The Empirical Comparison of Machine Learning Algorithm for the Class Imbalanced Problem in Conformational Epitope Prediction

Binti Solihah[1,2], Azhari Azhari[3], Aina Musdholifah[4]

[1,3,4]*Department Computer Science and Electronics, Universitas Gadjah Mada, Indonesia*

[2]*Informatics Dept, FTI, Universitas Trisakti, Indonesia*

[1]binti@trisakti.ac.id, [3]arisn@ugm.ac.id, [4]aina_m@ugm.ac.id

Abstract - A conformational epitope is a part of a protein-based vaccine. It is challenging to identify using an experiment. A computational model is developed to support identification. However, the imbalance class is one of the constraints to achieving optimal performance on the conformational epitope B cell prediction. In this paper, we compare several conformational epitope B cell prediction models from non-ensemble and ensemble approaches. A sampling method from Random undersampling, SMOTE, and cluster-based undersampling is combined with a decision tree or SVM to build a non-ensemble model. A random forest model and several variants of the bagging method is used to construct the ensemble model. A 10-fold cross-validation method is used to validate the model. The experiment results show that the combination of the cluster-based under-sampling and decision tree outperformed the other sampling method when combined with the non-ensemble and the ensemble method. This study provides a baseline to improve existing models for dealing with the class imbalance in the conformational epitope prediction.

Keywords: sampling-based method, class imbalance, conformational epitope, B-cell, machine learning-based

I. INTRODUCTION

The development of computational methods for epitope prediction is an active research area for more than 30 years. Although more than 90 percent of B cell epitopes are conformational epitopes, a linear epitope prediction model was developed first. The conformational epitope's prediction model was started by CEP, which utilizes solvent accessibility properties [1]. Various methods have utilized the physicochemical properties of amino acids (Amino Acid Index, B factor), structure (ASA, RSA, Protrusion Index, CN, HSE), and statistics (log odd ratio), which have been implemented to improve the performance of the model. Several machine learning models have been created [2-6], but the resulting models' performance is still not satified.

Among the existing methods to handle the class imbalance, sampling is the simple approach and independent to the classifier. The undersampling method is superior to oversampling [7]. The combined method of undersampling and oversampling is superior to the undersampling method. Still, according to [7], in the ensemble approach, the Bagging Method is superior to other methods such as Boosting and cost-sensitive. Some sampling approaches have been implemented in the conformational epitope's predictive models [2-3], [5]. The other approach is cost-sensitive method [6]. The cost sensitive is superior compared to several ensemble methods, both boosting and hybrid between boosting and bagging (Easy ensemble and Balance Cascade [8]). However, performance of modified bagging model in conformational epitope prediction is unknown. Study of [9] show that bagging extention based can improve the model's performance in class imbalace problems.

The handling of class imbalance is still active research. Many methods have been applied to handle class imbalance, namely the data level approach, the algorithm level, the cost-sensitive approach that can work at both levels, and the use of an ensemble [7]. A simple approach that is easy to implement and independent of the classification method used is sampling-based. The most simple sampling is random oversampling and random undersampling [10-11]. Several sampling approaches that consider sample conditions can improve model performance [11-12]. The ensemble approach includes bagging and boosting. The boosting approach selects the sample and gives the sample weight based on misclassification costs. The bagging approach uses a simple approach by forming a dataset at the bootstrapping stage. The sample with replacement mechanism in bagging still produces resampling results with a distribution still imbalance. Some sampling methods, such as random oversampling, SMOTE, simple undersampling, and cluster-based sampling, is used to change the bootstrapping [9].

The remainder of this paper is organized as follows. Section II briefly explains the methods for handling class imbalance. Sections III discuss the experiment result and their significance. The last section provides the conclusion of the study.

II. METHOD

This section describes preparing a dataset for model building and methods for handling machine learning-based class returns.

