JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Comparative analysis of protein sequence classification using machine learning and deep learning techniques

Shubhankit Sudhakar
M.Tech (Artificial Intelligence)
ASET, Amity Unniversity
Noida, India
shubhankit.sudhakar14@gmail.com

Jitendra Singh Jadon
Asst. Professor
dept. of Artificial Intelligence
ASET, Amity University
Noida, India
jsjadon@amity.edu

Prof. Dr. Archana Singh
Head of dept. of Artificial Intelligence
ASET, Amity University
Noida, India

***Abstract*— With the advent of COVID-19 pandemic protein sequencing and identification of the protein is necessary to track the functionality of the organism and the activity associated with it. Using spectrometer, Xray diffraction and other experimental methods to determine the class of protein it belongs to is very expensive. Therefore, the need for automated tools that can classify proteins with the known family of classification. This paper gives comparative analysis of several ML and DL algorithms that learn such classification from the protein sequence itself. In addition, numerous techniques that mix different learning algorithms to improve prediction accuracy are reviewed. The experiment results reveal insights that can assist biologists and computer scientists construct high-performance, high-quality protein categorization systems. Using data from the Protein Data Bank (PDB), With our model proposed, we achieved the accuracy of 96.43%.**

[1] ***Keywords—Machine Learning; Exploratory Data Analysis; Decision Tree; Random Forest; KNN; LinearSVC; Logistic Regression; Multinomial Naïve Bayes; XGBoost; Gradient Descent; Adaboost; CNN; RNN; LSTM; BiLSTM Accuracy; Precision; Recall; F1-Score, computation biology***

I. INTRODUCTION

The macromolecules known as proteins are in charge of controlling and regulating how the body's tissues and organs work. Proteins are made up of chains of amino acids, known as polypeptide chains, whose structure is dictated by the arrangement of nucleotides in a gene. Whenever a novel protein is identified, its structural and functional characteristics can be proposed depending on the classes of previously characterized proteins, the new one is closely related to. Proteins that are homologous share comparable structures, functions, and overall behaviours. This is what protein categorization aims to achieve. Protein categorization applications where proteins in the same class are anticipated to share comparable activity and structure have found success using this subfield of machine learning, known as supervised learning. Classification techniques are effective for grouping specimens within data into groups with similar patterns and have been widely utilised in stats, information science, genetics, and the humanities. Since billions of proteins have been examined still not properly divided into protein families, employing classification methods to protein homology modelling is a crucial issue. For instance, the weakly annotated TrEMBL dataset has around 150 million proteins, whereas the well-curated protein dataset has 720K annotated proteins.

II. APPROACH

Exploratory Data Analysis (EDA)

After cleaning the data from protein data, data from first dataset include the columns i.e., structureID, classification, experimental Technique, Macromolecule type, residueCount, resolution, structure molecular weight, crystallization method, crystallization Temp(K), density Matthews, density PercentSol, pbx Details, ph value, publication year. The data from second dataset include four columns i.e., structureId, Sequence, residue count, macromolecule type. An exploratory data analysis performed on both the dataset, For first dataset, primarily number of unique items in classification is 5050, experimental technique is 33. Correlation between the data is shown in figure 1,

