algorithms

MDPI

Article

DrugFinder: Druggable Protein Identification Model Based on Pre-Trained Models and Evolutionary Information

Mu Zhang, Fengqiang Wan and Taigang Liu *

College of Information Technology, Shanghai Ocean University, Shanghai 201306, China; 2052435@st.shou.edu.cn (M.Z.); 2052439@st.shou.edu.cn (F.W.)
* Correspondence: tgliu@shou.edu.cn

Abstract: The identification of druggable proteins has always been the core of drug development. Traditional structure-based identification methods are time-consuming and costly. As a result, more and more researchers have shifted their attention to sequence-based methods for identifying druggable proteins. We propose a sequence-based druggable protein identification model called DrugFinder. The model extracts the features from the embedding output of the pre-trained protein model Prot_T5_Xl_Uniref50 (T5) and the evolutionary information of the position-specific scoring matrix (PSSM). Afterwards, to remove redundant features and improve model performance, we used the random forest (RF) method to select features, and the selected features were trained and tested on multiple different machine learning classifiers, including support vector machines (SVM), RF, naive Bayes (NB), extreme gradient boosting (XGB), and k-nearest neighbors (KNN). Among these classifiers, the XGB model achieved the best results. DrugFinder reached an accuracy of 94.98%, sensitivity of 96.33% and specificity of 96.83% on the independent test set, which is much better than the results from existing identification methods. Our model also performed well on another additional test set related to tumors, achieving an accuracy of 88.71% and precision of 93.72%. This further demonstrates the strong generalization capability of the model.

Keywords: druggable protein; transformer-based models; machine learning; feature extraction

check for updates

Citation: Zhang, M.; Wan, F.; Liu, T. DrugFinder: Druggable Protein Identification Model Based on Pre-Trained Models and Evolutionary Information. *Algorithms* **2023**, *16*, 263. https://doi.org/10.3390/a16060263

Academic Editor: Louxin Zhang

Received: 28 April 2023
Revised: 20 May 2023
Accepted: 23 May 2023
Published: 25 May 2023

1. Introduction

Druggability, a fundamental concept in drug discovery, pertains to the capacity of a biological target to exhibit a strong binding affinity towards therapeutic drugs [1]. Proteins, being essential components and regulators of critical physiological processes in the human body, represent a significant reservoir of bio-druggable targets with substantial potential for therapeutic interventions. However, it is important to recognize that not all proteins possess the necessary attributes to effectively serve as drug targets. The current landscape of drug development programs is confronted by a limited pool of known druggable proteins, encompassing only approximately 2% of the entire human proteome [2,3]. This scarcity of druggable proteins poses notable constraints, impeding the discovery and advancement of novel drugs. It constrains the range of potential therapeutic targets, hindering progress towards the development of innovative treatment options. Consequently, there exists a compelling imperative to identify and comprehensively characterize additional druggable proteins. The expansion of the repertoire of viable therapeutic targets assumes critical importance in surmounting the limitations imposed by this scarcity. By widening the spectrum of druggable protein targets, researchers can unlock novel avenues for drug discovery, thereby fostering opportunities for innovative therapeutic interventions spanning diverse disease contexts. Therefore, there is an urgent need to discover more druggable proteins [4].

The traditional approach in biology involves analyzing the physical and chemical properties of proteins in a laboratory to identify druggable proteins [5,6]. This provides limited help in drug development. However, with the completion of the Human Genome

Project [7], protein sequence data became more readily available than structural data. The large amount of protein sequence data also benefits most machine learning algorithms [8]. Therefore, many researchers have proposed models and computational methods for the identification of druggable proteins based on protein sequence information. Yu et al. have achieved good results on both the fivefold cross-test and the external independent dataset based on the support vector machine and random forest [9]. Chen et al. used support vector machines for the identification and prediction of druggable proteins using a combination of three features: sequence, structure and subcellular localization, with an accuracy of up to 85.13% [10]. Jamali et al. innovatively utilized a neural network classifier to predict druggable protein targets using protein physicochemical properties, amino acids and dipeptides as the feature composition, achieving 92.1% accuracy on the cross-validation set [11]. Lin et al. extracted protein features by dipeptide composition, pseudo-amino acid composition and approximately simple sequences and then deployed a genetic algorithm for feature selection. Finally, an SVM classifier improved by bagging ensemble learning was used for prediction with an accuracy of 93.78% [12]. Later, with the continuous development of deep learning techniques, some deep learning methods were also used for the identification and prediction of druggable proteins. Yu et al. developed the first deep-learning-based classifier for druggable proteins by testing a combination of different deep learning methods and protein feature data, achieving 90% accuracy on the test dataset [13]. Sikander et al. improved the accuracy to 94.86% using combined features and the XGB classifier [14].

Most of the above studies use machine learning and deep learning methods such as neural networks, SVM and RF. In addition, in the field of bioinformatics, natural language processing (NLP) models are also attracting more and more attention from researchers [15]. NLP is a machine learning technique that enables computers to interpret, process and understand human language. Many NLP models have performed very impressively with text-related problems, such as LSTM-based models that use hidden layers to capture contextual information and transformer-based models that rely on self-attention mechanisms [16–18]. Pre-trained language models are also a class of NLP models, which differ from other NLP models in that they are trained on very large datasets in a self-supervised manner. Pre-trained language models have a very strong ability to extract features from text information [19], which has attracted the wide attention of bioinformaticians. As a result, some pre-trained models that treat biological sequences as sentences and use large-scale protein data for training have been developed [20].

