applied sciences

MDPI

Article

Application of Machine Learning Algorithms for Prediction of Tumor T-Cell Immunogens

Stanislav Sotirov and Ivan Dimitrov *

Drug Design and Bioinformatics Lab, Faculty of Pharmacy, Medical University-Sofia, 1000 Sofia, Bulgaria
* Correspondence: idimitrov@pharmfac.mu-sofia.bg

Abstract: The identification and characterization of immunogenic tumor antigens are essential for cancer vaccine development. In light of the impracticality of isolating and evaluating each putative antigen individually, in silico prediction algorithms, particularly those utilizing machine learning (ML) approaches, play a pivotal role. These algorithms significantly reduce the experimental workload necessary for discovering vaccine candidates. In this study, we employed six supervised ML methods on a dataset comprising 212 experimentally validated human tumor peptide antigens and an equal number of non-antigenic human peptides to develop models for immunogenicity prediction. These methods encompassed k-nearest neighbor (*k*NN), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), support vector machine (SVM), random forest (RF), and extreme gradient boosting (XGBoost). The models underwent validation through internal cross-validation within 10 groups from the training set and were further assessed using an external test set. Remarkably, the *k*NN model demonstrated superior performance, recognizing 90% of the known immunogens in the test set. The RF model excelled in the identification of non-immunogens, accurately classifying 93% of them in the test set. The three top-performing ML models according to multiple evaluation metrics (SVM, RF, and XGBoost) are to be subsequently integrated into the new version of the VaxiJen server, facilitating tumor antigen prediction through a majority voting mechanism.

Keywords: cancer; immunogenicity; machine learning; in silico; bioinformatics

Citation: Sotirov, S.; Dimitrov, I. Application of Machine Learning Algorithms for Prediction of Tumor T-Cell Immunogens. *Appl. Sci.* **2024**, *14*, 4034. https://doi.org/10.3390/app14104034

Academic Editors: Glenn Hawe and Aidan Meade

Received: 15 March 2024
Revised: 6 May 2024
Accepted: 7 May 2024
Published: 9 May 2024

1. Introduction

1.1. The Immune Aspects of Peptide-Based Cancer Vaccines

The immune system plays a critical role in the detection and elimination of cancerous cells [1]. Immunogenicity refers to the capacity of a foreign biomolecule to induce a humoral and/or cell-mediated immune response within the host organism. Molecules that elicit an immune response leading to the formation of memory cells are classified as protective immunogens. In recent years, numerous immunotherapeutic approaches for cancer treatment have emerged, including immune checkpoint inhibitors, chimeric antigen receptor (CAR) T-cell therapy, and cancer vaccines.

Cancer cells are distinguished by the presence of specific molecules known as antigens. These antigens are presented on the surface of cancer cells and bound to major histocompatibility complex (MHC) molecules [2]. When this complex is recognized by the T-lymphocytes, an immune response is initiated, and the antigen becomes immunogenic. This recognition mechanism forms the basis for the immune system's ability to identify and target cancerous cells for destruction.

Peptide-based cancer vaccines have emerged as an important milestone in modern oncology research as the search for innovative cancer treatments continues to gain momentum. They operate by directing the immune system to recognize tumor antigens as foreign entities [3]. Despite the widely recognized safety and tolerability profiles in clinical contexts, none of these vaccines have yet secured approval from the drug agencies for human patient

use. Consequently, the realm of cancer vaccine research has garnered considerable scientific interest and investment. Substantial endeavors are now being directed towards optimizing the vaccine selection and production processes, with a central emphasis placed on the cautious selection of the most efficacious antigenic components. The main stage in the development of cancer vaccines lies in the strategy used for antigen selection [4]. In an ideal scenario, each prospective antigen would undergo isolation or synthesis, followed by an empirical assessment of its immunogenic properties. However, this approach is impeded by logistical challenges and practical constraints. Consequently, to streamline and rationalize the antigen discovery process, in silico prediction algorithms have emerged as innovative tools in recent years [5].

1.2. Computational Methods for Tumor T-Cell Antigen Prediction

In recent years, a multitude of methodologies has been devised for the prediction of immunogenicity in tumor antigens (Table 1). They can be conditionally defined into two groups, namely methods that rely on classical machine learning prediction algorithms and methods that implement a combination of other algorithms into complex biological pipelines.

1.2.1. Machine Learning Prediction Tools

TTAgP stands as a precise bioinformatic tool utilizing the random forest algorithm to predict antitumor peptides, presented in the context of MHC Class-I [6]. It is trained and validated based on several features of 922 peptides derived from the TANTIGEN dataset (http://cvc.dfci.harvard.edu/tadb/ (accessed on 16 February 2024)). The performance of this method was evaluated against other human tumor antigen prediction programs, demonstrating superior results. iTTCA-Hybrid stands as an automated, sequence-based predictor, meticulously designed for the identification of tumor MHC Class-I T cell antigens [7]. This predictive framework employs two machine learning models, Random Forest (RF), and Support Vector Machine (SVM) [8]. Leveraging a sophisticated approach, iTTCA-Hybrid integrates five distinct feature encoding strategies, encompassing amino acid composition, dipeptide composition, pseudo-amino acid composition, distribution of amino acid properties in sequences, and physicochemical properties derived from the AAindex [9]. Additionally, an oversampling approach, specifically the Synthetic Minority Over-sampling Technique (SMOTE), is incorporated to enhance the model's predictive performance [10]. TAP 1.0 is an immunoinformatic tool specifically tailored for the prediction of tumor T cell antigens [11]. In this work, 544 descriptors of chemical–physical properties extracted from the AAindex database [9] were calculated for a total of 1184 tumor and non-tumor antigens. Its development involved a meticulous assessment of 15 machine learning algorithms applied to the classification of tumor antigens. Notably, the Quadratic Discriminant Analysis (QDA) model demonstrated optimal equilibrium across various performance measures, positioning itself as the most adept choice within this evaluative framework. PSRTTCA is a recently developed algorithm aimed at enhancing the identification and characterization of tumor T-cell antigens [12]. This novel approach utilizes propensity score representation learning, building on the Scoring Card Method (SCM), which generates diverse sets of propensity scores for amino acids, dipeptides, and g-gap dipeptides extracted from sequences associated with tumor T-cell antigens. They are later fed to an RF meta-classifier and parameter optimization is performed. After that, the best feature vectors are selected based on their statistical performance and a final predictive model is constructed. Furthermore, PSRTTCA offers the flexibility to extend its application to classify various other protein and peptide functions based solely on their primary sequences.