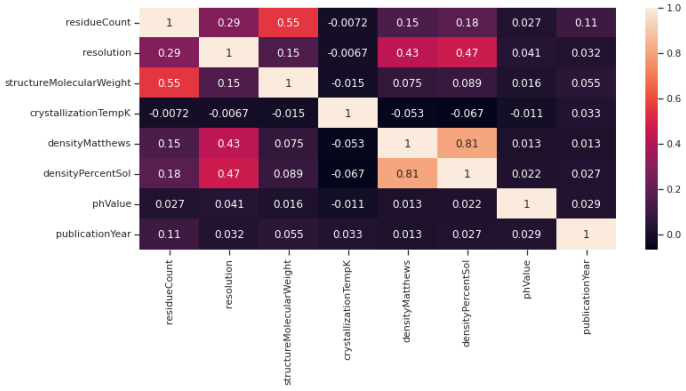

Figure:1 Correlation Matrix

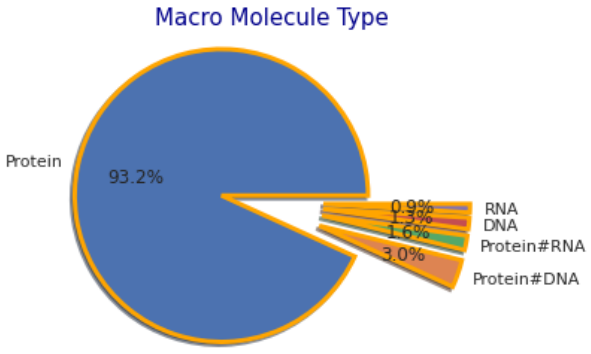

Figure:2 Macromoleculetypes

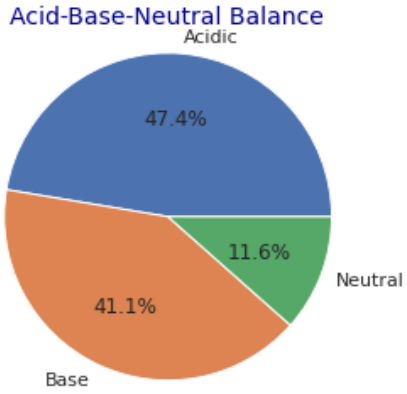

Figure:4 ph categorizarion

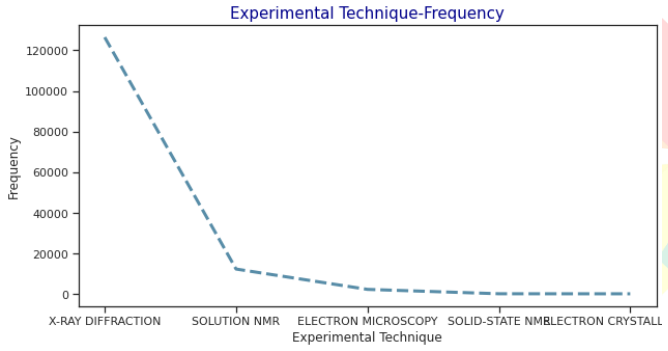

Figure:3 Line Chart

	Missing Values	% Value
crystallizationMethod	45159	31
crystallizationTempK	44362	31
phValue	36293	25
publicationYear	23800	16
densityMatthews	16677	11
densityPercentSol	16652	11
resolution	12812	9
macromoleculeType	3765	2
classification	2	0
structureId	0	0
experimentalTechnique	0	0
residueCount	0	0
structureMolecularWeight	0	0

Figure:5 Missing Value Statistics

Macromolecules type is shown in figure 2, type of Experimental technique frequency used to sequence protein is shown in figure 3 with the line chart, ph-value of proteins classifying them among acidic(<7), neutral(=7) and basic (>7) is shown in figure 4, Missing value counts and its percentage in the dataset is shown in figure 5 and handling the missing values after the count, we used mean, std – method, k-NN imputer and also removing the NaN values to get the rightful data for the classification. For second dataset which has the sequence with the molecules of different amino acids. The combined data set after preprocessing both the dataset we took only 2 columns i.e., Sequence and Classification with maximum count possible and the limiting the classes for classification to 20, for feeding the data to our different models for comparative analysis. As shown in figure 6, the maximum number of Hydrolase and minimum count of metal Binding Protein. A histogram of seq. length is shown in figure 7.

