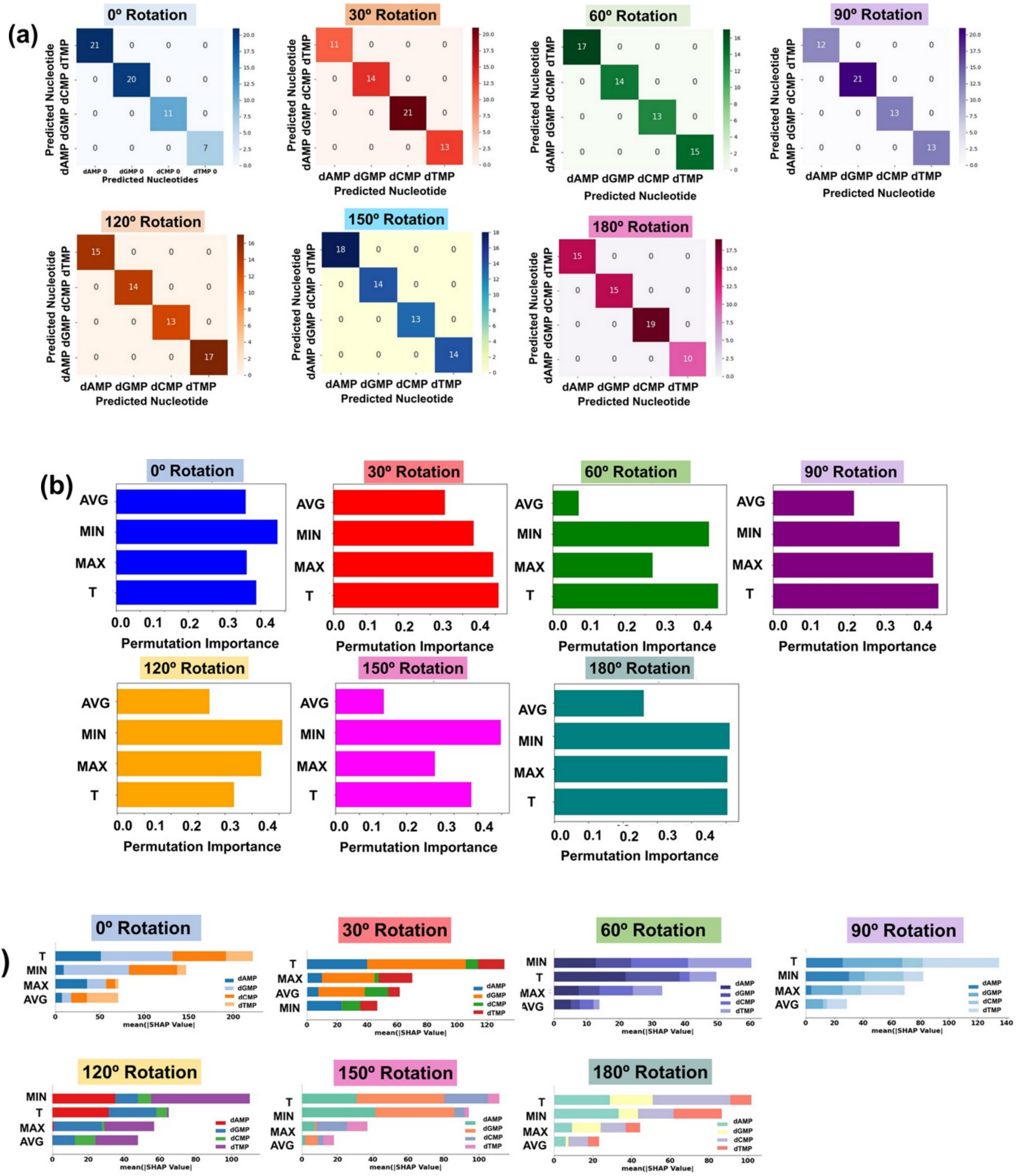


Figure S10. LR-assisted single nucleotide identification for each considered rotation of DNA nucleotide from a dataset of four types of nucleotides. **(a)** Confusion matrixes for each rotation, **(b)** permutation feature importance plots, and (c) SHAP summary bar plots. Here, Max, Min, T, and Avg stand for maxima normalized transmission (T/Tmax), minima normalized transmission (T/Tmin), transmission, and average normalized transmission (T/Tavg), respectively.