In this study, we propose a new method for identifying druggable proteins based on pre-trained models and evolutionary information in the PSSM. The workflow is shown in Figure 1. Initially, the protein sequence is fed into the pre-trained model to generate embeddings, resulting in 1024-dimensional features for each protein. Then, the PSSM information of the protein sequence is captured using the Blast software package [21], and after processing the PSSM, KSB-PSSM (400 dimensions), DPC-PSSM (400 dimensions) and S-FPSSM (400 dimensions) vectors are obtained. Next, feature selection is performed on the three combined vectors and embedding information using a random forest algorithm. Finally, the results are input into a machine learning classifier, and performance testing is performed on an independent test set. This study compares three protein pre-trained models (Prot_T5_Xl_Uniref50(T5), Prot_Bert_BFD(BFD), SeqVec), ten feature selection lengths (from 100 to 2224 dimensions) and five classification algorithms (SVM, RF, NB, XGB, KNN). The best model comes from the combination of T5 and the XGB classifier at 1500 dimensions, with an accuracy of 94.98%.

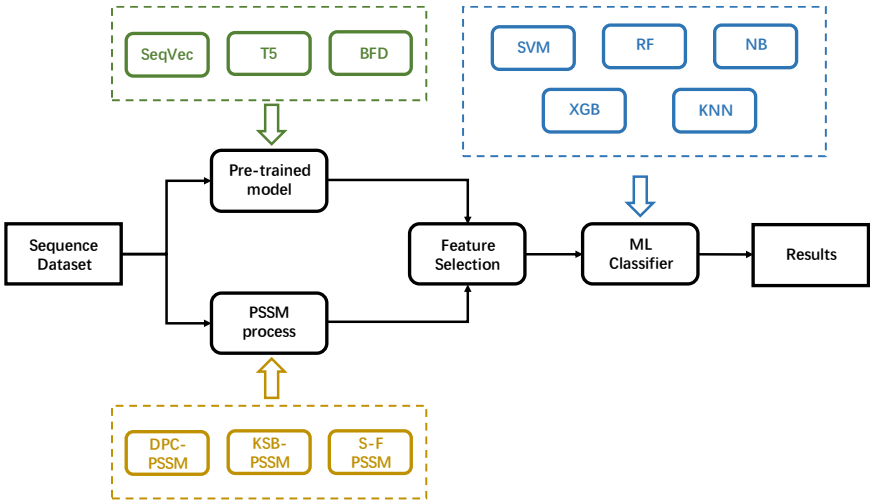

Figure 1. Flowchart of our druggable protein identification model.

2. Materials and Methods

2.1. Dataset

One of the datasets used in this paper is a gold standard dataset established by Jamali et al. [11]. It consists of 1224 positive samples (druggable proteins) and 1319 negative samples (non-druggable proteins). The positive samples are sourced from DrugBank and consist of druggable proteins corresponding to a variety of diseases, including leukemia, thrombocytopenia, angina pectoris and hypertension [2]. All positive samples with highly similar sequence content are removed to avoid their impact on the classifier. The negative samples come from the Swiss-Prot database, where druggable proteins that have been discovered and their related families are also removed. After these treatments, the probability of druggable proteins still existing in the negative sample dataset will be very low and have little impact on the prediction results.

To better demonstrate the model performance, we segmented about 20% of this dataset as an independent test set, consisting of 218 positive and 260 negative samples. The remaining 80% of the sequences were used for model training. Both the test set and the training set are sample size balanced, which also helps to improve the model's performance.

However, Jamali's dataset contains a large number of target proteins corresponding to different diseases. Therefore, using only this dataset may not adequately evaluate the model's performance in identifying specific disease targets. To address this issue, we have also proposed a small test set consisting of druggable proteins associated with tumors. This dataset comprises 64 positive samples and an equal number of negative samples. The positive data are derived from approved target proteins of various anticancer drugs in DrugBank, while the negative data are sourced from the Swiss-Prot database. This additional test set also eliminates highly similar samples and excludes druggable protein data from the negative samples.

2.2. Methods

2.2.1. Pre-Trained Models

Protein pre-trained models are generally based on NLP-related research and trained on large protein corpus databases, including Unrief50, UniRef100, Big Fantastic Database [22,23] and other non-redundant protein sequence databases. Models trained on different corpora often perform differently on protein classification and prediction tasks. This paper uses three protein pre-trained models based on different NLP models. They are T5, BFD and SeqVec. BFD is proposed based on the Bert model and trained on the Big Fantastic Database corpus [24]. T5 is proposed based on the T5 model and built on the Unrief50 corpus [25]. Bert and T5 are both large-scale pre-trained language models based on the transformer

model, proposed by Google. SeqVec is proposed based on the ELMo model and trained on the Unrief50 corpus [26,27].

We extract the embeddings of these pre-trained models as the next step in feature extraction. Each embedding typically consists of an array of size L * 1024, where L represents the length of the input sequence. To achieve a fixed-dimensional feature representation, various methods can be considered. One alternative method is padding, where shorter sequences are padded with zeros to match the length of the longest sequence in the dataset. However, padding can introduce challenges such as increased memory usage and noise in the feature representation. Another method is truncation, where longer sequences are truncated to a specific length. However, truncation can result in the loss of important information.