Finally, we used tokenizer and TF-IDF transformer on final dataset before using various techniques for classification. We used test-train split with the test size of 0.30 i.e, 30% of the data.[13]

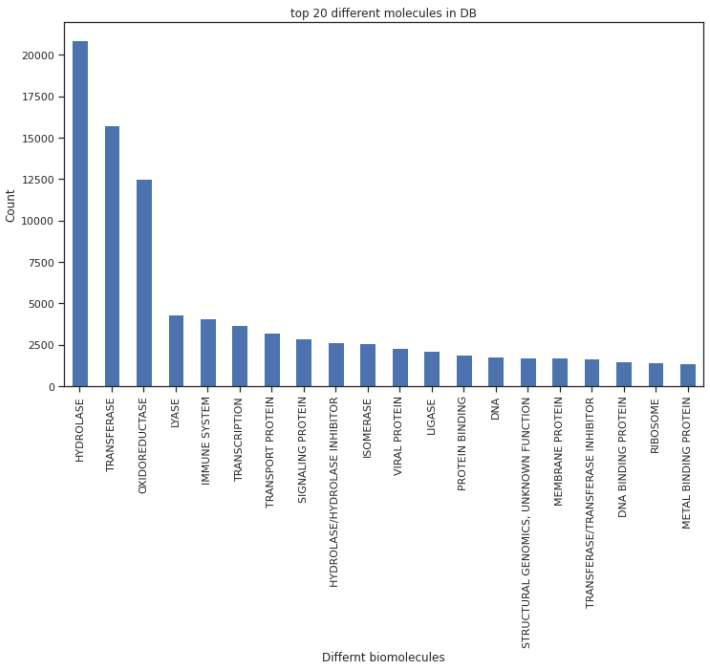

Figure:6 Top 20 classes count

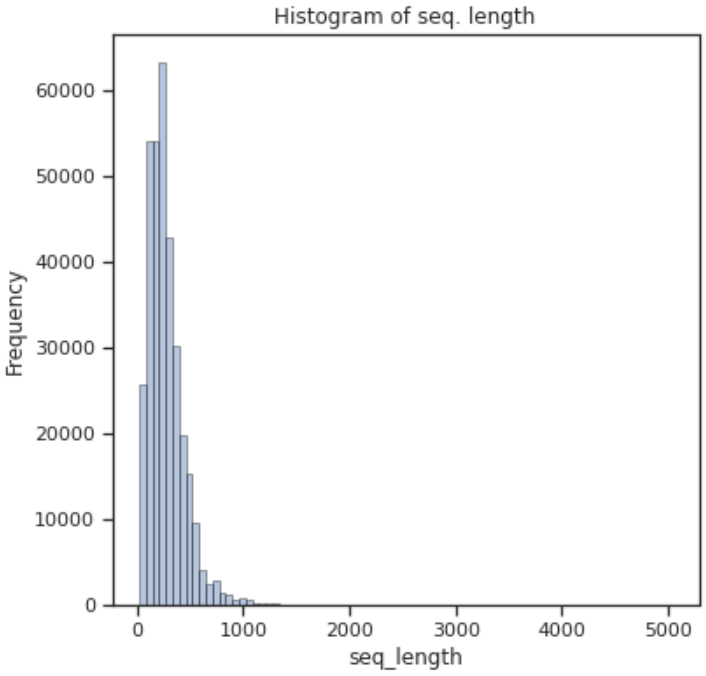

Figure:7 Histogram of sequence length

Machine Learning Techniques

Multinomial Naïve Bayes

Naïve bayes Multinomial For the examination of categorical textual data, it is one of the most widely used supervised learning classifications, mostly used for natural language processing (NLP). The technique is based on the Bayes theorem, and it determines which tags have the highest output probabilities by computing their probabilities.
The projected class as output is c, if the no. of docs(n) fall into k classes where k ∈ {c1, c2, .., ck.} The Naive Bayes method may be expressed as follows: where P(w|c) is the probability of seeing word w given class c and n_{wd} is the number of times word w occurs in document. [30]:

$$P(w \mid c) = \frac{1 + \sum_{d \in D_c} n_{wd}}{k + \sum_{w'} \sum_{d \in D_c} n_{w'd}}$$

Logistic regression (LR)

