

Article

Prediction of Transcription Factor Binding Sites of SP1 on Human Chromosome1

Maiada M. Mahmoud, Nahla A. Belal * and Aliaa Youssif

College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Smart Village 12577, Egypt; maiada@aast.edu (M.M.M.); aliaay@aast.edu (A.Y.)

* Correspondence: nahlabelal@aast.edu

Abstract: Transcription factors (TFs) are proteins that control the transcription of a gene from DNA to messenger RNA (mRNA). TFs bind to a specific DNA sequence called a binding site. Transcription factor binding sites have not yet been completely identified, and this is considered to be a challenge that could be approached computationally. This challenge is considered to be a classification problem in machine learning. In this paper, the prediction of transcription factor binding sites of SP1 on human chromosome1 is presented using different classification techniques, and a model using voting is proposed. The highest Area Under the Curve (AUC) achieved is 0.97 using K-Nearest Neighbors (KNN), and 0.95 using the proposed voting technique. However, the proposed voting technique is more efficient with noisy data. This study highlights the applicability of the voting technique for the prediction of binding sites, and highlights the outperformance of KNN on this type of data. The study also highlights the significance of using voting.

Keywords: classifier; machine learning; supervised; voting; transcription factor; binding site

Citation: Mahmoud, M.M.; Belal, N.A.; Youssif, A. Prediction of Transcription Factor Binding Sites of SP1 on Human Chromosome1. *Appl. Sci.* **2021**, *11*, 5123. <https://doi.org/10.3390/app11115123>

Academic Editor: Giancarlo Mauri

Received: 28 April 2021

Accepted: 27 May 2021

Published: 31 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

DNA sequences of living organisms contain information that create proteins. An important step in the creation of proteins from DNA is the transcription step. Transcription is the process of copying the information of a gene's DNA strand into a messenger RNA (mRNA) molecule [1].

An enzyme called RNA polymerase and proteins called transcription factors (TFs) carry out the transcription process. When a gene is to be transcribed, the enzyme must bind itself to the DNA of the gene at a specific sequence called the “promoter” sequence. This is done with the help of the transcription factors. Both the enzyme and the transcription factors start the transcription process after binding to the promoter site and terminate the transcription once the mRNA strand is completed. The generated mRNA copy serves as a blueprint for protein synthesis [1–3].

Most of the transcription factor binding sites are close to the gene's promoter. However, due to the repetitive nature of DNA sequences, the binding sites could also be found at a further point or at multiple locations in the DNA, while still affecting the transcription of that gene. Accurate prediction of binding sites for TFs is a known problem that is still challenging in the computational biology field due to its sequence variation.

Transcription factor binding sites play a key role in gene expression. Therefore, to understand gene regulation networks, it is obligatory to understand transcription factor binding at the genome scale. Moreover, transcription factor binding sites play an important role in drug design.

The problem targeted in this paper concentrates on the transcription factor binding sites of SP1 on human chromosome 1. SP1 is a transcription factor protein coding gene. This gene is responsible for a number of diseases, among which is Huntington Disease and Embryonal Carcinoma.

Supervised machine learning techniques could be used to extract knowledge from known binding and non-binding sequences to predict whether a given new sequence is also a binding site or not. This is considered as a binary classification problem, in which there are only two possible classes for the result. In the following section, a brief background about machine learning techniques is given, followed by related work in the prediction of transcription factor binding sites.

1.1. Background

Logistic Regression [4] is a commonly used technique in classification problems that is applied when the values are completely different from each other. It is a statistical model in which a logistic curve is fitted to the dataset. It calculates the probability of the default class based on the features. If the probability turns out to be greater than a specified threshold, it predicts the value of the target value as the default class.

Linear Discriminant Analysis (LDA) [5] is mainly used for dimensionality reduction. It finds a linear combination of features that separates two or more classes of objects. After the dimensionality reduction, it could be viewed as a classification algorithm.

K-Nearest Neighbor (KNN) [6] classifies a new data point by searching the entire training set for the k most similar instances by being closest to the test data point. The value for k is preferred to be an odd number and it has a major effect on the algorithm's performance. If k is too small, the model becomes sensitive to outliers.

Decision Trees [6,7] use the divide and conquer approach. They are not affected much by outliers and can deal with linearly inseparable data. They split the data based on the features, and there are many splitting criteria available (e.g., Gini coefficient, entropy metrics, etc.). Errors propagate through trees, which becomes a big problem as the number of classes increases. Without proper pruning, the Decision Tree can easily overfit.