VaxiJen stands as the pioneering server for the alignment-independent prediction of protective antigens [13]. It employs distinct positive and negative sets, each comprising 100 well-established antigens and 100 known non-antigens, respectively. Adopting a machine learning (ML) approach, the method represents each protein within the set as a string of z-scales [14], encapsulating the principal physicochemical properties of constituent

amino acid residues. These representations are subsequently transformed into uniform vectors through auto-cross covariance (*ACC*) [15]. The sets of vectors undergo a systematic analysis facilitated by the genetic algorithm (GA) [16], which serves to discern pertinent variables. Subsequently, partial least squares-based discriminant analysis (PLS-DA) [17] is applied to construct the final prediction model.

1.2.2. Other Prediction Tools

The Landscape of Effective Neoantigens Software (LENS) offers predictions for tumor-specific and tumor-associated antigens, considering diverse genomic alterations such as single nucleotide variants, insertions and deletions, fusion events, splice variants, cancer-testis antigens, overexpressed self-antigens, as well as viral and endogenous retroviral elements [18]. It is a pipeline of over two thousand separate tools for predicting the full suite of tumor antigens from genomics data. As an open-source project, it offers extendibility through both the introduction of new modules, tools, and reference datasets, along with refinement of its tumor antigen prioritization capabilities. OpenVax provides a computational workflow designed for the identification of somatic variants, prediction of neoantigens, and the curation of personalized cancer vaccine contents [19]. It performs alignment of the tumor and normal DNA samples as well as the tumor RNA. A prioritization of the identified somatic variants is made based on expression and predicted MHC Class-I binding affinity after multi-step processing via several computational programs. OpenVax has been used in three clinical trials assessing synthetic long peptides (NCT02721043, NCT03223103, NCT03359239). pVACtools support the identification of altered peptides arising from various mechanisms, including point mutations, in-frame and frameshift insertions and deletions, and gene fusions [20]. It is a pipeline consisting of several steps executed via a modular workflow featuring tools for neoantigen prediction from somatic alterations (pVACseq and pVACfuse), prioritization, and selection using a graphical web-based interface (pVACviz), and the design of DNA vector-based vaccines (pVACvector) and synthetic long peptide vaccines. The Nextflow NEOantigen prediction pipeline (nextNEOpi) represents a comprehensive and fully automated bioinformatic tool for predicting tumor neoantigens from raw DNA and RNA sequencing data [21]. It takes as input raw whole exome sequencing (WES) or whole genome sequencing (WGS) data from matched tumor–normal samples and, optionally, bulk-tumor RNA-seq data. After data preprocessing, nextNEOpi derives the germline and phased somatic mutations, copy number variants, tumor purity and ploidy, and subsequently selects high-confidence variants through majority voting. This method can predict both MHC Class-I and MHC Class-II neoantigens. TIminer (Tumor Immunology Miner) represents a user-friendly framework designed for the facilitation of integrative immunogenomics analyses, particularly focusing on tumor RNA-seq and mutational data at the level of individual patients or samples [22]. It integrates state-of-the-art bioinformatics tools to analyze single-sample RNA-seq data and somatic DNA mutations to characterize the tumor–immune interface including the following: genotyping of HLAs from NGS data; prediction of tumor neoantigens using mutation data and HLA types; characterization of tumor-infiltrating immune cells from bulk RNA-seq data; and quantification of tumor immunogenicity from expression data. Initiated with somatic mutations obtained from WES or WGS data, TIminer predicts mutated proteins arising from nonsynonymous variants, leveraging the Ensemble Variant Effect Predictor (VEP) [23]. Determination of Class-I HLA alleles is achieved through the utilization of Optitype applied to RNA-seq data [24]. Subsequently, the binding affinities of mutated peptides, derived from predicted proteins, to the reconstructed HLA alleles are forecasted using NetMHCpan [25]. Those peptides demonstrating high binding affinity, referred to as strong binders, are subsequently singled out as candidate neoantigens. StackTTCA introduces a novel stacking ensemble learning framework tailored for the identification of tumor T cell antigens [26]. The framework integrates 12 distinct feature-encoding schemes and 13 widely used machine learning methods. Initially, the baseline models are trained using these diverse approaches, generating a broad spectrum of predictive outputs. These outputs

are then aggregated to form a new probabilistic feature vector. Through an optimization process that incorporates feature selection strategies, this vector serves as the foundation for constructing the stacked model, which aims to leverage the collective strengths of the individual models for improved performance in identifying tumor T-cell antigens.

Table 1. List of existing computational methods for tumor T-cell antigen prediction.

Method	Year	Method [a]	Availability Online (as of 1 March 2024)
TTAgP [6]	2019	RF	https://github.com/bio-coding/TTAgP
iTTCA-Hybrid [7]	2020	RF, SVM	http://camt.pythonanywhere.com/iTTCA-Hybrid
iTTCA-RF [8]	2021	RF	http://112.124.26.17:7002/
TAP [11]	2021	ML	No
PSRTTCA [12]	2023	RF	http://pmlabstack.pythonanywhere.com/PSRTTCA
VaxiJen v2.0 [13]	2007	PSL-DA	http://www.ddg-pharmfac.net/vaxijen/VaxiJen/VaxiJen.html
LENS [18]	2023	Over two dozen separate tools to generate tumor antigen predictions	https://gitlab.com/landscape-of-effective-neoantigens-software
OpenVax [19]	2020	Bioinformatics pipeline	https://github.com/openvax
pVACtools [20]	2020	Various MHC-I prediction algorithms	https://github.com/griffithlab/pVACtools
nextNEOpi [21]	2022	WES/WGS/RNA-Seq pipeline	https://github.com/icbi-lab/nextNEOpi
TIminer [22]	2017	NGS pipeline	https://bio.tools/timiner
StackTTCA [26]	2023	Stacking ensemble-learning algorithm	No

[a] WES—whole exome sequencing, WGS—whole genome sequencing, RNA-seq—RNA sequencing, NGS—next generation sequencing, RF—random forest, SVM—support vector machine, ML—machine learning, PSL-DA—partial least squares—discriminant analysis.