In order to include more than two potential categorical labels, multinomial regression is an extension of LR. Multinomial (or multilabel) logistic classification [4][5][8][31] uses the probability of x belonging to
class i (as defined in Equation)

$$p\left(y^{(i)} = 1 \mid x, \theta\right) = \frac{\exp\left(\theta^{(i)^T} x\right)}{\sum_{j=1}^{m} \exp\left(\theta^{(i)^T} x\right)}$$

where $\theta^{(i)}$ is the weight vector for class i. For multinomial logistic regression (m > 2), the SoftMax function is typically used, as opposed to binary classification (m = 2), referred to as a fundamental LR. This is how normalisation works:

$$\sum_{i=1}^{m} p\left(y^{(i)} = 1 \mid x, \theta\right) = 1$$

The training data subset is used to determine the component that belongs to class I in a classification problem as supervised learning context, where i ∈ {1, .. n}. We must maximise the log-likelihood function as follows in order to execute maximum likelihood (ML) estimation of:

$$\ell(\theta) = \sum_{j=1}^{n} \log p\left(y_j = 1 \mid x_j, \theta\right)$$

$$= \sum_{j=1}^{n} \left[\sum_{i=1}^{m} y_j^{(i)} \theta^{(i)^T} x_j - \log \sum_{i=1}^{m} \exp\left(\theta^{(i)^T} x_j\right) \right]$$

$$\hat{\theta}_{MAP} = arg \max_{\theta} L(\theta) = arg \max_{\theta} \left[\ell(\theta) + \log p(\theta)\right]$$

This result naturally suggests that, as demonstrated by [6], maximising likelihood is equivalent to minimising KL divergence, i.e., maximising the likelihood that our observed data will follow our model's predictions while minimising the discrepancy between those predictions and the actual data distribution.

Decision Trees

As similarly as a tree is observed in nature with branches, leaves and roots, decision tree follows with the same structure, consisting of nodes instead d of physical entities naming as “root node”, “branches” and “leaf nodes”. Implicitly computing the attribute division at each split level, as the split is observed over the node to generate a branch along with class or the categorical label, resulting in the generation of the leaf node or multiple leaves as for the case of multivariate values, with designated labels for each leaf node. The advantage that the decision tree provides is with the ability of selection for the most biased feature in the dataset and the comprehensibility nature, performance not being affected by the non-linear flow of algorithm. The attributes used to split to test at any node in order the determine the ’Best’ splitting in individual classes, resulting in the branching to be as ’Pure’, provided the splitting criteria has to be identical [38].
The decision tree algorithm was utilized here with two classification criterions are followings:

Gini index or the Gini Impurity computes the probabilistic classification of feature that is incorrectly specified upon selected randomly. The scalability of the Gini Index varies between 0 and 1, pure classification being expressed by the 0 whereas 1 point towards the random distribution of the samples across the classes. The Gini Index is deployed in the CART, Classification and Regression Tree algorithm.

$$\text{Gini Index} = 1 - \text{Sum}(p_i)^2$$

p_i: probability of a sample to classified among classes

The information gains us used for selection of the feature providing the maximal information about the classification based upon the entropy, uncertain, disorder or impurity is quantified. The entropy is descending from the root nodes to leaves.

p_i: Probability of being a function of entropy

$$\text{Entropy} = -\text{Sum}\ (p_i * \log_2(p_i))$$

Random Forest

The Random Forest is an operative consisting of multiple ensembled individual decision trees. Each tree or a unit is capable of individually predicting a class, following the persona of wisdom of crowds. Each feature is capable of performing individual classification as primary splitting criteria for individual trees in the forest. The criteria used for the features is, feature importance, calculated as the reduction in the node impurity weighted by the probability of the attaining the node, the higher the value the more desirable that feature tends to be [11] [33]. For each node importance is calculated, assuming for binary classification, where n is priority of node j, W is number of weighted samples at node j and C is impurity associated with the node.

$$Nij = W_j\ C_j - W_{left(j)} C_{left(j)} - W_{right(j)}\ C_{right(j)}$$

KNN Classifier

A classification method that is non-parametric is the k-nearest Neighbors algorithm (KNN) [10]. Many research fields [34] have employed this method for text classification applications in the past. The notion is The KNN algorithm locates a test document's k closest neighbors among all the documents in the training set, and then ranks the category choices based on the k neighbors’ classification.
The score of the category of the neighbour documents could be determined by how similar x and each neighbouring document are. The class k document’s similarity score to the test document x would be calculated by adding the scores of all KNN documents that fall within the same class. The

method places the candidate in the class with the highest test score after sorting the score values, document x [34]. It is designed by a N-D data set. The decision rule of KNN is:

$$f(x) = arg\max_{j} S(x, C_j)$$
$$= \sum_{d_i \in KNN} sim(x, d_i) y(d_i, C_j)$$