Random Forest [8] is a bagging ensemble algorithm, meaning that it uses multiple different algorithms and generates a final result based on them. Random Forest trains many Decision Trees and returns the class that had the majority in the trees' decisions. With the increase in the number of trees the algorithm becomes computationally slow.

Naïve Bayes [6,9] is a simple yet powerful classification algorithm. It is a conditional probability model in which it assumes that a value of a particular feature is independent of the value of any other feature. This assumption is unrealistic in real life data; however, the algorithm is still effective in multiple problems. An advantage of this algorithm is that it requires a small number of training data to estimate the parameters for classification, and it is considered extremely fast when compared to other more advanced algorithms. The Naïve Bayes model can be directly calculated by calculating the probability of each class of the problem and calculating the conditional probability for each class given each x value.

Support Vector Machine (SVM) [6,10] is considered a complex algorithm but can provide high accuracy. It works well even if data are not linearly separable in feature space, given the appropriate kernel. The algorithm is based on the concept of maximizing the minimum distance from hyperplane to the nearest sample point. Its performance is dependent on choice of features.

AdaBoost [10–12], short for adaptive boosting, is an ensemble boosting algorithm. A boosting algorithm uses weak learner classifiers and transforms their output into a strong learner. AdaBoost adapts multiple Decision Trees by using them consecutively; each tree improves on the results of the previous one by attempting to correct its errors. Models are added until the training set is predicted perfectly, or a maximum number of models are added. Predictions of the last model is the sum of weights of the predictions made by all of the previous models.

Gradient Boosting [13] is also an ensemble boosting algorithm that is similar to AdaBoost. However, the main difference is the technique used to identify the errors of the weak learners. Gradient Boosting identifies the shortcoming of weak classifiers by using gradients in the loss function.

The Extra Trees [14] algorithm is short for extremely randomized trees. It is another bagging ensemble algorithm, and it is similar to Random Forest, with the main difference of choosing decision boundaries randomly instead of those same boundaries being based on the best choice, as with Random Forest.

Multi-Layer Perceptron (MLP) [15] is a type of feedforward neural network that consists of at least three layers: input, output and hidden layers. However, it can contain more than one hidden layer. This algorithm is a supervised learning algorithm that is made possible through backpropagation. This algorithm differs from a single-layer neural network in that it can be used in classification problems that are not linearly separable. An MLP can learn to draw convex lines around data points.

Voting Classifier [16] is another ensemble algorithm. However, it differs from AdaBoost and Gradient Boosting algorithms as it is not a “boosting” algorithm, i.e., it does not use same type models to fix their predictions and turn them into strong learners. A voting classifier typically uses multiple models of different types and combines their predictions into a final result using simple statistics. There are two types of voting: hard voting and soft voting. In hard voting, each base model has one vote for its predicted result and the ensemble model makes a decision based on the majority of the classifiers’ predictions, i.e., if the majority of the models voted that the result is class 0, then the ensemble’s prediction is also class 0. In soft voting, each model outputs a probability for its prediction instead of just a vote, and the ensemble model takes the average probability of the classifiers for each class and makes a prediction based on that average.

1.2. Related Work

There are several approaches in the literature targeting the prediction of TF binding sites. In [17], a tool was developed named “DRAF” for the prediction of transcription factor binding sites (TFBSs). The tool improves prediction accuracy of previous models that were based on position weight matrices (PWMs). It combines features from Transcription Factor Binding Sites (TFBS) sequences and physicochemical properties of TF DNA binding into classification algorithms of machine learning. Specifically, it uses Random Forest to make predictions. The authors tested their tools against other classification algorithms, namely neural network, Support Vector Machine (SVM) and Gaussian mixture regression; however, Random Forest provided the best accuracy results. In [18], a method was developed to predict TFBS using three features: nucleotide composition, nucleotides distribution and the transition between nucleotides. This method was implemented using two SVM classification models; each one was tested on a different dataset. The accuracies of the model were 81.84% and 82.27%. Moreover, a back propagation neural network was trained to classify the SP1 TFBS on human chromosome1 [19]. The proposed neural network consisted of two hidden layers. The input layer consisted of twenty-eight neurons and the output layer consisted of two neurons to predict whether the input is a binding site or not. The authors compared the results of their trained neural network with those of other classification models, namely the SVM, Linear Discriminant Analysis (LDA) and K-Nearest Neighbors (KNN) on the same dataset. It was shown that the neural network outperformed the other classification models with an accuracy of 84.4%. The work in [20] also presents a fragment-based prediction method, which splits a binding sequence into overlapping pentamers (5 base pairs) to calculate interaction energy. Their algorithm shows improved efficiency and accuracy, especially for long binding sites. In [21], a deep learning model combines both a Multi-Scale Convolutional Network and Long Short-Term Memory Network (MCNN-LSTM). They also proposed a new encoding scheme to represent nucleotide positions. The method presented was tested on several datasets, and the accuracy reached 80%. Zhou et al. use CNNs in [22] to present a multi-task learning framework. Their method is mainly concerned with unlabeled transcription factor binding sites data. The performance of the proposed method shows a 5.7% performance improvement over other compared methods in terms of the Area Under Curve (AUC). Phy-