13. Transmission and Current-Voltage Plots with Underlying Physics:

The electric current I (V_b) is calculated using the equation,

$$I(V_b) = \frac{2e}{h}\int T(E,V_b)\big(f_L(E-\mu_L) - f_R(E-\mu_R)\big)dE$$

Where e is the electron charge, h is the Planck's constant, $T(E,V_b)$ is transmission function, and $f_L(E-\mu_L)$ and $f_R(E-\mu_R)$ are the Fermi functions for the electrons in the left (L) and right (R) leads, respectively.[18]

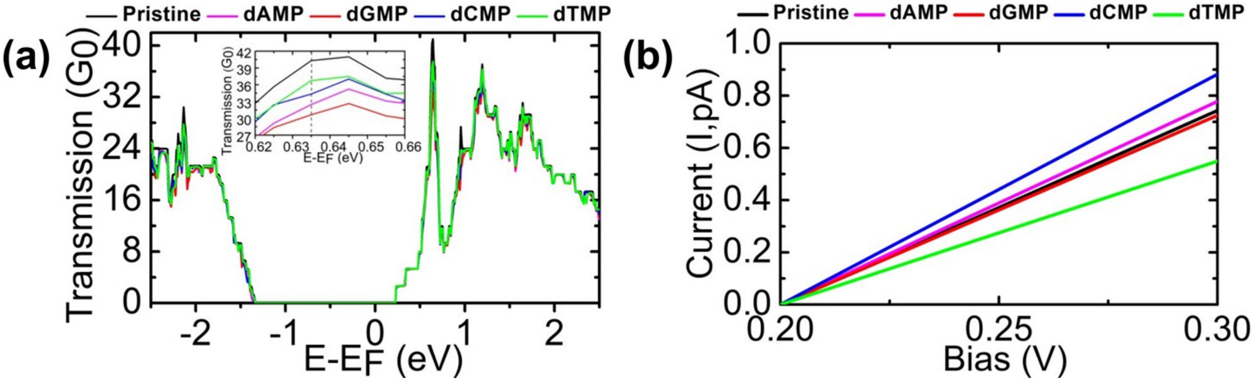


Figure S11. (a) Transmission functions plot of all four DNA nucleotides at the energy window of ± 2.5 eV, and the inset picture shows the zoomed transmission function, **(b)** I-V curve for pristine MoS_2 nanochannel and MoS_2 nanochannel+nucleotide systems. The Fermi energy (E-E_F) level is shifted to zero.

Text S4:

The transmission sensitivity is calculated by using the equation,

Transmission sensitivity (S%) = $|(G_0 - G)/G_0| \times 100\%$

where G_0 and G is the conductance of the pristine MoS_2 nanochannel and MoS_2 nanochannel +nucleotide systems, respectively.

The current-sensitivity values are calculated by using the equation, Current-Sensitivity (S%) = $|(I_0 - I)/I_0| \times 100\%$

where I_0 and I is the current of the pristine MoS_2 nanochannel and MoS_2 nanochannel+nucleotide systems, respectively.

Underlying Physics:

To better understand the underlying physics involved in the interaction between the proposed MoS_2 nanochannel device and DNA nucleotides, we further study the molecular orbitals (MOs). To investigate how different nucleotides are interacting, we study the MOs of the device in the presence and absence of DNA molecules (**Figure S12**). We found that different nucleotides affect the distribution of MOs differently, which in turn results in unique transmission fingerprints.