AdaBoost

Adaptive Boosting also known as AdaBoost is an ensemble method in machine learning building upon the decision trees, as utilizing individual split decision trees known as weak learners or decision stumps. It then combines that weak learners or weak classifier together to form a united single strong classifier, as for every class to be classified for the available samples [35].

Linear SVC

Similar to SVC with kernel='linear', but implemented using liblinear rather than libsvm, giving it more flexibility in the selection of penalty and loss functions and allowing it to scale more well to huge numbers of samples.[37]

This class supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.[36]

Deep Learning Techniques [12]

CNN

In this study, we use the Keras framework to create a CNN for classifying protein families. Convolution, rectifier, maxpooling, dropouts, and a MLP are the five steps that make up our model. Each convolution stage has 128 filters with sizes ranging from four to six, each of which learns a distinct pattern from the data. We discovered that these specific sizes and numbers of filters performed well for this assignment. The authors can choose the size and quantity of filters as a hyperparameter. ReLU, an element - wise function that changes all negative values to near zero, is a nonlinearity function that is used in conjunction with the previous stage during the rectification step. In this case, we use leaky ReLU rather than conventional ReLU to prevent vanishing gradient problems that can arise due to the scale of our model.

$$f_{\text{leaky ReLU}}(z) = \begin{cases} \alpha x & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

The output of earlier stages is then denoised and translational invariant characteristics are learned using recurrent maxpooling, which chooses the maximum values within a certain timeframe. We chose a window size of 2 for this CNN model, which is conservatively enough to allow enough information to produce a reliable prediction. After pooling, there is a technique called dropout [28] that works to avoid overfitting to the subsequent layer during training with a predetermined probability of 0.2. It should be emphasized that this approach is quite similar to conventional machine learning, we train this model to minimize our categorical cross-entropy [3] loss function using the Adam optimizer. This model's architecture has the following structure:

$$f(y) = f_{mlp}(f_{dropout}(f_{maxpool}(f_{lrelu}(fconv(x)))))$$

RNN

The LSTMs and bidirectional LSTM architecture used in this work's RNN is implemented using the Keras framework. We converted are final dataset into a format useful for these kinds of models because this architecture has been thoroughly researched and there is a wealth of scholarly literature on the subject.[16][17]
Remember that the input for RNN models is a list of tokens, each of which is made up of letters from the alphabet. The available data does not meet this requirement for lengthy protein sequences. So, using the following method, we may try to turn our model into a language by tokenizing a protein sequence [29]. After tokenization, the protein sequence data that our character-based language model uses, tokenized at the level of amino acids. These model's architecture has the following structure:

$$f(y) = fmlp(f_{BiLSTM}(f_{embedding}(x)))$$

We used dropouts and callbacks for early stopping to avoid overfitting.

Improvised CNN

In our improvised CNN, model architecture is combines with encoder model of transformer for better encoding the input data for improved accuracy, we used leaky RELU in between layer and sigmoid instead of SoftMax function and Nadam optimizer.[12]. Callbacks and dropouts with the initial probability of 0.20 is used to avoid overfitting.

CNN+RNN

In our ensemble model of CNN and LSTM with 20 nodes, flatten layer output was feed into LSTM, later same MLP process is used to get output. Similar process is used by using BiLSTM with 20 nodes. In both the models we used SoftMax in between layers for CNN and sigmoid after RNN networks and in MLP network.[28][29].
We used dropouts and callbacks for early stopping for avoiding overfitting.