In the years following the inception of VaxiJen, an abundance of new data concerning immunogenic proteins of tumor origin have accumulated in the scientific literature, necessitating an update to its dataset. Novel data expand the repertoire of immunogenic proteins, thereby facilitating the derivation of new models with heightened predictive power. In this study, a diverse array of machine learning (ML) methods, including *k*-nearest neighbor (*k*NN), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), random forest (RF), support vector machine (SVM), and extreme gradient boosting (XGBoost), were applied to an updated set of known tumor antigens. The resulting ML models underwent rigorous validation through receiver operating characteristic (ROC) statistics on an external test set. The three top-performing models are to be subsequently integrated into the latest version of the VaxiJen server.

2. Materials and Methods

2.1. Datasets

A comprehensive search was undertaken within the U.S. National Library of Medicine's PubMed database, with a specific focus on the retrieval of studies discovering immunogenic peptides. In addition, the built-in "Similar articles" tool within PubMed was harnessed to extend the exploration of pertinent literature. It is crucial to note that only human studies were considered throughout the selection process. The structures of peptides were carefully extracted from the respective source articles that reported them. Furthermore, the Immune Epitope Database (IEDB) [27] served as a pivotal reference for established peptide epitopes. This rigorous process resulted in the identification of 212 immunogenic peptides tested in humans, constituting the positive dataset. Additionally, a mirror dataset comprising 212 non-immunogenic peptides was systematically compiled from the literature to constitute the negative set. The criteria for determining immunogenicity included evidence from positive Major Histocompatibility Complex (MHC) binding assays and affirmative T-cell assays conducted in vivo within human subjects. Having accomplished these two conditions, a peptide antigen was considered immunogenic. Conversely, peptides categorized as non-immunogenic lacked evidence of T-cell response despite their bind-

ing to MHC molecules. Therefore, the negative set comprised non-immunogenic human peptide tumor antigens, while the positive set included human tumor peptide antigens that demonstrated immunogenic properties. This stringent categorization facilitated the precise classification of peptides based on their immunogenicity profiles. Data underwent cross-verification with the relevant peptide records accessible in the IEDB. Subsequently, all collected peptides were randomized, and 20% were randomly selected to form the test set which was later used for model evaluation. Consequently, the training set on which the models were derived comprised 170 immunogenic and 170 non-immunogenic peptides, while the test set comprised 42 immunogenic and 42 non-immunogenic peptides.

2.2. Descriptors

In the present study, *E*-descriptors were employed to provide a quantitative characterization of peptide sequences. These descriptors were introduced by Venkatarajan and Braun, who derived five numerical values for each of the 20 naturally occurring amino acids through multidimensional scaling based on 237 physicochemical properties [28]. The primary component, *E*1, exhibits a pronounced correlation with the hydrophobicity of amino acids, while the second component, *E*2, conveys information about the molecular size and steric properties. Components *E*3 and *E*5 delineate amino acid propensities for occurrence in α-helices and β-strands, respectively. The *E*4 component incorporates considerations of partial specific volume, the number of codons, and the relative frequency of amino acids in proteins. Each peptide in our datasets was represented as a string of $5n$ elements, where n denotes the length of the peptide. As these strings varied in length, they were homogenized into uniform vectors through auto- and cross-covariance (*ACC*) transformation.

2.3. Auto-Cross Covariance (ACC) Transformation

The auto- and cross-covariance (*ACC*) transformation of protein sequences was initially introduced in 1993 by Wold et al. [14]. This method serves as an alignment-independent preprocessing technique designed to convert polypeptide chains of varying lengths into uniform, equal-length vectors. Notably, the *ACC* transformation incorporates considerations for neighbor effects. The calculation of auto- and cross-covariance is executed according to the following formulas:

$$ACC_{j,j}(L) = \sum_{i}^{n-L} \frac{E_{j,i} \times E_{j,i+L}}{n-L} \quad (1)$$

$$ACC_{j,k}(L) = \sum_{i}^{n-L} \frac{E_{j,i} \times E_{k,i+L}}{n-L} \quad (2)$$

where:

E—the *E*-descriptor value
j, k ($j \neq k$)—the number of the *E*-descriptor (j, k = 1–5)
i—the position of amino acid in the peptide chain (i = 1, 2, 3. . .n)
n—the number of the amino acids in the protein
L—lag-value; the length of the frame of the contiguous amino acids

2.4. Machine Learning Methods

Multiple machine learning (ML) methods were employed in this study, as enumerated below. The scikit-learn library [29], implemented in the Python 3.7 programming language, was utilized for model construction. In order to achieve optimal results, every model underwent hyperparameter tuning using the Grid search technique [30], which is also implemented in the scikit-learn library. It is a process that searches exhaustively through a manually specified subset of the hyperparameter space of the targeted algorithm. The hyperparameters found to be optimal for every model are specified in Supplementary File S1. Each one of the models was designed to process a series of auto- and cross-covariance (*ACC*)-transformed amino acid sequences as input. The objective of each model was to predict the antigenic or non-antigenic classification of each peptide. Accordingly, the

output for peptides predicted as antigens was designated as 1, while for those predicted as non-antigens, it was designated as 0.

2.4.1. *k*-Nearest Neighbor (*k*NN)

The *k*-nearest neighbors (*k*NN) algorithm operates on the principle that akin data points tend to possess analogous labels or values [31]. In the training phase, the algorithm assimilates the complete training dataset as a reference. Subsequently, when generating predictions, it computes the distance between the input data point and all instances within the training dataset, employing a selected distance metric, such as the Euclidean distance.

2.4.2. Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) serves as a linear model employed for both classification and dimensionality reduction [32]. In the context of classification, LDA facilitates the projection of data from a D-dimensional feature space to a D′ (D > D′) dimensional space, with the objective of maximizing variability between classes while concurrently minimizing variability within classes.

2.4.3. Quadratic Discriminant Analysis (QDA)

Quadratic discriminant analysis (QDA) closely resembles linear discriminant analysis (LDA), with the notable distinction of utilizing a quadratic decision surface to segregate measurements among two or more classes of objects or events [32]. QDA operates under the assumption that each class adheres to a Gaussian distribution. The class-specific prior is determined by the proportion of data points belonging to the class, while the class-specific mean vector represents the average of the input variables associated with the class.

2.4.4. Support Vector Machine (SVM)

The Support Vector Machine (SVM) represents a supervised machine learning paradigm that endeavors to identify a hyperplane effectively demarcating two classes [33]. Given that there exists an infinite number of hyperplanes capable of precisely segregating the two classes, SVM aims to select the optimal hyperplane by identifying the one with the maximum margin. The maximum margin signifies the greatest separation between hyperplanes, thereby maximizing the distance between the two classes.