In comparison, averaging the embeddings offers advantages. It captures salient features while discarding less relevant information, resulting in a focused representation. Additionally, averaging enables the generation of fixed-dimensional feature vectors, ensuring compatibility and consistency in subsequent analysis and modeling.

By choosing averaging as the method for achieving a fixed-dimensional feature representation, we strike a balance between capturing essential information and maintaining a consistent representation. It provides a practical approach for handling varying sequence lengths in protein analysis, supporting accurate classification and prediction. Since the embedding is usually an array of size L * 1024, we also need to average the embedding to obtain a 1024-dimensional feature.

2.2.2. PSSM Process

The PSSM is a feature that incorporates evolutionary information from protein sequences [28]. This feature has been widely used in feature extraction steps for various protein-related experiments. The PSSM calculated by Blast is usually a matrix of size L * 20 [21], where L represents the length of the protein sequence. However, commonly used machine learning classifiers cannot handle such size-variable data very well. To better extract evolutionary information from protein sequences and adapt to machine learning models, many researchers have proposed PSSM-based features and corresponding calculation methods [29]. In this paper, we used three PSSM-based features to fully obtain the evolutionary information of protein sequences and improve the performance of the classifier.

DPC-PSSM: Dipeptide composition (DPC) is a method that can calculate the probability of consecutive amino acids in a protein sequence [30]. Unlike some other feature extraction methods, DPC not only extracts amino acid composition information but also contains a part of amino acid sequence information. The evolutionary information alone cannot fully characterize the entire protein. So we try to add the DPC information into the PSSM and generate an 800-dimensional (20 × 20) feature [31,32]. The features can be formulated as:

$$\mathrm{DPC-PSSM} = (D_{1,1}, \ldots, D_{1,20}, D_{2,1}, \ldots, D_{2,20}, \ldots, D_{20,1}, \ldots, D_{20,20})^T \quad (1)$$

where

$$D_{i,j} = \frac{1}{L-1} \sum_{k=1}^{L-1} D_{k,i} \times D_{k+1,j} (1 \le i, j \le 20) \quad (2)$$

KSB-PSSM: KSB-PSSM provides good feedback on the relationship between disjoint amino acids in protein sequences [33]. It is an extension of DPC-PSSM. In this method, the key parameter k indicates the amino acid distance at the time of calculation. In this paper, KSB-PSSM is calculated by setting k = 3. In this case, the transfer probability between amino acids split by two amino acids is calculated, and a 400-dimensional feature is generated. The calculation is as follows:

$$\mathrm{KSB-PSSM} = (K_{1,1}(k), \ldots, K_{1,20}(k), K_{2,1}(k), \ldots, K_{2,20}(k), \ldots, K_{20,1}(k), \ldots, K_{20,20}(k))^T \quad (3)$$

and

$$K_{i,j}(k) = \sum_{t=1}^{L-k} K_{t,i} \times K_{t+k,j} (1 \leq i, j \leq 20) \tag{4}$$

S-FPSSM: The S-FPSSM is a 400-dimensional feature derived from the FPSSM by row transformation [34]. In addition, the FPSSM is a matrix filtering the negative values in the original PSSM. The S-FPSSM is calculated as follows:

$$S_j^{(i)} = \sum_{k=1}^{L} fp_{k,j} \times \delta_{k,i} \tag{5}$$

which

$$\begin{cases} \delta_{k,i} = 1 \text{ if } r_k = a_i \\ \delta_{k,i} = 0 \text{ otherwise} \end{cases} (1 \leq i, j \leq 20) \tag{6}$$

In the above equation, $fp_{k,j}$ denotes the element in the kth row and jth column of the FPSSM, r_k denotes the kth amino acid in the original protein sequence, and a_i denotes the amino acid ranked at position i in the PSSM (20 positions in total).

2.2.3. Feature Selection

Not all features extracted from protein sequences using the above methods are valid. Especially in the case of multiple feature combinations, many features carry redundant information. Overly long features can burden the classifier and affect the model's performance. Therefore, this paper utilizes the random forest to eliminate the redundant features [35]. This method first obtains classification accuracy based on the original features. Then, it randomly changes the value of a certain feature and obtains new accuracy from the modified feature [36]. The difference in accuracy acquired from two calculations is processed and used as a ranking indicator of importance. The larger the difference, the more important the altered feature is considered.

2.2.4. Machine Learning Classifier

In this paper, we compare the classification performance of various machine learning classifiers on selected features, including SVM, RF, NB, XGBoost and KNN [37–40]. After optimizing the parameters of these models, they are trained on the same set of features. We use the Scikit-learn library in Python to test the performance of the five algorithms [41], and the results show that XGBoost classifier performed best on this type of problem.