loReg was presented in [23]. PhyloReg is a semi-supervised learning method used to better train earlier deep learning models. The approach presents increases in prediction accuracy. In [24], the attention mechanism in deep learning is employed to develop DeepGRN. The results obtained show that DeepGRN achieves higher unified scores in 6 out of 13 targets than the compared methods. In addition, the importance of histone modifications and chromatin accessibility in the identification of TFBS along with DNA sequence was studied in [25]. A Convolutional Neural Network (CNN) model was developed to combine these new features for cell-specific TFBSs prediction. The developed model was compared with other classification techniques: Logistic Regression, KNN, Random Forest and gradient boosted regression trees. Combining the three features along with CNN produced the best results.

In this paper, multiple machine learning classification algorithms were used to enhance the accurate prediction of TF binding sites of SP1 on human chromosome 1. In addition, voting and boosting algorithms were proposed to enhance the prediction performance. The results obtained in this paper show an enhanced performance for predicting TF binding sites, with an AUC equals 0.97 using KNN and 0.95 using the proposed voting technique. Moreover, the accuracy obtained by both KNN and the proposed voting technique are comparable, with 92% using KNN, and 88.1% using voting.

The rest of the paper is organized as follows. Section 2 provides a description of the model used for classification in this paper. Section 3 describes the results obtained. This is then followed by Section 4, which offers a discussion of the obtained results and outlines future directions. Finally, Section 5 concludes the paper.

2. Materials and Methods

The problem of predicting binding sites is considered a classification problem. There are different types of classification algorithms in machine learning used in this paper. The following subsections describe the different stages employed in this study. The model starts with the dataset section, where the dataset is explained, followed by the classification stage, where different classifiers are employed to predict the TFBS of SP1 human chromosome 1. Finally, the model evaluation is explained.

2.1. Dataset

The TFBS dataset used is obtained from Kaggle.com. It contains 2400 records of SP1 transcription factor binding and non-binding sites. The protein encoded by SP1 TF is involved in immune responses, chromatin remodeling, cell growth and other cellular processes. Each record contains 28 features. Half of the records are of binding sites and the other half are of non-binding sites. The records are classified into two classes (i.e., 1 for a binding site and 0 for a non-binding site).

Each record is a DNA sequence of 14 nucleotides. Each nucleotide is then transformed into a 2-bit binary number based on a binary coding rule, transforming the 14 nucleotides into a 28 binary sequence that acts as the features. The binary coding rule is shown in Table 1.

For example, a record could be a sequence such as: “ATCCGTTTCCGGGT, binding site” which would be transformed into features such as: “0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1” (28 features and the resulting class).

Table 1. Binary Coding rules for Nucleotides.

Nucleotide	Binary Equivalent
A	00
T	01
C	10

G

11

2.2. Classification

This paper employs several classification methods to predict the SP1 binding site for human chromosome 1, namely Logistic Regression, Linear Discriminant Analysis (LDA), Naive Bayes, K-Nearest Neighbor (KNN), Decision Tree, Random Forest, Extra Tree, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Adaboosting and Gradient Boosting, in addition to the suggested voting schemes. All of the models used for classification randomly divide the dataset into 80% training data and 20% testing data.

All of the experiments are implemented using Python's scikit-learn library [26]. Table 2 shows the specified parameters that are used in the proposed model for each tested classifier.

Several voting mechanisms were employed. The best performing schemes were SVM with Extra Trees, SVM with Logistic Regression, SVM with Logistic Regression and Extra Trees, and SVM with MLP. The highest performance achieved was for voting with SVM and MLP. The voting classifier is implemented by ensembling the two classifiers based on the results of the accuracy and AUC percentage. The voting model is implemented with a soft voting algorithm. It was found that the best results of the voting classifier were achieved by giving equal weight to each classifier in the ensemble. Moreover, Gradient Boosting was also proposed to enhance the classification accuracy. The following schemes were used: Random Forest and Naïve Bayes, Decision Trees with Naïve Bayes, Extra trees and Naïve Bayes, Logistic regression and Naïve Bayes, SVM and Logistic Regression. The boosting scheme is implemented by first developing a voting classifier and then using it as the base for the Gradient Boosting algorithm.