A. Data Preparation

Preparation of the dataset, as depicted in Fig. 1, consists of four steps: (1) Data collection of the 3D, Ag-Ab complex structure and separation of the antigen chains, (2) Identification of the residue exposed to the specified antigen, (3) Labelling the exposed residue as epitope or non-epitope, (4) Formation of feature vectors. The final goal of this stage is the formation of a conformational epitope dataset with the arff format. Each row in the dataset represents a characteristic vector of a residue that is part of the residue exposed to a particular antigen.

The process of collecting the 3D structure of Ag-Ab complex data and separating certain antigen chains is described in the diagram in Fig. 2. The 3D structure of the Ag-Ab complex is collected by downloading it from the PDB database by the PDBID. The list of PDBID and its antigen chains used in this study refers to [13]. TreeD (3D) antigen structure data obtained by separating the antigen chain from the Ag-Ab complex based on the information in the .pdb file metadata on the keyword CMPND. There are 78 antigen chains derived from the 60 Ag-Ab complexes. In Fig. 3, the separation of the C antigen chain from the 1A2Y complex structure is shown.

Identification of exposed residues was carried out based on the RSA threshold value. In detail, the steps for identifying the exposed residue are described in the diagram in Fig. 4. RSA is defined as the ASA value divided by Maximum ASA. The investigators used different RSA limits in defining the residual exposed. In this study, the RSA limits were 0.01.

The next step is labeling the exposed residue as epitope residue or non-epitope residue. Epitope residues are antigen residues that interact with antibody residues. Interactions between residues were identified using the Euclid distance function with the PSAIA [14]. Two residues are said to interact if their distance is less than 4 Angstroms. The distance between residue d (a, b) is defined as the minimum Euclidean distance between Cα residue a and C α residue b. In detail, the residue labeling process is explained by the diagram in Fig. 5.

The last stage of preparing the dataset is the extraction of residual features. Each residue is characterized as ASA, RSA, CN, HSE 2, QSE 8, FSE 4, EFSE 8, SFSE 8, b factor, b factor ca, lo, PSAIA 23, AAI 544 [4]. ASA, RSA, CN, HSE, QSE 8, FSE 4, EFSE 8, SFSE 8, and PSAIA values represent their geometric structures' residues. The value of B factor, log odds ratio, and AAindex represent atomic or residue flexibility, propensity score, and amino acid residues' physicochemical properties, respectively [5], [15-16]. The dataset is presented in arff format with 602 feature vectors. The small RSA threshold accommodates more epitope data taken from the complex while increasing the negative class taken. The ratio of positive to negative classes at the RSA threshold is 0.07: 0.93.

B. Conformational Epitope Prediction Model

In this study, some models that combined the SVM or decision tree and sampling method are developed to predict conformational epitope. As shown in Table I, the sampling methods are SMOTE, borderline SMOTE, random undersampling, random oversampling, and cluster-based undersampling. The sampling method is also implemented in the Bagging method variations. The Model comparisons were carried out on the performance parameters AUC, Gmean, Adjusted G mean, and F score.