2.2.5. Performance Evaluation

Choosing appropriate performance evaluation metrics is essential for model development. True positives (TP), false negatives (FN), true negatives (TN) and false positives (FP) are the most basic model evaluation standards. Among them, TP and TN represent samples that are true positives or negatives and are correctly classified by the model, while FN and FP represent positive or negative samples that are incorrectly classified by the model. Based on these four metrics, we derived the following six commonly used metrics to evaluate the performance of the model in this paper:

Accuracy (Acc): this metric measures the proportion of correctly classified instances out of all the instances in the dataset.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

Precision (Pre): this metric measures the proportion of true positives out of all the instances that were predicted as positive.

$$Pre = \frac{TP}{TP + FP} \tag{8}$$

Sensitivity (Sen): this metric, also known as recall or true positive rate (TPR), measures the proportion of true positives out of all the instances that are actually positive.

$$\text{Sen} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

Specificity (Spe): this metric measures the proportion of true negatives out of all the instances that are actually negative.

$$\text{Spe} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{10}$$

F-score: this metric is a harmonic mean of precision and recall and is useful when both high precision and high recall are important.

$$\text{F} - \text{score} = \frac{2 \times \text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{11}$$

Matthews Correlation Coefficient (MCC): This metric is a correlation coefficient between the observed and predicted binary classifications. It ranges from -1 to 1, where 1 represents a perfect prediction, 0 represents a random prediction, and -1 represents a completely wrong prediction. It takes into account true positives, true negatives, false positives and false negatives.

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FN} \times \text{FP})}{\sqrt{(\text{TP} + \text{FN}) \times (\text{TN} + \text{FP}) \times (\text{TP} + \text{FP}) \times (\text{TN} + \text{FN})}} \tag{12}$$

In addition, we use receiver operating characteristic (ROC) and area under curve (AUC) to more intuitively demonstrate the model's ability to identify druggable proteins.

3. Results

3.1. Comparison of Pre-Trained Models

This paper uses three pre-trained models, which are T5, BFD and SeqVec. They both generate an embedding of length 1024. We tested the performance of these models using the SVM classifier without adding any other feature information. As shown in Table 1, in the test without adding other features, the SeqVec model performed the best. Its accuracy and precision were 90.17% and 87.67%, respectively, which was about 5% better than the worst-performing PBD model. Based on these results, we also found that using pre-trained model embeddings directly to identify and classify druggable proteins performs poorly.

Table 1. Performance comparison of different pre-trained models on independent test set.

Model	Dimension	Acc	Pre	Sen	Spe	F-Score	MCC
T5	1024D	0.8661	0.8438	0.867	0.8858	0.8552	0.7310
T5+PSSM	2224D	0.9100	0.9070	0.8945	0.9125	0.9007	0.8185
PBD	1024D	0.8473	0.8139	0.8634	0.8785	0.8374	0.6947
PBD+PSSM	2224D	0.8536	0.8083	0.8899	0.8991	0.8472	0.7102
SeqVec	1024D	0.9017	0.8767	0.9128	0.9243	0.8944	0.8031
SeqVec+PSSM	2224D	0.8723	0.8398	0.8899	0.9028	0.8641	0.7451

Afterwards, we incorporated PSSM-based features extracted from the sequences into the model. The added features consisted of DPC-PSSM, K-Separated-Bigrams-PSSM and S-FPSSM, each with a length of 400. Therefore, the length of the features increased from 1024 to 2224. After the addition of the newly extracted features, the model's performance changed a lot, with the specific data still displayed in Table 1. In the three models, the T5 model and PBD model showed some improvement in most of the indicators, with accuracy reaching 91% and 90.17%, respectively. However, the Acc, Pre, Sen and Spe indicators of

the SeqVec model were lower than the results before adding new features. This may be due to the excessive number of features, which caused the model to overfit on the training set.

3.2. Feature Selection

To enhance the generalization ability of the model and reduce the possibility of overfitting, we used a random forest to select features for the feature set with a length of 2224. To obtain the optimal feature length that maximizes the model's performance, we tested the feature sets with lengths of 2000, 1800, 1500, 1300, 1000, 800, 500, 300 and 100 after the feature selection. To form an effective comparison, we did not change any other parameters except for the length of the features. The results of the tests are presented in Table 2. Compared to the model without feature selection, the new model's accuracy has increased by up to 6%. In all feature dimension tests, the T5 model performs best with around 1500-dimensional feature inputs. Its accuracy reaches 92.26%. As shown in Figure 2, we also found that the model's accuracy reaches its peak at around 1300 to 1800 dimensions and decreases on both sides of this range. This indicates that our feature selection is positive and effective in optimizing the model's performance.

Table 2. Model accuracy with different lengths of feature inputs.

Model	2000D	1800D	1500D	1300D	1000D	800D	500D	300D	100D
T5	0.9184	0.9163	0.9226	0.9079	0.9121	0.9016	0.8744	0.8682	0.8410
PBD	0.9100	0.9142	0.9142	0.9142	0.9184	0.9058	0.8723	0.8619	0.8431
SeqVec	0.9016	0.9100	0.9016	0.8924	0.8924	0.8835	0.8761	0.8647	0.8400