Table 2. Parameters Used for Each Classifier.

Classifier	Parameters
Logistic Regression	Default parameters were used
LDA	Default parameters were used
KNN	It was tested with K = 3
Decision Trees	Default parameters were used
Naïve Bayes	The multinomial naïve bayes was used
SVM	A grid search was used to find the best combination of parameters which turned out to be the "rbf" kernel with C = 10 and gamma = 0.5. The grid search searched through four kernels (i.e., rbf, linear, sigmoid and poly) with different ranges for C, gamma and degree
AdaBoost	50 estimators were used
GradientBoosting	Default parameters were used
Random Forest	100 estimators were used with maximum depth of 10
Extra Trees	100 estimators were used
MLP	It was exhaustively tested with the numbers of hidden layers and neurons ranging from 1 to 5. The best results were obtained with 3 hidden layers and 3 neurons in each layer
Voting	This classifier was tested in the end by ensembling four of the best resulted previous classifiers

2.3. Model Evaluation

The performance metrics used to analyze the proposed model are the accuracy rate, sensitivity, specificity and area under the curve (AUC) of precision-recall. Accuracy is the measure of how accurate the predictions are of the model, and it is calculated as shown in Equation (1). Sensitivity (also known as recall) is the measure of actual positives that are correctly identified, while specificity is the measure of actual negatives that are correctly identified. Sensitivity and specificity are calculated as shown in Equations (2) and (3), respectively. Precision is the measure of the relevancy of the results, and it is calculated as shown in Equation (4).

$$\text{Accuracy} = \frac{\text{truepositive} + \text{truenegative}}{\text{totalpopulation}} \quad (1)$$

$$\text{Sensitivity(recall)} = \frac{\text{truepositive}}{\text{truepositive} + \text{falsenegative}} \quad (2)$$

$$\text{Specificity} = \frac{\text{truenegative}}{\text{truenegative} + \text{falsepositive}} \quad (3)$$

$$\text{Precision} = \frac{\text{truepositive}}{\text{truepositive} + \text{falsepositive}} \quad (4)$$

3. Results

This section describes the experimental environment used for implementation, followed by the experimental results obtained.

3.1. Experimental Environment

All of the experiments are implemented using Python's scikit-learn library [26], namely Python 3.8 scikit-learn library. The machine used runs a Mac OS, with a processor of 2.3 GHz and 8-Core Intel Core i9, in addition to a memory of 16 GB 2667 MHz DDR4.

3.2. Experimental Results

Different classifiers along with the proposed voting and boosting methods are experimented with, and Table 3 shows the results of all the classifiers.

As seen in Table 3, the highest accuracy obtained for the prediction of transcription factor binding sites of SP1 on human chromosome1 is 92% using KNN and 88.1% using voting. On the other hand, the sensitivity is recorded highest for SVM and KNN, 92.1% and 90.3%, respectively. Specificity confirms the outperformance of KNN and voting, with 93.6% and 84.3%, respectively. Finally, the AUC is 97% for KNN and 95% for SVM and voting. The results show the high performance of KNN, SVM and voting. These results outperform the work found in the literature, which recorded the highest accuracy of 84.4%. The voting classifier is implemented by ensembling two classifiers based on the results of the accuracy and AUC percentage. The ensembled classifiers are the SVM and MLP classifiers. The voting model was implemented with a soft voting algorithm. It was found that the best results of the voting classifier were achieved by giving equal weight to each classifier in the ensemble.

Several voting techniques were attempted. SVM with Extra Trees achieved an accuracy of 87%, with sensitivity, specificity and AUC of 91.4%, 86.2% and 95%, respectively. SVM with Logistic Regression achieved an accuracy of 87.9%, with sensitivity, specificity and AUC of 88.5%, 87.3% and 95%, respectively. SVM with Logistic Regression and Extra Trees achieved an accuracy of 86.6%, with sensitivity, specificity and AUC of 88.1%, 85.1% and 95%, respectively. SVM with MLP outperformed similar models and achieved an ac-