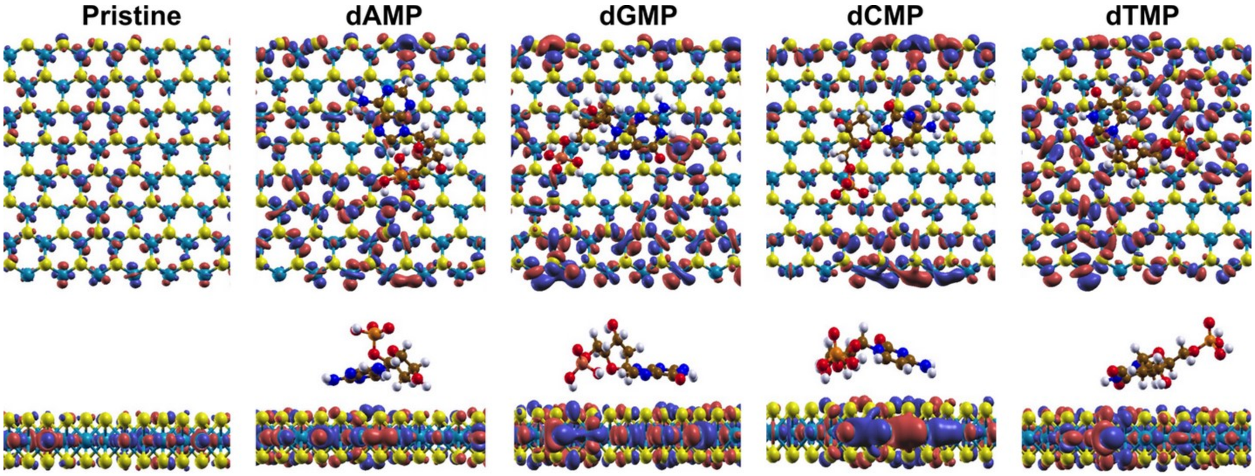


Figure S12. Isosurface plots (isosurface value is 0.05 e/Å^3) of the molecular orbitals (MOs) responsible for the sharp transmission peaks of MoS_2 nanochannel device with and without DNA nucleotides systems at an energy value of 0.635 eV. The negative and positive lobes are shown in blue and red colors, respectively.

14. Adsorption Energy (E_a) and translocation Time (τ):

To calculate the adsorption energy (E_a), the below-given equation is used,[19]

$$E_a = [E_{(MoS_2\,nanochannel\,+\,DNA\,nucleotide)} - (E_{MoS_2\,nanochannel} + E_{DNA\,nucleotide})]$$

Here, $E_{(MoS_2\,nanochannel\,+\,DNA\,nucleotide)}$ denotes the total optimized energy of the (MoS_2 nanochannel+DNA nucleotide) system; $E_{MoS_2\,nanochannel}$ and $E_{DNA\,nucleotide}$ are the single-point energy values of isolated MoS_2 nanochannel and isolated DNA nucleotide in the optimized structure of the (MoS_2 nanochannel+DNA nucleotide) setup, respectively.

Table S5. Adsorption energy values (E_a in eV) and translocation time ($\tau \propto e^{\frac{-E_i}{k_BT}}$) for MoS_2 nanochannel device with DNA nucleotides placed inside.

Nucleotide	**Adsorption Energy E_a (eV)**	**Translocation time ($\tau \propto e^{\frac{-E_i}{k_BT}}$)**
dAMP	0.73	5.66×10^{-13}
dGMP	1.01	1.35×10^{-17}
dCMP	0.99	2.38×10^{-17}
dTMP	0.66	6.06×10^{-12}

15. Charge Density Difference (CDD) Plot:

The charge density difference [$\Delta\rho(r)$] plots have been evaluated by using the following equation,

$$\Delta\rho(r) = [\rho_{(MoS_2\,nanochannel\,+\,nucleotide)}(r) - (\rho_{MoS_2\,nanochannel}(r) + \rho_{nucleotide}(r))]$$

Here, $\rho_{(MoS_2\,nanochannel\,+\,nucleotide)}(r)$ is the total charge density of the (phosphorene nanoslit+amino acid) system, and $\rho_{MoS_2\,nanochannel}(r)$ and $\rho_{nucleotide}(r)$ is the charge density on the isolated MoS_2 nanochannel device and isolated DNA nucleotide molecule, respectively, in the optimized geometry of the MoS_2 nanochannel +nucleotide systems.