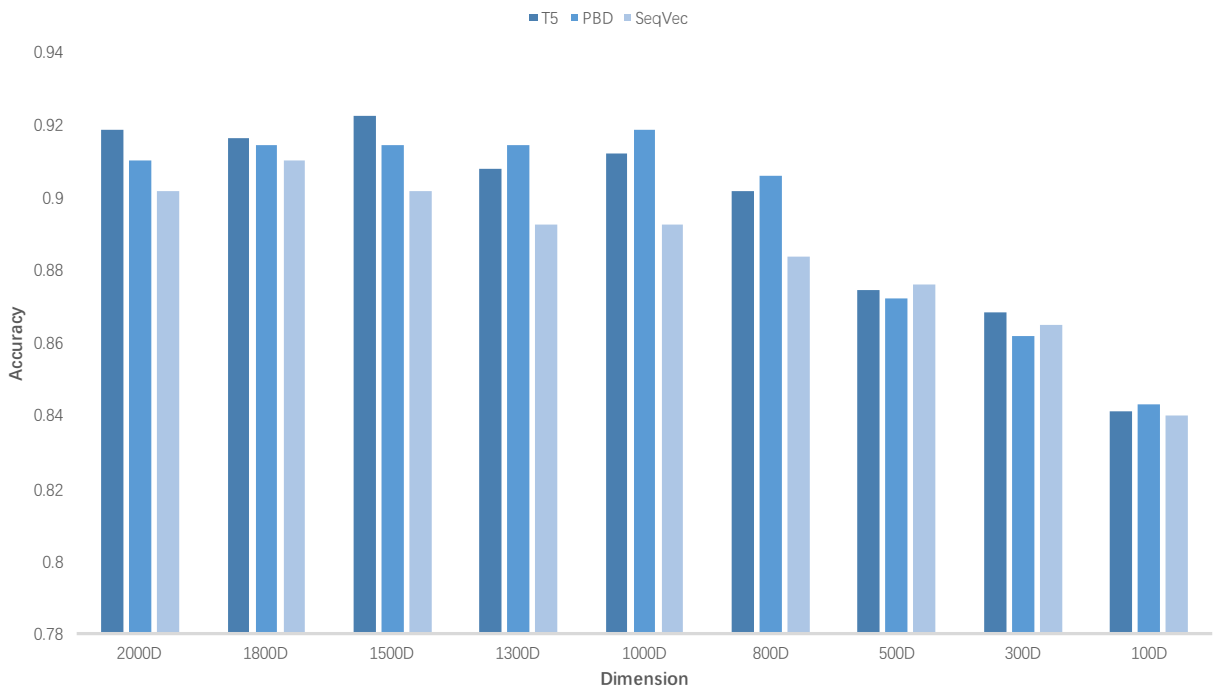

Figure 2. Visual demonstration of the impact of feature length on model performance.

3.3. Machine Learning Classifier

We compared the performance of five different machine learning classifiers on an independent test set. They are SVM, RF, NB, XGB and KNN. All the classifier calls were implemented through a Python library called Scikit-learn. Based on the results shown in Table 3, we found that T5+XGB performed the best, achieving an accuracy of 94.98% and precision of 92.92%. Sensitivity and specificity both exceeded 96%. XGB also performed very well on the results of the other two pre-trained models, achieving accuracies of 92.89%

and 94.56%, respectively. Besides the XGB classifier, the model using SVM also achieved an accuracy of around 92% on the independent test set. However, we also found that the NB, RF and KNN classifiers did not perform well, with accuracies below 90%, which is significantly lower than the results of XGB.

Table 3. Performance comparison of different classifiers on independent test set.

Model	Dimension	Classifier	Acc	Pre	Sen	Spe	F-Score	MCC
T5	1500D	SVM	0.9226	0.9330	0.8945	0.9145	0.9133	0.8441
		RF	0.8494	0.8042	0.8853	0.8950	0.8428	0.7018
		NB	0.8493	0.8349	0.8349	0.8615	0.8349	0.6964
		XGB	**0.9498**	**0.9292**	**0.9633**	**0.9683**	**0.9460**	**0.8996**
		KNN	0.8765	0.8597	0.8716	0.8911	0.8656	0.7516
PBD	1500D	SVM	0.9142	0.9078	0.9037	0.9195	0.9057	0.8271
		RF	0.8724	0.8398	0.8899	0.9028	0.8641	0.7440
		NB	0.8661	0.8598	0.8440	0.8712	0.8519	0.7298
		XGB	0.9289	0.9220	0.9220	0.9346	0.9220	0.8566
		KNN	0.8703	0.8482	0.8716	0.8898	0.8597	0.7394
SeqVec	1800D	SVM	0.9100	0.9035	0.9122	0.9045	0.9031	0.8032
		RF	0.8975	0.8658	0.9174	0.9271	0.8909	0.7956
		NB	0.8410	0.8859	0.7477	0.8129	0.8109	0.6827
		XGB	0.9456	0.9324	0.9495	0.9570	0.9409	0.8907
		KNN	0.8494	0.8202	0.8578	0.8760	0.8386	0.6981

On the other hand, we utilized ROC to evaluate the performance of these models. We calculated the AUC to compare the performance of the models in a more intuitive way. From Figure 3, we can see that the AUC values corresponding to XGB are 0.99, 0.98 and 0.99 for the three models, which are much better than the other classifiers. The SVM classifier also performs quite well, with an AUC value of 0.97.