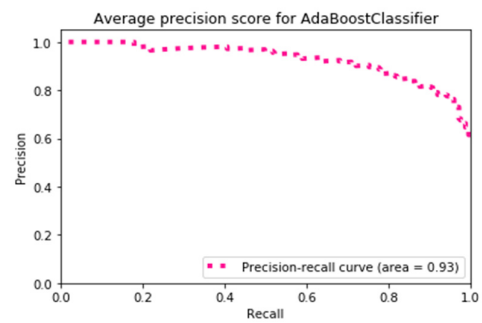
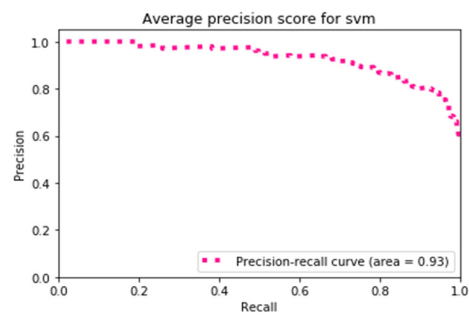
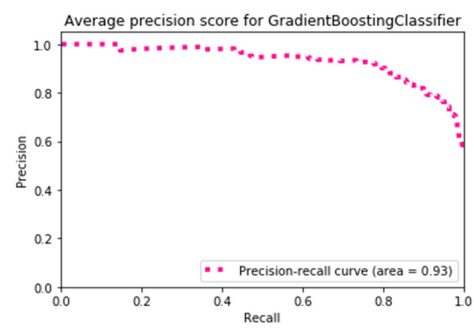
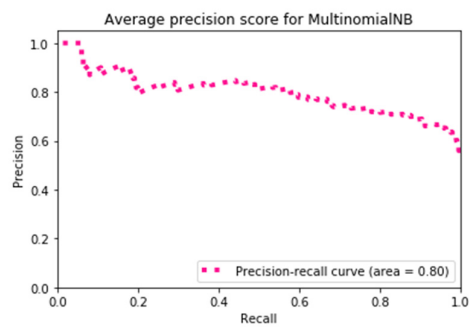
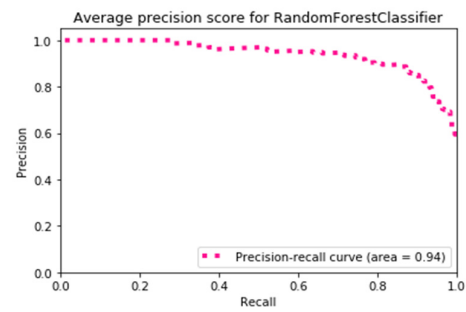
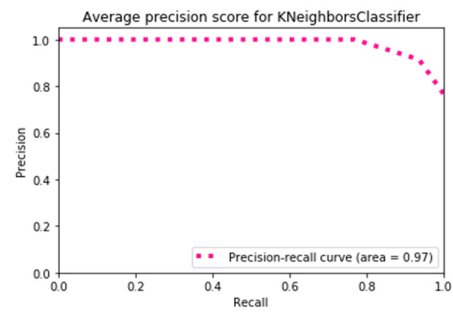
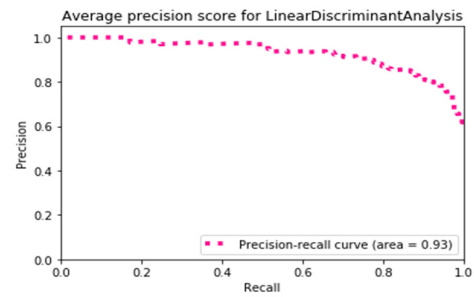
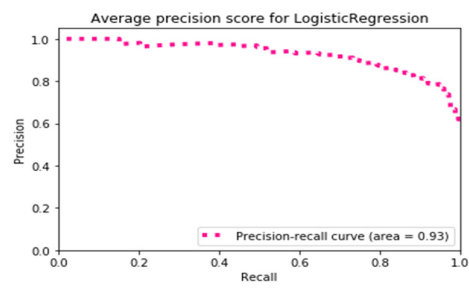
curacy of 88.1%, with sensitivity, specificity and AUC of 88.5%, 87.1% and 95%, respectively. These results show that the voting proposed enhances the accuracy performance of the sole classifier.

Moreover, boosting was also proposed to enhance the classification accuracy, where the developed voting scheme was used as a base learner for gradient boosting. Random Forest and Naïve Bayes achieved an accuracy score of 85.8%, sensitivity of 85.6%, specificity of 86% and AUC of 94%. Decision Trees with Naïve Bayes obtained 82.5%, 83.1%, 81.7% and 92%, for accuracy, sensitivity, specificity and AUC, respectively. Extra Trees and Naïve Bayes obtained 86%, 86.6%, 85.1% and 94%, for accuracy, sensitivity, specificity and AUC, respectively. Logistic Regression and Naïve Bayes obtained 84.7%, 86.4%, 83% and 94%, for accuracy, sensitivity, specificity and AUC, respectively. SVM and Logistic Regression obtained 87%, 88.9%, 85.1% and 95%, for accuracy, sensitivity, specificity and AUC, respectively.