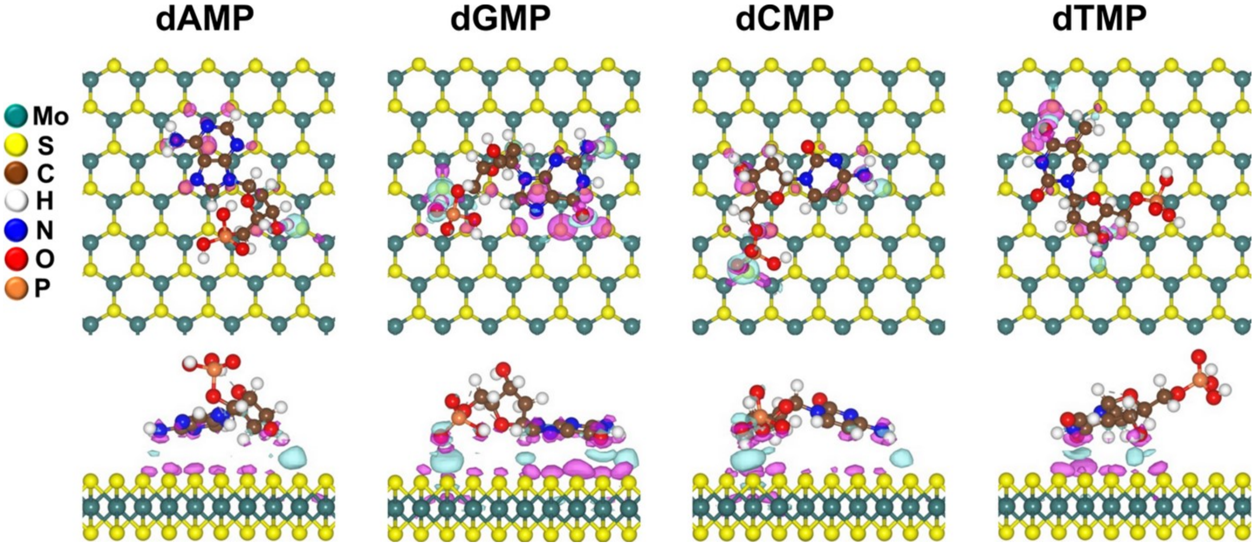


Figure S13. Charge density difference (CDD) plots for the most stable configuration of MoS_2 nanochannel+nucleotide systems with isosurface value 0.005 e/Å^3. The cyan and magenta colors represent the charge accumulation and charge depletion, respectively.