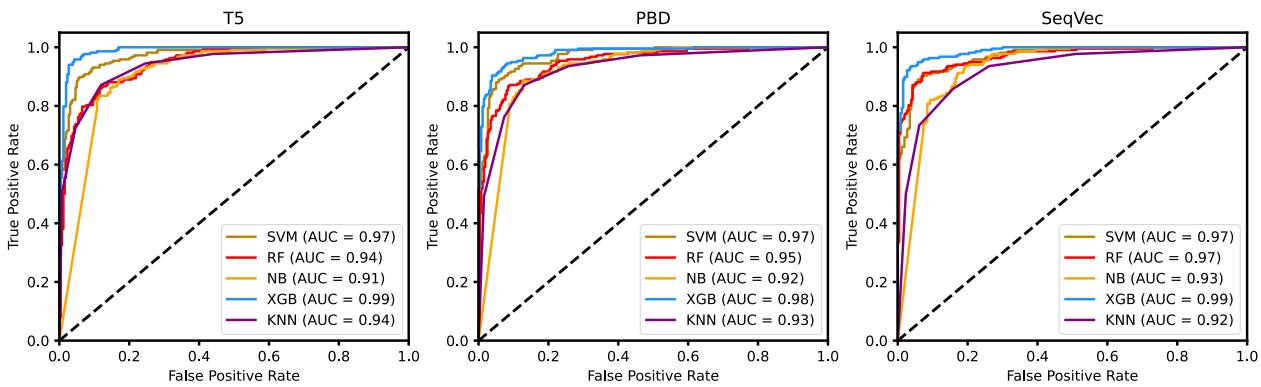

Figure 3. ROC curves of different machine learning classifiers.

3.4. Model Performance on Specific Disease Target Test Set

The model performance tests in the previous chapters were conducted on the dataset proposed by Jamali et al., which contains target proteins for multiple diseases. To better assess the DrugFinder's generalization ability for individual disease targets, we performed model testing on an additional test set consisting of druggable proteins related to tumors. The results showed that the model achieved an accuracy of 88.71%, precision of 93.72% and sensitivity of 81.68% on this test set. These results indicate that DrugFinder can maintain relatively high accuracy and precision in identifying druggable proteins corresponding to specific diseases. This will contribute to drug development in various disease domains.

Figure 4 provides a radar chart for a more intuitive comparison of the model's performance on the two different test sets. From the test data and radar chart, it can be observed that the model's performance metrics on the specific disease (tumor) target test set are

slightly lower than the metrics on Jamali's test set, which may be attributed to some differences in the data distribution between the two test sets. The tumor-related test set contains fewer sequences of druggable proteins, which could also introduce some randomness in the performance testing.

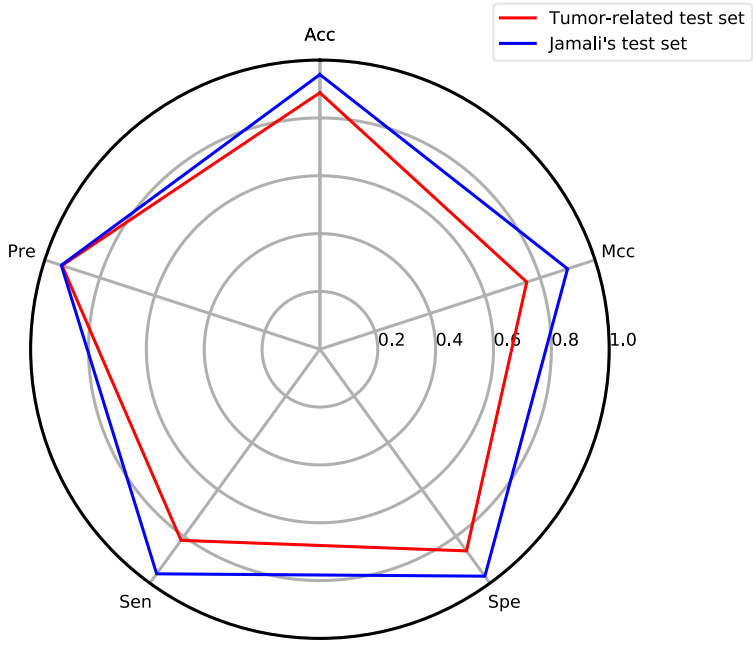

Figure 4. Model performance comparison on two different test sets.

3.5. Comparison with Other Models

To better demonstrate the effectiveness and superiority of the model, we compared it with other existing models for druggable protein identification. These models were also trained on the dataset proposed by Jamali et al. The comparison results are shown in Table 4, where all computational results are from the papers proposing these models. From Table 4, we discovered that our model has an accuracy similar to the previously best-performing method XGB-DrugPred, but the sensitivity and specificity are improved by 2.6% and 1.1%, respectively. In addition, our F-score and MCC values are also superior to the three existing models. This indicates that our model performs more sensitively and accurately in identifying and predicting drug–protein interactions.

Table 4. Comparison with other druggable protein identification models.

Model	Acc	Sen	Spe	F-Score	MCC
DrugMiner	0.9210	0.9280	0.9134	0.9241	0.8417
GA-Bagging-SVM	0.9378	0.9286	0.9445	0.9358	0.8781
XGB-DrugPred	0.9486	0.9375	0.9574	0.9417	0.8900
DrugFinder	0.9498	0.9633	0.9683	0.9460	0.8996