Table 3 shows the best performing classifier in voting and in boosting. In the boosting classifier proposed, the base classifier used is obtained from voting between the proposed machine learning techniques. Figure 1 shows the precision recall curve of the proposed methods to show the tradeoff between precision and recall, where the KNN and proposed voting and boosting show an area of 0.97 and 0.95, consecutively. This confirms the high performance of KNN and the proposed voting scheme.

Table 3. Results of Each Classifier.

Classifier	Accuracy%	Sensitivity%	Specificity%	AUC%
Logistic Regression	83.7	84.2	83.2	93
LDA	83.5	84.2	82.8	93
KNN	92	90.3	93.6	97
Decision Trees	75.2	73.3	76.8	73
Naïve Bayes	72.5	67.2	77.2	80
SVM	86.2	92.1	80	95
AdaBoost	83.5	84.2	82.8	93
GradientBoosting	84.5	87.7	81.6	93
Random forest	86.2	88.2	84	94
Extra Trees	86.8	89.9	84	94
MLP	83	86.8	81.2	93
Boosting (Logistic Regression+SVM)	87	88.9	85.1	95
Proposed Voting (SVM+MLP)	88.1	88.5	87.7	95



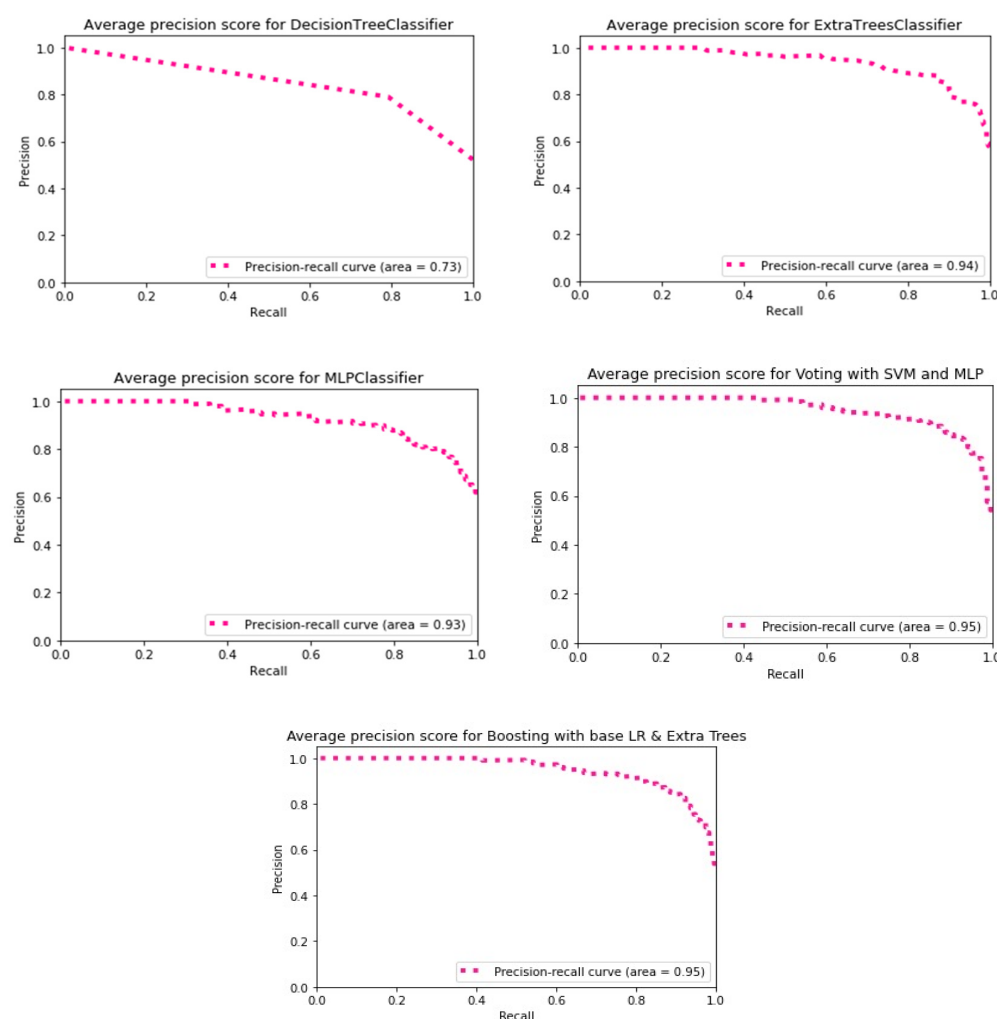


Figure 1. Precision-Recall Curve for all Classifiers.

4. Discussion

Transcription factors have a major role in the identification of the TFBSs. The identification of TFBS should lead to the understanding of transcriptional gene regulation. This is considered to be one of the challenges in computational biology as not all of the TFBS have been identified yet.