2.4.5. Random Forest (RF)

Random Forest (RF) amalgamates the outcomes of multiple decision trees to produce a consolidated result, thereby mitigating overfitting and rendering the model less susceptible to noise and outliers within the data [34]. In the context of classification tasks, the output of the random forest corresponds to the class most frequently selected across the constituent trees. Alternatively, for regression tasks, the returned prediction is typically the mean or average prediction derived from the individual trees.

2.4.6. Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) represents an optimized distributed gradient boosting library meticulously crafted for the efficient and scalable training of machine learning models [35]. Functioning as an ensemble learning method, XGBoost amalgamates predictions from multiple weak decision trees, culminating in a robust prediction. The algorithm generates a series of weak decision trees, each incrementally grown on the weighted data derived from the preceding tree. The assignment of weights is achieved through the optimization of a function that gauges the alignment of model coefficients with the underlying data, commonly referred to as the loss function. The prediction yielded by the final ensemble model is the weighted sum of predictions derived from the preceding tree models.

2.5. Machine Learning Models Validation

The machine learning (ML) models developed in the present study underwent validation through a rigorous process, encompassing 10-fold cross-validation and assessment against an independent test set. Cross-validation is a technique for estimating the performance of a predictive model [36]. In this approach, the training set is split into k smaller sets (in our case $k = 10$). For each fold, a model is trained using $k - 1$ folds as training data and the resulting model is validated on the remaining part of the data (i.e., it is used as a test set). The performance measure reported by k-fold cross-validation is then the average of the values computed in the loop. This approach can be computationally expensive but does not waste too much data (as is the case when fixing an arbitrary validation set), which is a major advantage in problems such as an inverse inference where the number of samples is very small. The predictive performance of each model was evaluated through Receiver Operating Characteristic (ROC) statistics, which categorize outcomes into four possibilities as follows: true positives (TP, accurately predicting an immunogen as an immunogen), true negatives (TN, accurately predicting a non-immunogen as a non-immunogen), false positives (FP, inaccurately predicting a non-immunogen as an immunogen), and false negatives (FN, inaccurately predicting an immunogen as a non-immunogen). Based on these outcomes, key parameters were computed [37]. *Sensitivity*, (3), measures the proportion of correctly identified positive samples out of the total positive samples, offering insight into a model's ability to detect immunogenic peptides. Conversely, *specificity* (4) gauges the accuracy of identifying negative samples out of the total negative samples, indicating the model's proficiency in recognizing non-immunogenic peptides. *Accuracy* (5), a widely utilized metric, quantifies the ratio of correctly classified samples to the total sample size, providing an overall assessment of classification performance. The *Area Under the ROC Curve (AROC)* assesses predictive efficacy, with values ranging from 0.5 for random prediction to 1.0 for perfect prediction. *Matthew's Correlation Coefficient (MCC)* (6) evaluates the correlation between the observed and predicted classifications, accounting for imbalanced data. An MCC of +1 denotes a perfect prediction, −1 signifies complete disagreement between predictions and true values, while zero indicates performance no better than random guessing.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{6}$$

To assess the robustness of the model predictions and ascertain that they are not mere chance occurrences, a Y-scrambling method was employed [38]. This method involves the following steps: Initially, the model is trained using the original feature–target pairs, and its performance is noted. Subsequently, for a specified number of iterations, the target column is systematically shuffled, and the model is retrained on the newly formed feature–target pairs. The expectation is that the model exhibits proficient performance on the original data while manifesting diminished efficacy on the shuffled data. If this anticipated behavior is not observed, and the metric displays minimal variance, it suggests that the predictions lack robustness and may be attributed to chance.

We evaluated the attribute performance for each of the top three performing models, namely SVM, RF, and XGBoost. This assessment utilized two model inspection techniques to gauge the contribution of individual features to the statistical performance of a fitted model on a given dataset. Firstly, permutation feature importance, implemented using the scikit-learn library, involved randomly shuffling the values of a single feature and observing the resultant degradation of the model's score. Secondly, drop-column feature

importance resulted in the removal of an entire feature column from the dataset, followed by training a new model and assessing its performance. Both methods operate on the premise that the removal of an unimportant feature from the dataset should not result in a decrease in the model's performance.

3. Results and Discussion

A dataset comprising 212 antigenic and immunogenic, and 212 antigenic but non-immunogenic, peptides was assembled following the methodology delineated previously. This dataset was partitioned into a training set consisting of 170 immunogenic and 170 non-immunogenic peptides, and a test set comprising 42 immunogenic and 42 non-immunogenic peptides. Each peptide was numerically represented as strings of 5*n* *E*-descriptors, where *n* denotes the number of amino acid residues. To standardize the representation, the strings of varying lengths were subjected to *ACC*-transformation with a $L = 7$. The *lag* value of 7 was chosen as the length of the shortest peptide in the dataset. Consequently, the training set was transformed into a 340×175 (7×5^2) matrix, while the test set was converted into an 84×175 matrix. The flowchart of data preprocessing is presented in Figure 1.

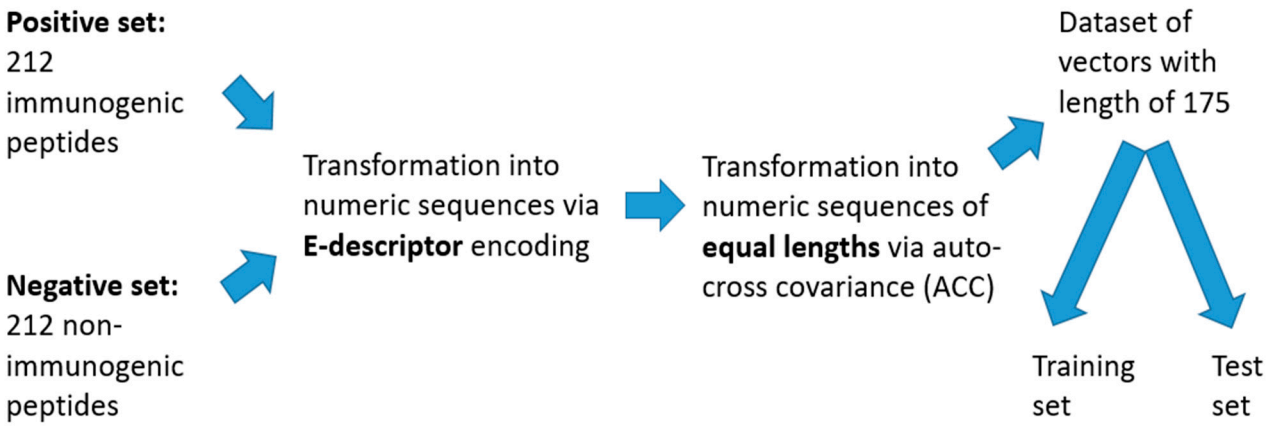

Figure 1. Flowchart of data preprocessing.