4. Discussion

Accurately identifying druggable proteins is a prerequisite for drug development. In recent years, with the development of related sequencing technologies [42,43], an increasing amount of protein sequence information has been obtained. Obtaining this information is much less costly than traditional laboratory methods. The large amount of protein sequence data makes it possible to identify druggable proteins through machine learning methods. In this context, we propose a druggable protein identification model based on pre-trained models and protein evolution information. We first used pre-trained models to extract the embeddings of proteins in the dataset and compared the extraction effects

of three pre-trained models. Subsequently, we used Blast to calculate the PSSM of the sequence. Since the size of the PSSM is usually L * 20, which is not conducive to subsequent classification, we further processed the PSSM to obtain a 1200-dimensional feature. This feature was combined with the output of the pre-trained model. In order to eliminate the influence of redundant features on the accuracy of the classifier, we also used RF to perform feature selection on the combined features. Finally, we compared five commonly used machine learning classifiers and obtained the best result on the XGB classifier: 94.98% accuracy, 96.33% sensitivity and 96.83% specificity. Our model was also tested on an additional dataset of druggable proteins related to tumors, achieving an accuracy of 88.71% and precision of 93.72%. Although these results are slightly lower than those obtained on Jamali's test set, they are still sufficient to demonstrate the model's generalization capability. From Table 4, it can be seen that compared with other models using the same dataset, this model has significantly improved the ability to identify druggable proteins. The proposed design and construction method of the druggable protein identification model in this paper also help with future drug development and related research fields.

Author Contributions: Conceptualization, M.Z. and F.W.; methodology, M.Z.; software, M.Z.; validation, M.Z. and T.L.; investigation, M.Z. and F.W.; resources, T.L.; data curation, M.Z.; writing—original draft preparation, M.Z.; writing—review and editing, T.L.; visualization, F.W.; supervision, T.L.; project administration, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used to support the findings of this study are freely available online at https://github.com/Melo-1017/DrugFinder (accessed on 19 May 2023).

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

References

1. Owens, J. Determining druggability. *Nat. Rev. Drug Discov.* **2007**, *6*, 187. [CrossRef]
2. Wishart, D.S.; Feunang, Y.D.; Guo, A.C.; Lo, E.J.; Marcu, A.; Grant, J.R.; Sajed, T.; Johnson, D.; Li, C.; Sayeeda, Z.; et al. DrugBank 5.0: A major update to the DrugBank database for 2018. *Nucleic Acids Res.* **2018**, *46*, D1074–D1082. [CrossRef] [PubMed]
3. Lacombe, D.; Butler-Smith, A.; Therasse, P.; Fumoleau, P.; Burtles, S.; Calvert, H.; Marsoni, S.; Sessa, C.; Verweij, J. Cancer drug development in Europe: A selection of new agents under development at the European Drug Development Network: NEW DRUGS. *Cancer Investig.* **2003**, *21*, 137–147. [CrossRef] [PubMed]
4. Lombardino, J.G.; Lowe, J.A. The role of the medicinal chemist in drug discovery—Then and now. *Nat. Rev. Drug Discov.* **2004**, *3*, 853–862. [CrossRef] [PubMed]
5. Roy, A. Challenges with risk mitigation in academic drug discovery: Finding the best solution. *Expert Opin. Drug Discov.* **2019**, *14*, 95–100. [CrossRef] [PubMed]
6. Zhang, L.C.; Kong, L. iRSpot-ADPM: Identify recombination spots by incorporating the associated dinucleotide product model into Chou's pseudo components. *J. Theor. Biol.* **2018**, *441*, 1–8. [CrossRef]
7. Dai, Y.F.; Zhao, X.M. A Survey on the Computational Approaches to Identify Drug Targets in the Postgenomic Era. *Biomed Res. Int.* **2015**, *2015*, 239654. [CrossRef]
8. Roh, Y.; Heo, G.; Whang, S.E. A Survey on Data Collection for Machine Learning: A Big Data-AI Integration Perspective. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 1328–1347. [CrossRef]
9. Yu, H.; Chen, J.X.; Xu, X.; Li, Y.; Zhao, H.H.; Fang, Y.P.; Li, X.X.; Zhou, W.; Wang, W.; Wang, Y.H. A Systematic Prediction of Multiple Drug-Target Interactions from Chemical, Genomic, and Pharmacological Data. *PLoS ONE* **2012**, *7*, e37608. [CrossRef]
10. Huang, C.; Zhang, R.J.; Chen, Z.Q.; Jiang, Y.S.A.; Shang, Z.W.; Sun, P.; Zhang, X.H.; Li, X. Predict potential drug targets from the ion channel proteins based on SVM. *J. Theor. Biol.* **2010**, *262*, 750–756. [CrossRef]
11. Jamali, A.A.; Ferdousi, R.; Razzaghi, S.; Li, J.Y.; Safdari, R.; Ebrahimie, E. DrugMiner: Comparative analysis of machine learning algorithms for prediction of potential druggable proteins. *Drug Discov. Today* **2016**, *21*, 718–724. [CrossRef] [PubMed]
12. Lin, J.; Chen, H.; Li, S.; Liu, Y.; Li, X.; Yu, B. Accurate prediction of potential druggable proteins based on genetic algorithm and Bagging-SVM ensemble classifier. *Artif. Intell. Med.* **2019**, *98*, 35–47. [CrossRef] [PubMed]
13. Yu, L.; Xue, L.; Liu, F.; Li, Y.; Jing, R.; Luo, J. The applications of deep learning algorithms on in silico druggable proteins identification. *J. Adv. Res.* **2022**, *41*, 219–231. [CrossRef]
14. Sikander, R.; Ghulam, A.; Ali, F. XGB-DrugPred: Computational prediction of druggable proteins using eXtreme gradient boosting and optimized features set. *Sci. Rep.* **2022**, *12*, 1–9. [CrossRef] [PubMed]
15. Chen, J.X.; Gu, Z.H.; Xu, Y.J.; Deng, M.H.; Lai, L.H.; Pei, J.F. QuoteTarget: A sequence-based transformer protein language model to identify potentially druggable protein targets. *Protein Sci.* **2023**, *32*, e4555. [CrossRef] [PubMed]

16. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
17. Wang, J.Y.; Hu, F.; Li, L. Deep Bi-directional Long Short-Term Memory Model for Short-Term Traffic Flow Prediction. In Proceedings of the International Conference on Neural Information Processing, ICONIP 2017, Guangzhou, China, 14–18 November 2017; Volume 10638, pp. 306–316.
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
19. Yang, S.; Feng, D.; Qiao, L.; Kan, Z.; Li, D. Exploring Pre-trained Language Models for Event Extraction and Generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics(ACL 2019), Florence, Italy, 28 July–2 August 2019; pp. 5284–5294.
20. Indriani, F.; Mahmudah, K.R.; Purnama, B.; Satou, K. ProtTrans-Glutar: Incorporating Features From Pre-trained Transformer-Based Models for Predicting Glutarylation Sites. *Front. Genet.* **2022**, *13*, 1201. [CrossRef]
21. Altschul, S.F.; Gish, W.; Miller, W.; Myers, E.W.; Lipman, D.J. Basic local alignment search tool. *J. Mol. Biol.* **1990**, *215*, 403–410. [CrossRef]
22. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
23. Tran, C.; Khadkikar, S.; Porollo, A. Survey of Protein Sequence Embedding Models. *Int. J. Mol. Sci.* **2023**, *24*, 3775. [CrossRef]
24. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K.; Assoc Computat, L. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Minneapolis, MN, USA, 3–5 June 2019; Volume 1, pp. 4171–4186.
25. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
26. Heinzinger, M.; Elnaggar, A.; Wang, Y.; Dallago, C.; Nechaev, D.; Matthes, F.; Rost, B. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinform.* **2019**, *20*, 1–17. [CrossRef] [PubMed]
27. Villegas-Morcillo, A.; Gomez, A.M.; Sanchez, V. An analysis of protein language model embeddings for fold prediction. *Brief. Bioinform.* **2022**, *23*, bbac142. [CrossRef] [PubMed]
28. Altschul, S.F.; Madden, T.L.; Schaffer, A.A.; Zhang, J.H.; Zhang, Z.; Miller, W.; Lipman, D.J. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* **1997**, *25*, 3389–3402. [CrossRef]
29. Wang, J.W.; Yang, B.J.; Revote, J.; Leier, A.; Marquez-Lago, T.T.; Webb, G.; Song, J.N.; Chou, K.C.; Lithgow, T. POSSUM: A bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics* **2017**, *33*, 2756–2758. [CrossRef] [PubMed]
30. Khan, A.; Majid, A.; Hayat, M. CE-PLoc: An ensemble classifier for predicting protein subcellular locations by fusing different modes of pseudo amino acid composition. *Comput. Biol. Chem.* **2011**, *35*, 218–229. [CrossRef]
31. Guruprasad, K.; Reddy, B.V.; Pandit, M.W. Correlation between stability of a protein and its dipeptide composition: A novel approach for predicting in vivo stability of a protein from its primary sequence. *Protein Eng.* **1990**, *4*, 155–161. [CrossRef]
32. Yu, B.; Lou, L.; Li, S.; Zhang, Y.; Qiu, W.; Wu, X.; Wang, M.; Tian, B. Prediction of protein structural class for low-similarity sequences using Chou's pseudo amino acid composition and wavelet denoising. *J. Mol. Graph. Model.* **2017**, *76*, 260–273. [CrossRef]
33. Saini, H.; Raicar, G.; Lal, S.; Dehzangi, I.; Imoto, S.; Sharma, A. Protein Fold Recognition Using Genetic Algorithm Optimized Voting Scheme and Profile Bigram. *J. Softw.* **2016**, *11*, 756–767. [CrossRef]
34. Zahiri, J.; Yaghoubi, O.; Mohammad-Noori, M.; Ebrahimpour, R.; Masoudi-Nejad, A. PPIevo: Protein-protein interaction prediction from PSSM based evolutionary information. *Genomics* **2013**, *102*, 237–242. [CrossRef]
35. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
36. Scornet, E. Random Forests and Kernel Methods. *IEEE Trans. Inf. Theory* **2016**, *62*, 1485–1500. [CrossRef]
37. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
38. Cutler, D.R.; Edwards, T.C.; Beard, K.H.; Cutler, A.; Hess, K.T. Random forests for classification in ecology. *Ecology* **2007**, *88*, 2783–2792. [CrossRef] [PubMed]
39. Chen, T.Q.; Guestrin, C.; Assoc Comp, M. XGBoost: A Scalable Tree Boosting System. In Proceedings of the KDD'16: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 26–29 August 2001; pp. 785–794.
40. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
41. 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. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

42. Han, H.; Nutiu, R.; Moffat, J.; Blencowe, B.J. SnapShot: High-Throughput Sequencing Applications. *Cell* **2011**, *146*, 1044–1046. [CrossRef]
43. Zhang, H.; Zheng, Y. Application of high-throughput sequencing technology in dairy product. *J. Chin. Inst. Food Sci. Technol.* **2015**, *15*, 1–7. [CrossRef]