16. Effect of In-Plane Rotation on Transmission Function:

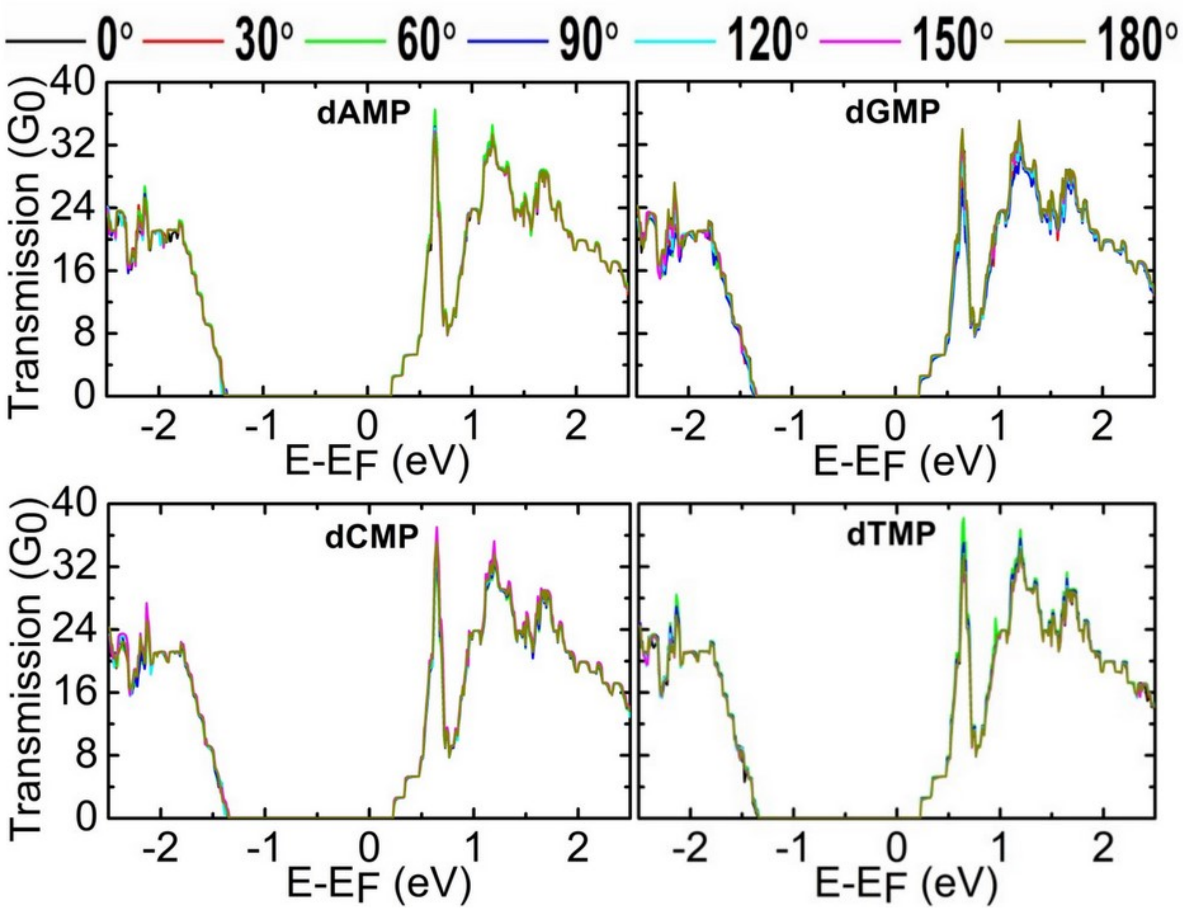


Figure S14. Variation in the transmission spectra due to in-plane rotation (from 0° to 180°) of DNA nucleotides adsorbed on the MoS_2 nanochannel surface along the *x*-axis in the yz-plane.

References:

1. A. Y.-T. Wang, R. J. Murdock, S. K. Kauwe, A. O. Oliynyk, A. Gurlo, J. Brgoch, K. A. Persson and T. D. Sparks, *Chem. Mater.*, 2020, **32**, 4954–4965.
2. K. Pearson, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 1901, **2**, 559–572.
3. H. Hotelling, *Journal of Educational Psychology*, 1933, **24**, 417–441.
4. T. Chen and C. Guestrin, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA, 2016, pp. 785–794.
5. J. H. Friedman, *The Annals of Statistics*, 2001, **29**, 1189–1232.
6. L. Breiman, *Mach. Learn.*, 2001, **45**, 5–32.
7. P. Geurts, D. Ernst and L. Wehenkel, *Mach. Learn.*, 2006, **63**, 3–42.
8. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye and T.-Y. Liu, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 3149–3157.
9. M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, Gaussian 09, revision A.1; Gaussian, Inc.: Wallingford, CT, 2009.
10. P. Ordejón, E. Artacho and J. M. Soler, *Phys. Rev. B*, 1996, **53**, R10441–R10444.
11. C. Lee, W. Yang and R. G. Parr, *Phys. Rev. B*, 1988, **37**, 785–789.
12. N. Troullier and J. L. Martins, *Phys. Rev. B*, 1991, **43**, 1993–2006.
13. D. R. Cox, *Journal of the Royal Statistical Society: Series B (Methodological)*, 1958, **20**, 215–232.

14. Y. SONG and Y. LU, *Shanghai Arch Psychiatry*, 2015, **27**, 130–135.

15. T. Cover and P. Hart, *IEEE Transactions on Information Theory*, 1967, **13**, 21–27.

16. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.

17. S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal and S.-I. Lee, *Nat. Mach. Intell.*, 2020, **2**, 56–67.

18. J. Prasongkit, A. Grigoriev, B. Pathak, R. Ahuja and R. H. Scheicher, *Nano Lett.*, 2011, **11**, 1941–1945.

19. J. Prasongkit, E. de Freitas Martins, F. A. L. de Souza, W. L. Scopel, R. G. Amorim, V. Amornkitbamrung, A. R. Rocha and R. H. Scheicher, *J. Phys. Chem. C*, 2018, **122**, 7094–7099.