III. RESULTS

Model	Accuracy	Precision	Recall	F1-Score	MAE (h)
MNB	73.57	86.03	76.0	76.7	0.67
MLR	84.72	89.12	85.6	84.57	0.71
DT(Gini)	68.44	71.22	70.56	68.3	0.82
DT(Entropy)	70.55	74.31	68.38	75.2	0.80
RF	92.04	88.65	87.24	89.3	0.81
KNN	78.99	73.1	71.01	77.3	0.77
AdaBoost	30.21	28.9	27.8	26.4	0.93
LinearSVC	90.42	91.1	89.76	88.7	0.43
CNN	91.48	83.2	82.55	86.54	0.49
LSTM	93.53	87.1	85.33	89.6	0.48
BiLSTM	94.14	93.89	91.2	92.77	0.41
CNN*	95.67	96.1	95.44	93.8	0.36
CNN+LSTM	95.32	95.4	94.72	95.2	0.28
CNN+BiLSTM	96.43	95.8	95.2	96.1	0.22

In our comparative analysis, we found that the worst model in performance was AdaBoost with accuracy approx. of 30% and MAE score of 0.93 and the best performers was improvised CNN and the ensemble model of CC and BiLSTM with the accuracy of almost 96% and 97% respectively. Our models outperform previous model

proposed KNN model whose accuracy was 0.41 [7] and our alone KNN classifier has accuracy of 0.789931.

IV. CONCLUSION

The problem statement we chose to classify the protein sequence to the respective classes that includes our 20 classes we used as shown in figure 6. The accuracy, precision and F1-score of our ensemble models out performed previous proposed models but further investigation is need with more data and use of top models which include latest transformer models.

V. REFERENCES

[1] Dongardive, Jyotshna, and Siby Abraham. "Protein Sequence Classification Based on N-Gram and K-Nearest Neighbor Algorithm." Computational Intelligence in Data Mining—Volume 2, edited by Himansu Sekhar Behera and Durga Prasad Mohapatra, vol. 411, Springer India, 2016, pp. 163–71, doi:10.1007/978-81-322-2731-1_15.

[2] Strodthoff, Nils, et al. "UDSMProt: Universal Deep Sequence Models for Protein Classification." Bioinformatics, edited by Yann Ponty, vol. 36, no. 8, Apr. 2020, pp. 2401–09, doi:10.1093/bioinformatics/btaa003.

[3] Czepiel, Scott A. Maximum Likelihood Estimation of Logistic Regression Models: Theory and Implementation. p.23.

[4] Böhning, Dankmar. "Multinomial Logistic Regression Algorithm." Annals of the Institute of Statistical Mathematics, vol. 44, no. 1, Mar. 1992, pp. 197–200, doi:10.1007/BF00048682.

[5] Royer, Clément W., et al. "A Newton-CG Algorithm with Complexity Guarantees for Smooth Unconstrained Optimization." Mathematical Programming, vol. 180, no. 1, Mar. 2020, pp. 451–88, doi:10.1007/s10107-019-01362-7.

[6] Murphy, K. P. Machine Learning: A Probabilistic Perspective. MIT Press, 2012, https://books.google.com/books?id=RC43AgAAQBAJ.

[7] Jiang, Shengyi & Pang, Guansong & Wu, Meiling & Kuang, Limin. (2012). An Improved k-Nearest Neighbor Algorithm for Text Categorization. Expert Systems with Applications. 39. 1503-1509. 10.1016/j.eswa.2011.08.040.

[8] R. Duncan Luce. Individual Choice Behavior: A Theoretical Analysis. John Wiley & Sons, 1959.

[9] Jurafsky, Daniel & Martin, James. (2008). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.

[10] Abu Alfeilat, Haneen Arafat, et al. "Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review." Big Data, vol. 7, no. 4, Dec. 2019, pp. 221–48, doi:10.1089/big.2018.0175.

[11] Geurts, Pierre, et al. "Extremely Randomized Trees." Machine Learning, vol. 63, no. 1, Apr. 2006, pp. 3–42, doi:10.1007/s10994-006-6226-1.

[12] Jha, Anupama, Matthew R. Gazzara, and Yoseph Barash. "Integrative deep models for alternative splicing."Bioinformatics 33.14 (2017): i274-i282.