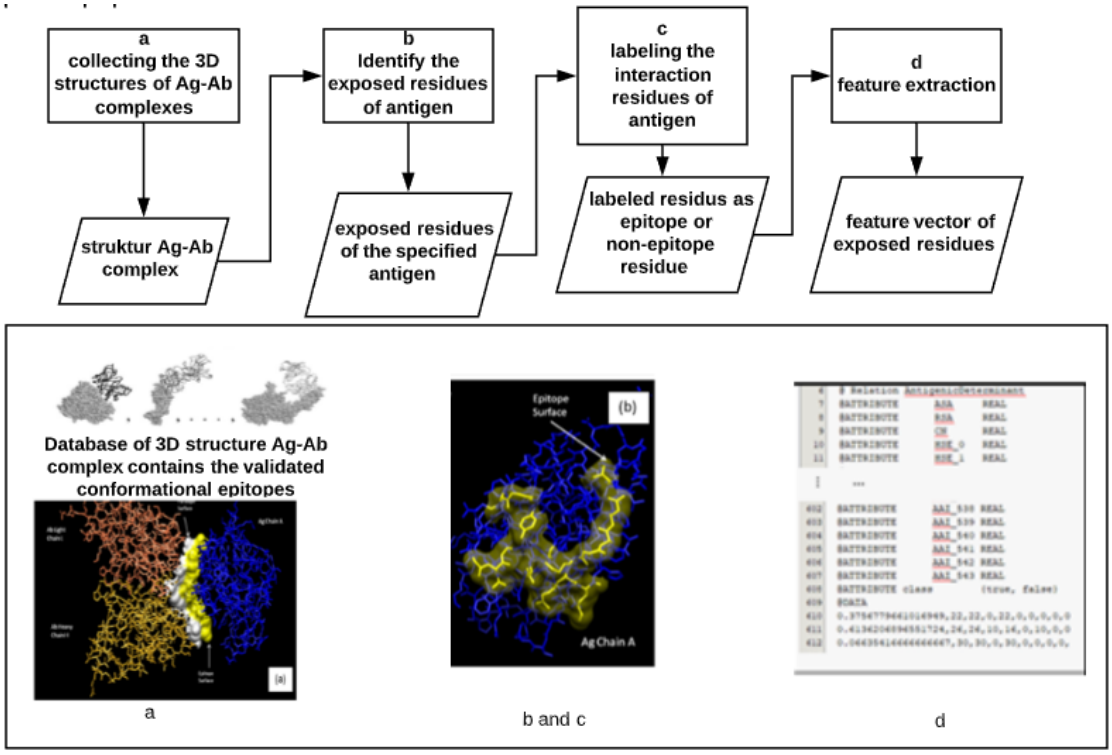

Fig. 1 Data preprocessing steps

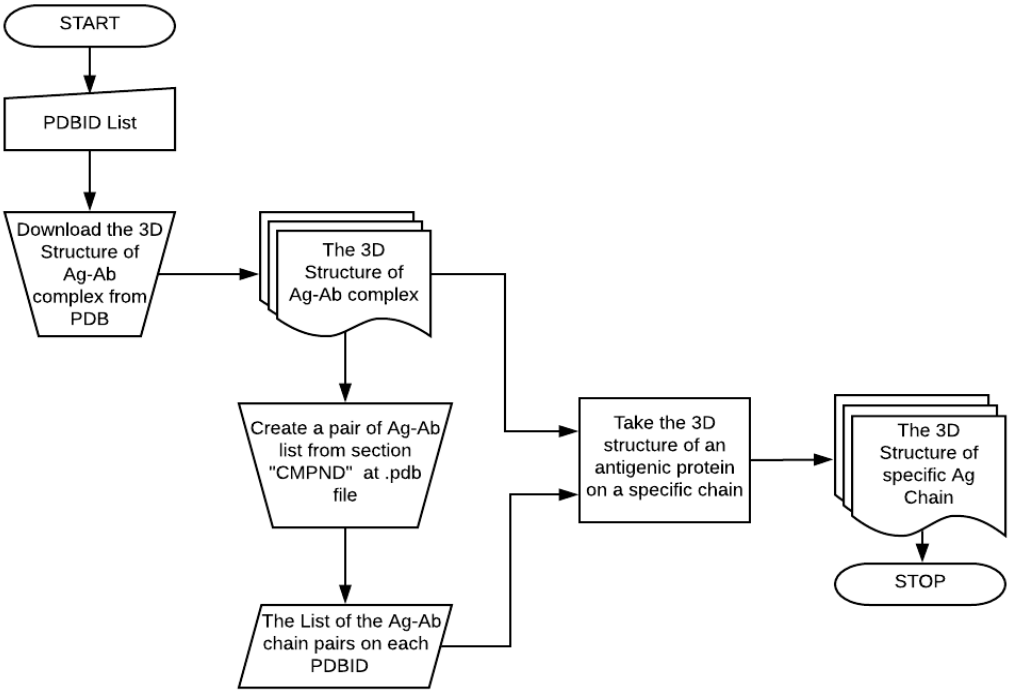

Fig. 2 Flow of the Ag-Ab complex 3D structure data collection

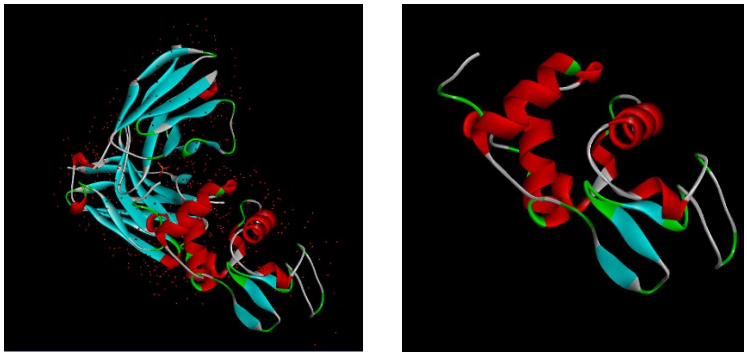

Fig. 3 (a) 3D structure of 1A2Y; (b) 3D structure of chain C (part of 1A2Y) (visualized by Biovia Software)