Six supervised machine learning algorithms were employed to construct classification models for predicting immunogenicity, utilizing the training set. To enhance the precision of the models, their hyperparameters underwent tuning via the grid search algorithm in combination with 10-fold cross-validation on the training set. This method entails predefined dictionaries encompassing hyperparameters and their respective value ranges, with the model evaluated across all feasible combinations employing cross-validation. Optimal hyperparameters were determined for each model as follows: the *k*NN model exhibited its peak predictive performance at $k = 2$, while the LDA method demonstrated superior predictions with *solver* = *svd*. Similarly, the QDA model achieved its highest performance with *reg_param* = *0.0*. The SVM method reached its apex predictive capabilities with parameters *C* = *2*, *gamma* = *10*, and *kernel* = *rbf*. Furthermore, the RF model displayed optimal predictions with *max_depth* = *80*, *max_features* = *2*, and *n_estimators* = *300*. Lastly, the XGBoost method showcased its highest performance with *learning_rate* = *0.3*, *max_depth* = *3*, and *n_estimators* = *100*.

The performance of the models on the training set is presented on Table 2. The *k*NN model demonstrated the best recognition capabilities for classifying the immunogenic peptides (*sensitivity* = 0.95), but a very poor performance on the non-immunogenic peptides (*specificity* = 0.29). Both LDA and QDA models demonstrated poor overall performance regarding both classes. In contrast, SVM and XGBoost models showcased balanced performances, demonstrating robust predictive capabilities for both immunogenic and non-immunogenic peptides. The RF model exhibited the highest accuracy, primarily at-

tributed to its superior predictive ability for non-immunogens (*specificity* 0.9). While the RF model's capability to predict immunogens was relatively strong (*sensitivity* 0.72), it was lower compared to its performance in predicting non-immunogens.

Table 2. Summary of the performance of the machine learning (ML) models on the training set (10-fold cross-validation). *TN*—true negatives; *FN*—false negatives; *TP*—true positives; *FP*—false positives; *AROC*—area under the ROC curve (sensitivity vs. 1-specificity); *MCC*—Matthew's correlation coefficient.

Model	*TN*	*FN*	*TP*	*FP*	Sensitivity	Specificity	Accuracy	*AROC*	*MCC*
*k*NN	49	9	161	121	0.95	0.29	0.62	0.62	0.31
LDA	92	58	112	78	0.66	0.54	0.60	0.61	0.20
QDA	115	105	65	55	0.38	0.68	0.53	0.53	0.06
SVM	138	38	132	32	0.78	0.81	0.80	0.88	0.59
RF	153	48	122	17	0.72	0.90	0.81	0.87	0.63
XGBoost	129	35	135	41	0.79	0.76	0.78	0.86	0.55

Once the models were trained, they underwent subsequent evaluation on the hold-out test set to give an objective conclusion about their generalization capabilities (Table 3). The observed results align well with the ones seen on the training set, indicating lack of overfitting. Although exceling in the recognition of immunogens, the *k*NN model exhibited a notably poor performance when classifying the non-immunogenic peptides, rendering it unsuitable for effective classification. Conversely, both the LDA and QDA models demonstrated subpar overall performance across various metrics. The SVM, RF, and XGBoost models emerged as the most promising predictive models, displaying balanced and robust performances across all evaluated metrics. Consequently, these models were selected for further validation.

Table 3. Summary of the performance of the machine learning (ML) models on the test set. *TN*—true negatives; *FN*—false negatives; *TP*—true positives; *FP*—false positives; *AROC*—area under the ROC curve (sensitivity vs. 1-specificity); *MCC*—Matthew's correlation coefficient.

Model	*TN*	*FN*	*TP*	*FP*	Sensitivity	Specificity	Accuracy	*AROC*	*MCC*
*k*NN	9	4	38	33	0.90	0.21	0.56	0.56	0.16
LDA	22	16	26	20	0.62	0.52	0.57	0.57	0.14
QDA	26	26	16	16	0.38	0.62	0.50	0.50	0.00
SVM	36	9	33	6	0.79	0.86	0.82	0.83	0.64
RF	39	10	32	3	0.76	0.93	0.85	0.80	0.70
XGBoost	32	9	33	10	0.79	0.76	0.77	0.83	0.55

Initially, Y-scrambling was conducted for each of the SVM, RF, and XGBoost models. This process involved randomly shuffling the target pairs in the training data, constructing new models, and subsequently evaluating them on the test data. This iterative procedure was repeated 100 times, and the average accuracy value was computed. The accuracies for each model hovered around 0.5 (0.5075 for the SVM model, 0.5182 for the RF, and 0.4987 for the XGBoost), which is a noticeable deterioration in performance close to that of a random classifier (which has an accuracy score of 0.5 for binary class classification). The resulting poor performance of the model with the shuffled data compared to the good performance of the original data demonstrates the robustness of the models.

Next, we undertook an analysis to determine the significance of each feature in influencing the performance of the chosen models. Employing two feature importance techniques on the test set—permutation feature importance and drop-column feature importance—we evaluated the impact of each feature on the accuracy score of the three models (Supplementary File S1). Each feature was assessed by comparing the baseline accuracy of the model with the accuracy when the feature was altered. Positive values

indicated that altering (permuting or dropping) the feature decreased the model's performance, suggesting the feature's importance. Conversely, zero or negative values suggested that altering the feature either had no effect or even improved the model's performance, indicating a negative correlation with model performance. However, the observed values were generally minimal, precluding definitive conclusions regarding feature importance. After that we observed the common features between the different models. Table 4 delineates the top 10 most important features for each model. The *ACC* features are represented as follows: the first numerical index represents the *E*-descriptor of the first amino acid, the second index stands for the *E*-descriptor of the second amino acid, and the third one, for the *lag*-value.

Table 4. Top 10 attributes ranked according to their importance for SVM, RF, and XGBoost models. The common features between the two feature importance techniques for each of the three models are given in bold.