In this paper, the prediction of transcription factor binding sites of SP1 on Human Chromosome1 is carried out using eleven different classification models, in addition to a proposed voting technique and a boosting technique. The performance of the different classification techniques was measured by calculating their accuracy, specificity, sensitivity and AUC of precision-recall. The KNN, SVM, Extra Trees and Random Forest produced the best results. Their accuracies were 92%, 86.2%, 86.8% and 86.8%, respectively, while their AUCs were 97%, 95%, 94% and 94%, respectively.

The proposed suggested model ensembles two classification models, namely SVM and MLP, into a voting algorithm. Voting was implemented using soft voting (i.e., the average probability of their results was taken as the ensemble model's final prediction). The final accuracy and AUC achieved for the proposed model were 88.1% and 95%, respectively. The results show that the KNN classification algorithm solely outperformed

the ensemble algorithm. The problem with KNN is the dependence of performance accuracy and the quality of data, and with large datasets, such as transcription factor binding site data, the performance decreases and requires high memory. Moreover, KNN requires data to be scaled with all features being relevant to the prediction process. The proposed voting technique achieves a comparable accuracy to KNN and is more suitable for high dimensional data, with lots of possible noise as the case with the transcription factor binding sites data. It is known that bioinformatics data are inherently noisy, and machine learning algorithms are sensitive to this noise. Noise in the data arises from the laboratory techniques used, which are prone to human errors [27]. This noise can be overcome by multiple attempts of the laboratory experiments to ensure that the observations obtained are correct.

Moreover, the highest accuracy recorded for KNN is 92%, while it is 88.1% for the proposed voting algorithm. These results outperform the work in the literature which reported the accuracy. However, although the work done is performed on different datasets, they all target the TFBS problem. In [18], SVM was used and obtained 81.84% and 82.27% on two datasets. Moreover, neural networks achieved an accuracy of 84.4% in [19], while deep learning reached only an accuracy of 80% in [21]. This shows that the proposed work achieves promising results in the classification of TFBS, specifically, SP1 on human chromosome 1.

Furthermore, the dataset used in this study contained labeled sequences all of the same length, and all sequences were encoded. However, in the case of having missing labels or different lengths sequences, preprocessing methods can be employed. Regarding the issue of different lengths, padding of sequences could be done to unify the lengths before performing classification, where a dummy nucleotide value is used to append shorter sequences, then the sequences can go through the encoding stage. For the missing labels, sequences could be either deleted, or the missing values could be imputed.

This work can be further experimented upon using a larger dataset, as well as other ensemble algorithms. Moreover, the running time performance of the algorithms could be tested against different datasets.

5. Conclusions

In this paper, a prediction of transcription factor binding sites of SP1 on Human Chromosome1 was carried out. The paper employed several different classification models in addition to a proposed voting technique and boosting technique. The proposed suggested model ensembles two classification models, namely, SVM and MLP, into a voting algorithm. The accuracy and AUC achieved for the proposed model were 88.1% and 95%, respectively, whereas the accuracy for KNN was 92% and the AUC was 97%. The results obtained show that the KNN outperforms all other methods. However, the proposed voting technique obtains comparable results while overcoming drawbacks of other methods.