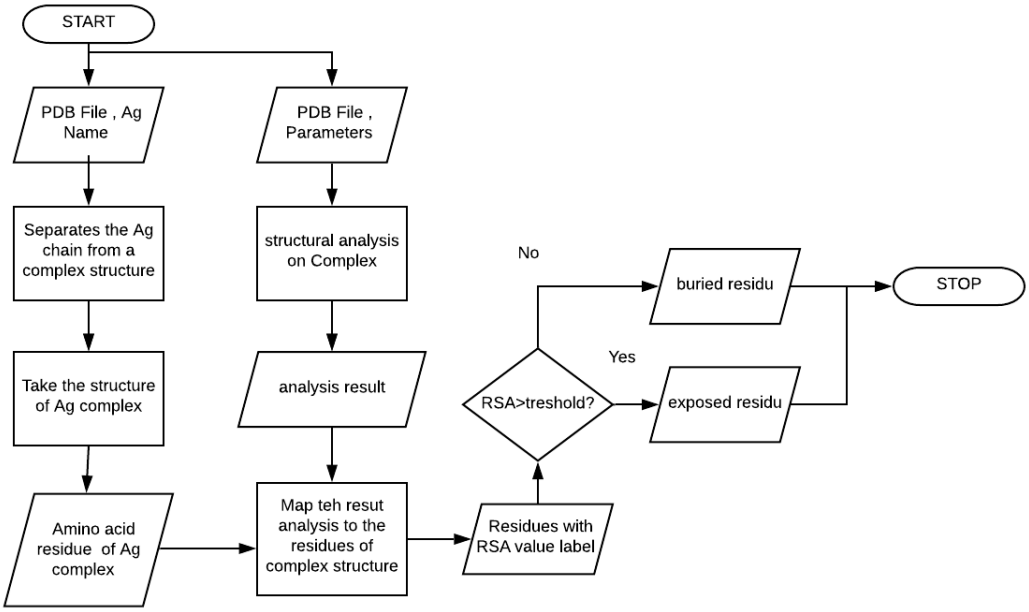

Fig. 4 Identification of exposed amino acid residues

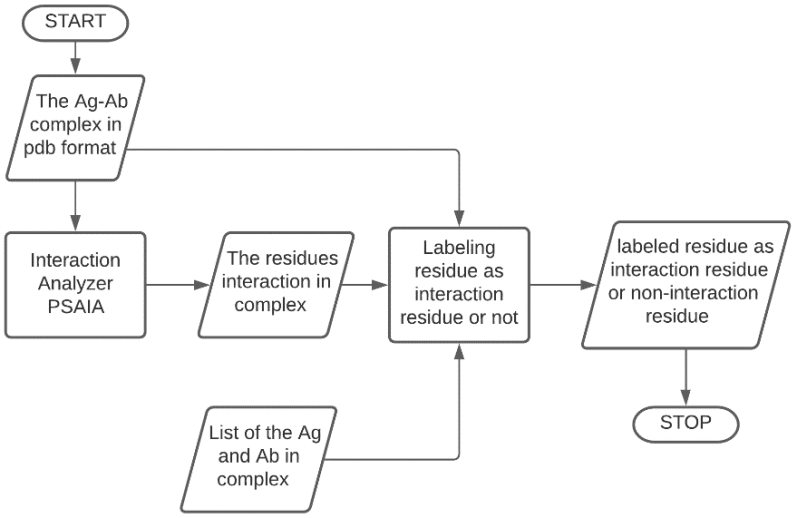

Fig. 5 Residue labeling process

TABEL I
DEVELOPED CONFORMATIONAL EPITOPE PREDICTION MODELS

Category	Conformational Epitope Prediction Method	Abbreviation
F1: Non-ensemble with balancing	SMOTE + SVM/DT	SMOTESVM, SMOTEDT
	Borderline SMOTE + SVM/DT	BorderSMOTESVM, BorderSMOTEDT
	Random Under sampling + SVM/DT	RusSVM, RusDT
	Cluster-based Under sampling + SVM/DT	CusSVM, CusDT
	RandomForest	RF
F2: Ensemble	Bagging with sampling modification + DT	OvBag DT, EBBag DT, SMOTEBag DT, CusBag DT

Several methods proposed to handle imbalance in the conformational epitope prediction shown in Table I is briefly explained in this section.

1) Sampling Methods: Random oversampling and Random undersampling are the simplest method to rabalance data. Over and under sampling are the simplest method to rebalance data. In oversampling, data from minority class is resampled with replacement until it number balance to data from majority class. Otherwise, in Random undersampling, majority class is sampled with replacement randomly as much as minority class number so that a balance data achieved. The main issue in Random oversampling and Random undersampling are overfitting and the loss of main features, respectively. Under-sampling result in better classifier than oversampling and the combination of both is better than oversampling or undersampling [11]. Oversampling is selected when minority data avaiable in dozens and undersampling is choosen when data available at hundreds. When the data size is large, combination of under and oversampling is appropriate.

The other approach is SMOTE [11], where the oversampling is done not by replacement, but is done by create synthetic data which is consider the neighborhood sample data around the sample point of min class. New data points are generated using (1).

$$X_{new} = X_i + \left(\hat{X}_i - X_i\right) \times \delta \quad (1)$$

where X_i is the random sample from the k neighbor and δ is a random number in the interval [0,1]. The combination of SMOTE and undersampling give the best performance [11]. Considering the influence of data points in the borderline area to the classifier performance, ref. [12] use Borderline-SMOTE in which the synthetic data is generated from the minority sample in the border area. Han uses the same technique with Chawla et al. [11] to generate the synthetic data point. If k is sum of the nearest neighbor from p_i sample, then p_i will be categorized as the border sample if the sum of neighbour from majority class is larger than from minority class $\left(\frac{k}{2} \leq k' \leq k\right)$.

If the oversampling method generates new data points based on existing sample points, the undersampling method uses the information on the sample to select the sample. Among the methods used to select samples is clustering. Researchers use a different approach in determining the dataset group used in model building. The first approach is to include the entire dataset without paying attention to the class label [17]. The second approach is only to use the majority class to form the dataset [19-20].