Model	Top 10 Features
SVM	
Permutation feature importance	***ACC*145**, *ACC*137, ***ACC*313**, *ACC*114, *ACC*117, *ACC*217, *ACC*116, ***ACC*147**, *ACC*111, ***ACC*234**
Drop-column feature importance	***ACC*313**, ***ACC*145**, *ACC*552, *ACC*446, *ACC*416, *ACC*324, *ACC*247, ***ACC*234**, ***ACC*147**, *ACC*141
RF	
Permutaion feature importance	*ACC*511, *ACC*417, *ACC*234, *ACC*117, *ACC*433, *ACC*514, *ACC*247, *ACC*315, *ACC*341, *ACC*254
Drop-column feature importance	*ACC*547, *ACC*541, *ACC*537, *ACC*536, *ACC*535, *ACC*534, *ACC*533, *ACC*532, *ACC*531, *ACC*525
XGBoost	
Permutaion feature importance	*ACC*535, *ACC*137, *ACC*516, *ACC*145, *ACC*254, *ACC*151, *ACC*442, *ACC*132, *ACC*531, ***ACC*441**
Drop-column feature importance	*ACC*135, *ACC*511, *ACC*457, *ACC*237, *ACC*455, *ACC*453, ***ACC*441**, *ACC*432, *ACC*311, *ACC*233

Both permutation feature importance and drop-column feature importance methods identified four attributes as significant for the SVM model, namely *ACC145*, *ACC147*, *ACC313*, and *ACC234*. *ACC145* and *ACC147* denote the cross-covariance between *E1* and *E4* descriptors at $L = 5$ and $L = 7$, respectively, reflecting the relationship between hydrophobicity, partial specific volume, the number of codons, and the relative frequency of amino acids within defined intervals in protein sequences. *ACC313* measures the cross-covariance between *E3* and *E1* descriptors at $L = 3$, indicating the association between amino acid occurrence in α-helices and hydrophobicity at the specific interval. *ACC234* represents the cross-covariance between *E2* and *E3* descriptors at $L = 4$, highlighting the relationship between molecular size, steric properties, and amino acid occurrence in α-helices within the defined interval.

Although there were no shared attributes between the permutation feature importance and drop-column feature importance techniques for the RF method, the drop-column feature importance technique highlighted *ACC537*, *ACC536*, *ACC535*, *ACC534*, *ACC533*, *ACC532*, and *ACC531* as its most crucial features. This discovery suggests a noteworthy relationship between amino acid propensities for occurrence in α-helices (descriptor *E*3) and β-strands (descriptor *E*5) across all possible *lag*-values (1–7). This indicates that these amino acid properties may exert a significant influence on the immunogenicity of tumor peptides.

ACC441 is the sole attribute shared between the permutation feature importance and drop-column feature importance techniques for the XGBoost model. It quantifies the autocovariance of E4 descriptors of adjacent amino acids, revealing crucial associations among their partial specific volume, the number of codons, and the relative frequency of amino acids in the protein sequence.

No feature appeared in the top 10 for all the models simultaneously, indicating variability in feature importance across the models. For the drop-column feature importance technique, there were no common features among any of the models.

Figure 2 illustrates the common attributes identified through permutation feature importance technique for the three models. The SVM and RF models both consider *ACC117* and *ACC234* as the most significant attributes. *ACC117* signifies the auto-covariance of hydrophobicity (E1 descriptor) at $L = 7$, while *ACC234* measures the cross-covariance between molecular size and steric properties (E2 descriptor) and the propensity of amino acids for occurrence in α-helices (E3 descriptor) at $L = 4$. Additionally, the SVM and XGBoost models share two important attributes, *ACC137* and *ACC145*. *ACC137* represents the cross-covariance between hydrophobicity (E1 descriptor) and the propensity of amino acids for occurrence in α-helices (E3 descriptor) at $L = 7$, whereas *ACC145* denotes the cross-covariance between E1 and E4 descriptors at $L = 5$. Finally, the RF and XGBoost models share *ACC254* as a significant attribute for both. *ACC254* quantifies the cross-covariance between molecular size and steric properties, along with the propensity of amino acids for occurrence in β-strands (E5 descriptor).

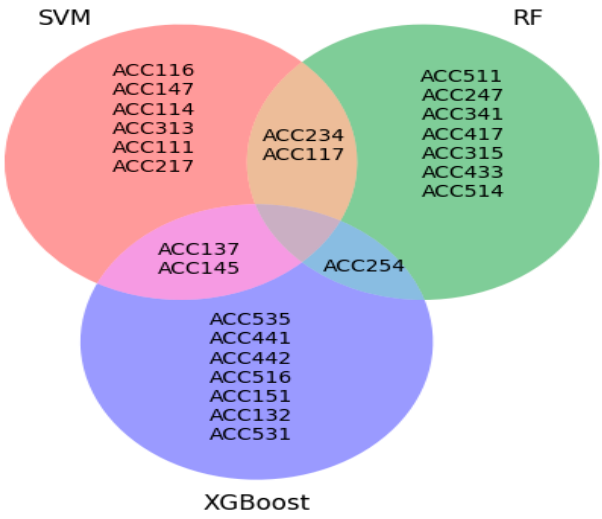

Figure 2. Key attributes shared among the three best-performing models: top 10 attributes for each model identified using permutation feature importance technique. The most significant attributes for SVM, RF, and XGBoost algorithms are highlighted in red, green, and blue circles, respectively. The overlapping areas represent joint significant attributes between the algorithms; pink for SVM and XGBoost, grey for RF and XGBoost, and brown for SVM and RF.

The two feature importance techniques utilized aim to identify the most significant features, yet using auto- and cross-covariance feature encoding alone cannot definitively explain why a particular feature holds importance. While it can provide insights into the correlation between specific biological properties encoded with corresponding E-descriptors, it falls short of elucidating causation. Moreover, each identified important feature represents a correlation between different biological properties, lacking consensus among them. Coupled with the generally minimal values regarding feature importance, this limitation prevents us from making conclusive suggestions or drawing conclusions about why one feature might outweigh another in importance.

We conducted a comparative analysis of the performance metrics of our three selected models with those of other in silico methods for predicting human tumor antigens. These methods were available online and could be applied without requiring specific programming knowledge. To facilitate this comparison, we evaluated their performance on the current test set and contrasted it with the consensus classification obtained from the majority voting of our three models. Specifically, if two or more models classified a peptide

as immunogenic, it received a consensus classification as an immunogen. The results demonstrated that our three models exhibited superior performance across all assessed statistical measures (Table 5).

Table 5. Comparison among the three selected models and other software for human tumor antigen predictions on the test set.