[13] Quang, Daniel, Yifei Chen, and Xiaohui Xie. "DANN: a deep learning approach for annotating the pathogenicity of genetic variants." Bioinformatics 31.5 (2015): 761-763.

[14] Liu, Feng, et al. "PEDLA: predicting enhancers with a deep learning-based algorithmic framework." Scientific reports 6.1 (2016): 1-14.

[15] Li, Yifeng, Wenqiang Shi, and Wyeth W. Wasserman. "Genome-wide prediction of cis-regulatory regions using supervised deep learning methods." BMC bioinformatics 19.1 (2018): 1-14.

[16] Angermueller, Christof, et al. "DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning." Genome biology 18.1 (2017): 1-13.

[17] Quang, Daniel, and Xiaohui Xie. "DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences." Nucleic acids research 44.11 (2016): e107-e107.

[18] Degroeve, Sven, et al. "Feature subset selection for splice site prediction." Bioinformatics 18.suppl_2 (2002): S75-S83.

[19] Bucher, P. "Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences." Journal of molecular biology vol. 212,4 (1990): 563-78. doi:10.1016/0022-2836(90)90223-9

[20] Heintzman, N., Stuart, R., Hon, G. et al. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. Nat Genet 39, 311–318 (2007).

[21] Segal, Eran et al. "A genomic code for nucleosome positioning." Nature vol. 442,7104 (2006): 772-8. doi:10.1038/nature04979

[22] Ohler, U., Liao, Gc., Niemann, H. et al. Computational analysis of core promoters in the Drosophila genome. Genome Biol 3, research0087.1 (2002)

[23] Alipanahi, B., Delong, A., Weirauch, M. et al. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol 33, 831–838 (2015).

[24] Kelley, David R., Jasper Snoek, and John L. Rinn. "Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks." Genome research 26.7 (2016): 990-999.

[25] Zhou, J., Troyanskaya, O. Predicting effects of noncoding variants with deep learning–based sequence model. Nat Methods 12, 931–934 (2015).

[26] Ritambhara Singh, Jack Lanchantin, Gabriel Robins, Yanjun Qi, DeepChrome: deep-learning for predicting gene expression from histone modifications, Bioinformatics, Volume 32, Issue 17, 1 September 2016.

[27] Qi, Yanjun et al. "A unified multitask architecture for predicting local protein properties." PloS one vol. 7,3 (2012): e32235. doi:10.1371/journal.pone.0032235

[28] Srivastava, Nitish & Hinton, Geoffrey & Krizhevsky, Alex & Sutskever, Ilya & Salakhutdinov, Ruslan. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 15. 1929-1958.

[29] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems (Vol. 26). Curran Associates, Inc.

[30] Frank, E.; Bouckaert, R.R. Naive bayes for text classification with unbalanced classes. In European Conference on Principles of Data Mining and Knowledge Discovery; Springer: Berlin/Heidelberg, Germany, 2006, pp. 503–510.

[31] Krishnapuram, B.; Carin, L.; Figueiredo, M.A.; Hartemink, A.J. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. IEEE Trans. Pattern Anal. Mach. Intell. 2005, 27, 957–968. [CrossRef]

[32] Johnson, W.B.; Lindenstrauss, J.; Schechtman, G. Extensions of lipschitz maps into Banach spaces. Isr. J. Math. 1986, 54, 129–138. [CrossRef]

[33] Ali. "Random Forests and Decision Trees." International Journal of Computer Science Issues (2012).

[34] Jiang, S.; Pang, G.; Wu, M.; Kuang, L. An improved K-nearest-neighbor algorithm for text categorization. Expert Syst. Appl. 2012, 39, 1503–1509. [CrossRef]

[35] Wang, Ruihu. "AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review." Physics Procedia (2012): 800-807.

[36] Burges CJC. A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov. 1998;2:121-167.

[37] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. New York, NY: Springer-Verlag; 2009.

[38] Patel, Harsh H and Purvi Prajapati. "Study and Analysis of Decision Tree Based Classification Algorithms." International Journal of Computer Sciences and Engineering (2018).