2) Bagging: Bagging (Bootstrap AGGregatING) categorized as a parallel ensemble method (Fig. 6). Creating the subsets of the data sample in the bootstrapping mechanism is conducted by sampling with replacement. Bagging should use unstable base-learner such as decision tree. Among the extension of bagging algorithm to handle imbalance class are Exactly Balance Bagging (EBBag), OverBagging (OvBag), and SMOTEBagging (SMOTEBag) [21]. EBBag is the derivative of Bagging with Random under sampling mechanism. In the EBBag the number of samples is fixed. OvBag and SMOTEBag are the variant of random oversampling. In OverBagging, The bootstrap sample is created by Random oversampling with replacement in the minority data. In SMOTEBag, the minority data is oversampled by SMOTE. The resampling rate of SMOTE is changed in each bootstrap to increase deversity of each sample. Another method which is combine undersampling with SMOTE is Resampling Ensemble Algorithm (REA) [22].

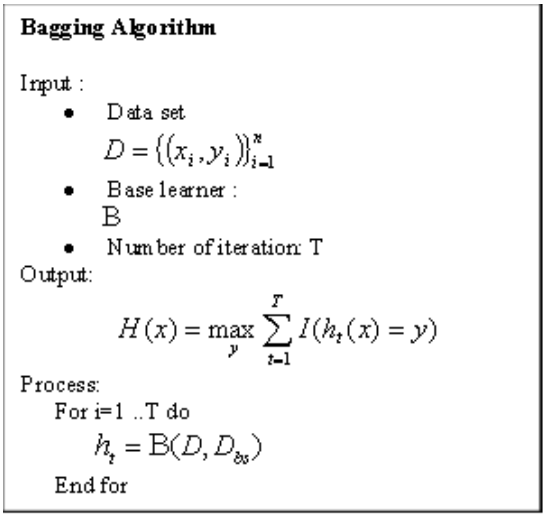

Fig. 6 Bagging method

3) Random Forest: Random Forest is the Bagging extension with trees as the base-learner. Random Forest differ from Bagging at how to use features to build the tree. Each tree is generated with features randomly selected from the available features.

III. RESULTS AND DISCUSSION

Experiments were carried out on a dataset that had been built using the steps described in Section III. The experimental scenario is described in the validation and performance section.

A. Validation and Performance Measurement

Internal model validation was done by using 10-fold cross-validation. The dataset is divided into ten parts, 9 of which are used as training data, and one is used as testing data. The prediction model of the conformational epitope is a prediction model developed with binary classification. A positive class is a class for epitope residues, and a negative class is a class for non-epitope residues. The test results of the model are presented in a confusion matrix. In the confusion matrix, there are four categories of prediction results, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Each class's performance is stated in the True Positive Rate (TPR) and True Negative Rate (TNR). The overall model performance is expressed by Area Under the Curve (AUC), Geometric mean (Gmean), Adjusted Graph, and F-score. Accuracy is not used as a performance measure due to bias in the class imbalance case.

B. Experiment Result

The model is implemented with the Netbean IDE in Java language and the JSAT statistics library [23]. The parameter settings are the same as in [4]. The non-ensemble model's performance on a dataset with a threshold of RSA $\geqslant$ 0.01 is shown in Table 2. Each cell in the table describes the average performance of the model. The highest performance SVM model performance is achieved in the BorderSMOTE SVM model. The results of statistical tests with the Friedman test and post hock analysis with alpha 0.05 at the AUC value resulted in a p-value of 1.29422E-11. There are significant differences between the various variants of the models developed with the SVM. Post hock test with nemenyi shows that there is no significant difference between the SVM model and other model variants except for Cus SVM, which is stated by a p-value of 0.002.

In the decision tree-based model, the highest model performance on the three measurement methods AUC, Gmean, and Adjusted Gmean, is achieved in the CusDT model. The best performance of the model at the F-score was achieved in the BorderSMOTE DT model. The results of statistical tests with Friedman test and post hock with alpha 0.05 at AUC resulted in a p-value of 4.79415E-07. It shows significant differences between the various variants of the model developed with the decision tree. The CusDT model is significantly different from the DT, SMOTE DT, BorderSMOTE DT, and RusDT based on the nemenyi post hock analysis. The post hock analysis between the CusDT and DT, SMOTE DT, BorderSMOTE DT, and RusDT models resulted in p-values of 0.002, 0.03, 0.01, and 0.02, respectively.