Algorithm	Sensitivity	Specificity	Accuracy	MCC
SVM + RF + XGBoost	0.76	0.91	0.83	0.67
TTAgP 1.0	0.60	0.00	0.30	−0.50
iTTCA-Hybrid	0.24	0.17	0.20	−0.60
iTTCA-RF	0.24	0.24	0.24	−0.52
PSRTTCA	0.76	0.26	0.51	0.03

All in silico tools and methods for predicting human tumor antigens exhibited very poor performance on the external test set utilized in our study. The Matthew's correlation coefficient (MCC) values for all tested models indicated complete disagreement between prediction and observation for TTAgP 1.0, iTTCA-Hybrid, and iTTCA-RF, with predictions no better than random for PSRTTCA. A possible explanation for these dismal results across all assessment metrics is a significant disparity between the datasets used to train these models and the training set employed for our models.

Given their remarkable performance, the three selected models have emerged as prime candidates for incorporation into the forthcoming third version of the web-based immunogenicity prediction server, VaxiJen. The newly gathered data spanning 15 years include insights into previously unknown immunogens, which contributed to the updated dataset and models. Consequently, the new dataset along with SVM, RF, and XGBoost models have been selected for integration into the third iteration of the VaxiJen web server (VaxiJen v3.0).

4. Conclusions

In this study, we employed six supervised machine learning (ML) methods on a dataset comprising 212 human immunogenic peptides validated in vivo, alongside a corresponding dataset of non-immunogenic human peptides, to develop models for immunogenicity prediction. Following parameter optimization, these models underwent validation through internal cross-validation and testing on a holdout dataset. The top-performing models—SVM, RF, and XGBoost—were further evaluated using Y-scrambling. These robust models are slated for integration into the upcoming iteration of the VaxiJen web server (VaxiJen v3.0).

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app14104034/s1: Supplementary File S1, Table S1: Feature importance for SVM model, Table S2: Feature importance for RF model, Table S3: Feature importance for XGBoost model; Supplementary File S2: Training set; Supplementary File S3: Test set.

Author Contributions: Conceptualization, S.S. and I.D.; methodology, I.D.; software, S.S.; validation, S.S.; formal analysis, S.S.; investigation, S.S.; resources, S.S.; data curation, S.S.; writing—original draft preparation, S.S.; writing—review and editing, I.D.; visualization, S.S.; supervision, I.D.; project administration, I.D.; funding acquisition, I.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Bulgarian National Plan for Recovery and Resilience through the Bulgarian National Science Fund, grant number BG-RRP-2.004-0004-C01, and by the Science and Education for Smart Growth Operational Program, as well as co-financed by the European Union through the European Structural and Investment funds (Grant No. BG05M2OP001-1.001-0003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The models developed in the present study will be implemented in the updated version of VaxiJen 3.0.

Acknowledgments: During the preparation of this work, the authors used ChatGPT for English editing. After using this tool, the manuscript underwent a comprehensive editing service. The authors take full responsibility for the content of the publication.

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

Abbreviations

ACC	auto- and cross-covariance
AROC	area under the ROC curve
CAR	chimeric antigen receptor
FN	false negatives
FP	false positives
GA	genetic algorithm
IEDB	The Immune Epitope Database
*k*NN	*k*-nearest neighbor
L	lag-value
LDA	linear discriminant analysis
LENS	Landscape of Effective Neoantigens Software
MCC	Matthew's correlation coefficient
MHC	Major Histocompatibility Complex
ML	machine learning
nextNEOpi	Nextflow NEOantigen prediction pipeline
NGS	next generation sequencing
PLS-DA	partial least squares—discriminant analysis
QDA	quadratic discriminant analysis
RF	random forest
RNA-seq	RNA sequencing
ROC	receiver operating characteristic
SCM	scoring card method
SMOTE	Synthetic Minority Over-sampling Technique
SVM	support vector machine
TIminer	Tumor Immunology Miner
TN	true negatives
TP	true positives
VEP	Variant Effect Predictor
WES	whole exome sequencing
WGS	whole genome sequencing
XGBoost	extreme gradient boosting

References

1. Singh, T.; Bhattacharya, M.; Mavi, A.K.; Gulati, A.; Rakesh, N.K.S.; Gaur, S.; Kumar, U. Immunogenicity of cancer cells: An overview. *Cell Signal.* **2024**, *113*, 110952. [CrossRef]
2. Woo, S.R.; Corrales, L.; Gajewski, T.F. Innate immune recognition of cancer. *Annu. Rev. Immunol.* **2015**, *33*, 445–474. [CrossRef] [PubMed]
3. Tsung, K.; Norton, J.A. In situ vaccine, immunological memory and cancer cure. *Hum. Vaccines Immunotherap.* **2016**, *12*, 117–119. [CrossRef] [PubMed]
4. Okada, M.; Shimizu, K.; Fujii, S.I. Identification of Neoantigens in Cancer Cells as Targets for Immunotherapy. *Int. J. Mol. Sci.* **2022**, *23*, 2594. [CrossRef] [PubMed]
5. Soria-Guerra, R.E.; Nieto-Gomez, R.; Govea-Alonso, D.O.; Rosales-Mendoza, S. An overview of bioinformatics tools for epitope prediction: Implications on vaccine development. *J. Biomed. Inform.* **2015**, *53*, 405–414. [CrossRef]