Author Contributions: Conceptualization, M.M.M., N.A.B. and A.Y.; methodology, M.M.M., N.A.B. and A.Y.; software, M.M.M.; validation, M.M.M., N.A.B. and A.Y.; formal analysis, M.M.M., N.A.B. and A.Y.; investigation, M.M.M., N.A.B. and A.Y.; resources, M.M.M.; writing—original draft preparation, M.M.M., N.A.B. and A.Y.; writing—review and editing, M.M.M., N.A.B. and A.Y.; visualization, M.M.M.; supervision, A.Y.; project administration, N.A.B. and A.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this study is available on Kaggle.com through the link: <https://www.kaggle.com/hobako1993/sp1-factor-binding-sites-on-chromosome1>, last accessed 15/5/2021. Moreover, the Python code for the model used is available upon request from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Alberts, B.; Johnson, A.; Lewis, J.; et al. From DNA to RNA. In *Molecular Biology of the Cell*, 4th ed.; Garland Science: New York, NY, USA, 2002.
- Lee, T.I.; Young, R.A. Transcription of Eukaryotic Protein-Coding Genes. *Annu. Rev. Genet.* **2000**, *34*, 77–137.
- Nikolov, D.B.; Burley, S.K. RNA polymerase II transcription initiation: A structural view. *Proc. Natl. Acad. Sci. USA* **1997**, *94*, 15–22.
- Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32.
- Zhang, H. The Optimality of Naive Bayes. In Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, FL, USA, 12–14 May 2004; p. 2, FLAIRS.
- Alzubi, J.; Nayyar, A.; Kumar, A. Machine Learning from Theory to Algorithms: An Overview. *J. Phys. Conf. Ser.* **2018**, *1142*, 1–15.
- Mitchell, T.M. Machine Learning. In *WCB McGraw Hill*; 2013, ISBN 13:978-1-25-909695-2/10: 1-25-909695-5.
- Mayr, A.; Binder, H.; Gefeller, O.; Schmid, M. The Evolution of Boosting Algorithms From Machine Learning to Statistical Modelling. *Methods Inf. Med.* **2014**, *53*.
- Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232.
- Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42.
- Murtagh, F. Multilayer perceptrons for classification and regression. *Neurocomputing* **1991**, *2*, 183–197.
- Kim, J.; Choi, S. Automated Machine Learning for Soft Voting in an Ensemble of Tree-based Classifiers. In Proceedings of the International Workshop on Automatic Machine Learning at ICML/IJCAI-ECAI, 2018.
- Khamis, A.M.; Motwalli, O.; Oliva, R.; Jankovic, B.R.; Medvedeva, Y.A.; Ashoor, H.; Essack, M.; Gao, X.; Bajic, V.B. A novel method for improved accuracy of transcription factor binding site prediction. *Nucleic Acids Res.* **2018**, *46*, e72.
- Lee, W.; Park, B.; Han, K. Sequence-Based Prediction of Putative Transcription Factor Binding Sites in DNA Sequences of Any Length. *IEEE ACM Trans. Comput. Biol. Bioinform.* **2018**, *15*, 1461–1469.
- Banki-Koshki, H.; Seyyedsalehi, S.A.; Zare-Mirakabad, F. Transcription factor binding sites identification on human genome using an artificial neural network. *Iran. Conf. Electr. Eng. ICEE* **2017**, 14–17.
- Farrel, A.; Guo, J.T. An efficient algorithm for improving structure-based prediction of transcription factor binding sites. *BMC Bioinform.* **2017**, *18*, 342.
- Bao, X.R.; Zhu, Y.H.; Yu, D.J. DeepTF: Accurate Prediction of Transcription Factor Binding Sites by Combining Multi-scale Convolution and Long Short-Term Memory Neural Network. *Lect. Notes Comput. Sci. Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.* **2019**, *11936 LNCS*, 126–138.
- Zhou, J.; Lu, Q.; Gui, L.; Xu, R.; Long, Y.; Wang, H. MTTFSite: Cross-cell type TF binding site prediction by using multi-task learning. *Bioinformatics* **2019**, *35*, 5067–5077.
- Ahsan, F.; Drouin, A.; Laviolette, F.; Precup, D.; Blanchette, M. Phylogenetic Manifold Regularization: A semi-supervised approach to predict transcription factor binding sites. In Proceedings of 2020 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2020, Seoul, Korea, 16–20 December 2020; Volume 9313437, pp. 62–66.
- Chen, C.; Hou, J.; Shi, X.; Yang, H.; Birchler, J.A.; Cheng, J. DeepGRN: Prediction of transcription factor binding site across cell-types using attention-based deep neural networks. *BMC Bioinform.* **2021**, *22*, 38.
- Jing, F.; Zhang, S.; Cao, Z.; Zhang, S. An integrative framework for combining sequence and epigenomic data to predict transcription factor binding sites using deep learning. *IEEE ACM Trans. Comput. Biol. Bioinform.* **2021**, *18*, 355–364.
- Peng, C.J.; Lee, K.L.; Ingersoll, G.M. An Introduction to Logistic Regression Analysis and Reporting. *J. Educ. Res.* **2002**, *96*, 3–14.
- Tharwat, A.; Gaber, T.; Ibrahim, A.; Hassanien, A. Linear discriminant analysis: A detailed tutorial. *Ai Commun.* **2017**, *30*, 169–190.
- Durga, S.N.; Rani, K.U. A Perspective Overview on Machine Learning Algorithms. In *Advances in Computational and Bio-Engineering*; Springer: Cham, Germany, 2020; p. 15.
- Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership inference attacks against machine learning models. In Proceedings of the Security and Privacy (SP), IEEE Symposium, 2017; pp. 3–18.
- Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *JMLR* **2011**, *12*, 2825–2830.
- Sloutsky, R.; Jimenez, N.; Swamidass, S.J.; Naegle, K.M. Accounting for noise when clustering biological data. *Brief. Bioinform.* **2013**, *14*, 423–436.