In general, the model performance, as stated by the AUC, Gmean, Adjusted Gmean, and F-score of the decision tree-based model, is superior to that of the SVM model (Table II). The combination of SMOTE and BorderSMOTE gives better model performance based on AUC and Gmean parameters. The combination of the sampling method with DT is better than combining the sampling method with SVM in AGM, except for BorderSMOTE. The p-value of the Wilconox test results on the AUC between decision tree vs. SVM and BorderSmote SVM vs. BorderSmote DT respectively 0.01 and 0.001. The p-value between RUS DT vs. RUS SVMand SMOTE SVM vs. SMOTE DT respectively 0.001 and 0.001. The p-value between CUS DT vs. CUS SVM is 0.03. The model developed with Smote SVM, and BorderSMOTE SVM is better than Smote DT and BorderSMOTE DT. The models developed with DT and RUS DT are better than the SVM and RUS SVM models. The use of CUS on DT and SVM did not provide a significant difference in AUC.

TABEL II
THE PERFORMANCE OF NON-ENSEMBLE MODELS

Family	Model	TPR	TNR	Precision	AUC	G Mean	Adjusted G Mean	F Measure
SVM	SVM	0.28	0.95	0.33	0.61	0.50	0.72	0.28
	SMOTESVM	0,82	0,68	0,19	0,75	0,74	0,68	0,30
	BorderSMOTESVM	0,76	0,81	0,27	**0,78**	**0,78**	**0,79**	**0,39**
	RusSVM	0,25	0,95	0,27	0,60	0,48	0,70	0,25
	CusSVM	0,72	0,73	0,18	0,73	0,72	0,73	0,29
DT	DT	0.29	0.97	0.43	0.63	0.53	0.74	0.35
	SMOTEDT	0,34	0,96	0,43	0,65	0,57	0,76	0,38
	BorderSMOTEDT	0,35	0,96	0,44	0,65	0,58	0,76	**0,39**
	RusDT	0,32	0,96	0,41	0,64	0,56	0,75	0,36
	CusDT	0,84	0,78	0,23	**0,81**	**0,81**	**0,78**	0,36

TABEL III
THE PERFORMANCE OF ENSEMBLE MODELS

Model	TPR	TNR	Precision	AUC	G Mean	Adjusted G Mean	F Measure
BagDT	0,24	0,99	0,56	0,61	0,49	0,73	0,34
OverBagDT	0,25	0,99	0,57	0,62	0,50	0,73	0,35
CusBagDT	0,90	0,81	0,26	**0,85**	**0,85**	**0,83**	**0,41**
CusRF	0,75	0,83	0,26	0,79	0,79	0,81	0,39
EBBag	0.25	0.99	0.57	0.62	0.50	0.73	0.35
RF	0.06	1.00	0.62	0.53	0.25	0.61	0.11

As presented in Table III in the ensemble model, the CusBag DT model's performance is the best in the four performance parameters. The Friedman test, the AUC value on the five models, resulted in a p-value of 5.61214E-11. The nemenyi analysis results resulted in a p-value of 0.02 for the DT Bag and CusBag DT. The post-hock analysis between the DT Bag model and the other models did not show any significant differences.

IV. CONCLUSION

The dataset for developing conformational epitope prediction is imbalanced. The machine learning-based method is sensitive to class imbalance. Applying cluster-based undersampling is quite effective than applying other sampling methods. Modified Bagging produces better performance than the Random Forest. The decision tree in this prediction model produces better performance than SVM with linear kernels. The performance of the conformational epitope prediction model still needs to be improved so that it can be used as a tool in vaccine development using the rational design method. Apart from handling the imbalance class, the use of other more representative features can be tried. Another thing that is also important to do is the prediction of the antigen with multiple epitopes.

ACKNOWLEDGMENT

This research was funded by Trisakti University in the form of doctoral scholarships with contract number 0341/USAKTI/SKR/BSDM/DT/IV/2016.