6. Beltrán, J.F.L.; Herrera, L.B.; Farias, J.G. TTAgP 1.0: A computational tool for the specific prediction of tumor T cell antigens. *Comp. Biol. Chem.* **2019**, *83*, 107103. [CrossRef] [PubMed]
7. Charoenkwan, P.; Nantasenamat, C.; Hasan, M.M.; Shoombuatong, W. iTTCA-Hybrid: Improved and robust identification of tumor T cell antigens by utilizing hybrid feature representation. *Anal. Biochem.* **2020**, *599*, 113747. [CrossRef] [PubMed]
8. Jiao, S.; Zou, Q.; Guo, H.; Shi, L. iTTCA-RF: A random forest predictor for tumor T cell antigens. *J. Transl. Med.* **2021**, *19*, 449. [CrossRef]
9. Kawashima, S.; Ogata, H.; Kanehisa, M. AAindex: Amino Acid Index Database. *Nucleic Acids Res.* **1999**, *27*, 368–369. [CrossRef]
10. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, P.W. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
11. Herrera-Bravo, J.; Herrera, L.B.; Farias, J.G.; Beltrán, J.F. TAP 1.0: A robust immunoinformatic tool for the prediction of tumor T-cell antigens based on AAindex properties. *Comput. Biol. Chem.* **2021**, *91*, 107452. [CrossRef] [PubMed]
12. Charoenkwan, P.; Pipattanaboon, C.; Nantasenamat, C.; Hasan, M.M.; Moni, M.A.; Lio, P.; Shoombuatong, W. PSRTTCA: A new approach for improving the prediction and characterization of tumor T cell antigens using propensity score representation learning. *Comput. Biol. Med.* **2023**, *152*, 106368. [CrossRef] [PubMed]
13. Doytchinova, I.A.; Flower, D.R. VaxiJen: A server for prediction of protective antigens, tumour antigens and subunit vaccines. *BMC Bioinform.* **2007**, *8*, 4. [CrossRef] [PubMed]
14. Hellberg, S.; Sjöström, M.; Skagerberg, B.; Wold, S. Peptide quantitative structure-activity relationships, a multivariate approach. *J. Med. Chem.* **1987**, *30*, 1126–1135. [CrossRef] [PubMed]
15. Wold, S.; Jonsson, J.; Sjöström, M.; Sandberg, M.; Rännar, S. DNA and peptide sequences and chemical processes multivariately modelled by principal component analysis and partial least squares projections to latent structures. *Anal. Chim. Acta* **1993**, *277*, 239–253. [CrossRef]
16. Leardi, R.; Boggia, R.; Terrile, M. Genetic algorithms as a strategy for feature selection. *J. Chemom.* **1992**, *6*, 267–281. [CrossRef]
17. Ståhle, L.; Wold, S. Partial least squares analysis with cross-validation for the two-class problem: A monte carlo study. *J. Chemom.* **1987**, *1*, 185–196. [CrossRef]
18. Vensko, S.P.; Olsen, K.; Bortone, D.; Smith, C.C.; Chai, S.; Beckabir, B.; Fini, M.; Jadi, O.; Rubinsteyn, A.; Vincent, B.G. LENS: Landscape of Effective Neoantigens Software. *Bioinformatics* **2023**, *39*, 6. [CrossRef] [PubMed]
19. Kodysh, J.; Rubinsteyn, A. OpenVax: An open-source computational pipeline for cancer neoantigen prediction. In *Bioinformatics for Cancer Immunotherapy*; Boegel, S., Ed.; Methods in Molecular Biology; Humana: New York, NY, USA, 2020; Volume 2120, pp. 147–160. [CrossRef]
20. Hundal, J.; Kiwala, S.; McMichael, J.; Miller, C.A.; Xia, H.; Wollam, A.T.; Liu, C.J.; Zhao, S.; Feng, Y.Y.; Graubert, A.P.; et al. pVACtools: A Computational Toolkit to Identify and Visualize Cancer Neoantigens. *Cancer Immunol. Res.* **2020**, *8*, 409–420. [CrossRef]
21. Rieder, D.; Fotakis, G.; Ausserhofer, M.; René, G.; Paster, W.; Trajanoski, Z.; Finotello, F. nextNEOpi: A comprehensive pipeline for computational neoantigen prediction. *Bioinformatics* **2022**, *38*, 1131–1132. [CrossRef]
22. Tappeiner, E.; Finotello, F.; Charoentong, P.; Mayer, C.; Rieder, D.; Trajanoski, Z. TIminer: NGS data mining pipeline for cancer immunology and immunotherapy. *Bioinformatics* **2017**, *33*, 3140–3141. [CrossRef] [PubMed]
23. McLaren, W.; Gil, L.; Hunt, S.E.; Riat, H.S.; Ritchie, G.R.S.; Thormann, A.; Flicek, P.; Cunningham, F. The Ensembl Variant Effect Predictor. *Genome Biol.* **2016**, *17*, 122. [CrossRef]
24. Szolek, A.; Schubert, B.; Mohr, C.; Sturm, M.; Feldhahn, M.; Kohlbacher, O. OptiType: Precision HLA typing from next-generation sequencing data. *Bioinformatics* **2014**, *30*, 3310–3316. [CrossRef] [PubMed]
25. Jurtz, V.; Paul, S.; Andreatta, M.; Marcatili, P.; Peters, B.; Nielsen, M. NetMHCpan-4.0: Improved Peptide-MHC Class I Interaction Predictions Integrating Eluted Ligand and Peptide Binding Affinity Data. *J. Immunol.* **2017**, *199*, 3360–3368. [CrossRef]
26. Charoenkwan, P.; Schaduangrat, N.; Shoombuatong, W. StackTTCA: A stacking ensemble learning-based framework for accurate and high-throughput identification of tumor T cell antigens. *BMC Bioinform.* **2023**, *24*, 301. [CrossRef]
27. Vita, R.; Overton, J.A.; Greenbaum, J.A.; Ponomarenko, J.; Clark, J.D.; Cantrell, J.R.; Wheeler, D.K.; Gabbard, J.L.; Hix, D.; Sette, A.; et al. The immune epitope database (IEDB) 3.0. *Nucleic Acids Res.* **2015**, *43*, D405–D412. [CrossRef]
28. Venkatarajan, M.S.; Braun, W. New quantitative descriptors of amino acids based on multidimensional scaling of a large number of physical-chemical properties. *J. Mol. Model.* **2001**, *7*, 445–453.
29. Scikit-Learn Machine Learning in Python. Available online: https://scikit-learn.org (accessed on 5 May 2024).
30. Sklearn.Model_Selection.GridSearchCV. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (accessed on 5 May 2024).
31. Goldberger, J.; Hinton, G.E.; Roweis, S.T.; Salakhutdinov, R.R. Neighbourhood components analysis. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 5–8 December 2005; pp. 513–520.
32. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer: New York, NY, USA, 2008; Section 4.3; pp. 106–119.
33. Bhavsar, H.P.; Panchal, M. A Review on Support Vector Machine for Data Classification. *IJARCET* **2012**, *1*, 185–189.
34. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
35. Chen, T.Q.; Guestrin, C. Xgboost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]

36. Ojala, M.; Garriga, G.C. Permutation tests for studying classifier performance. *J. Mach. Learn. Res.* **2010**, *11*, 1833–1863.
37. Tharwat, A. Classification assessment methods. *New Engl. J. Entrepr.* **2020**, *17*, 168–192. [CrossRef]
38. Wold, S.; Eriksson, L. Statistical Validation of QSAR Results. In *Chemometric Methods in Molecular Design*; Weinheim van de Waterbeemd, H., Ed.; Wiley: Hoboken, NJ, USA, 1995; pp. 309–318.