REFERENCES

[1] U. Kulkarni-kale, S. Bhosle, and A. S. Kolaskar, "CEP : a conformational epitope prediction server," Nucleic Acids Res., vol. 33, no. Web Server issue, pp. 168–171, 2005.

[2] G. A. Dalkas and M. Rooman, "SEPIa , a knowledge-driven algorithm for predicting conformational B-cell epitopes from the amino acid sequence," BMC Bioinformatics, vol. 18, no. 95, pp. 1–12, 2017.

[3] J. Ren, Q. Liu, J. Ellis, and J. Li, "Tertiary structure-based prediction of conformational B-cell epitopes through B factors," Bioinformatics, vol. 30, pp. 264–273, 2014.

[4] B. Solihah, A. Azhari, and A. Musdholifah, "Enhancement of conformational B-cell epitope prediction using CluSMOTE," PeerJ Comput. Sci., vol. 6, 2020.

[5] J. Ren, Q. Liu, J. Ellis, and J. Li, "Positive-unlabeled learning for the prediction of conformational B-cell epitopes," BMC Bioinformatics, vol. 16, no. Suppl 18,

pp. 1–15, 2015.

[6] J. Zhang, X. Zhao, P. Sun, B. Gao, and Z. Ma, "Conformational B-Cell Epitopes Prediction from Sequences Using Cost-Sensitive Ensemble Classifiers and Spatial Clustering," Biomed Res. Int., vol. 2014, pp. 1–12, 2014.

[7] M. Galar, A. Fern, E. Barrenechea, and H. Bustince, "Hybrid-Based Approaches," IEEE Trans. Syst. Cybern. -PART C Appl. Rev., vol. 42, no. 4, pp. 463–484, 2012.

[8] X. Liu, J. Wu, and Z. Zhou, "Exploratory Undersampling for," IEEE Trans. Syst. Cybern. -PART BCYBERNETICS, vol. 39, no. 2, pp. 539–550, 2009.

[9] J. Blaszczynski and J. Stefanowski, "Neighbourhood sampling in bagging for imbalanced data," Neurocomputing, vol. 2, no. 5–6, 2014.

[10] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," Comput. Intell., vol. 20, no. 1, pp. 18–36, 2004.

[11] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE : Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[12] H. Han, W. Wang, and B. Mao, "Borderline-SMOTE : A New Over-Sampling Method in," in ICIC LNCS, 2005, pp. 878–887.

[13] N. D. Rubinstein, I. Mayrose, D. Halperin, D. Yekutieli, J. M. Gershoni, and T. Pupko, "Computational characterization of B-cell epitopes," Mol. Immunol., vol. 45, pp. 3477–3489, 2008.

[14] J. Mihel, M. Šiki, S. Tomi, B. Jeren, and K. Vlahovi, "PSAIA – Protein Structure and Interaction Analyzer," BMC Struct. Biol., vol. 11, pp. 1–11, 2008.

[15] P. H. Andersen, M. Nielsen, and O. L. E. Lund, "Prediction of residues in discontinuous B-cell epitopes using protein 3D structures," Protein Sci., vol. 15, pp. 2558–2567, 2006.

[16] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa, "AAindex : amino acid index database , progress report 2008," Nucleid Acids Res., vol. 36, no. November 2007, pp. 202–205, 2008.

[17] S. Yen and Y. Lee, "Expert Systems with Applications Cluster-based under-sampling approaches for imbalanced data distributions," Expert Syst. Appl., vol. 36, pp. 5718–5727, 2009.

[18] R. A. Sowah, M. A. Agebure, G. A. Mills, K. M. Koumadi, and S. Y. Fiawoo, "New Cluster Undersampling Technique for Class Imbalance Learning," Int. J. Mach. Learn. Comput., vol. 6, no. 3, pp. 205–214, 2016.

[19] W. Lin, C. Tsai, Y. Hu, and J. Jhang, "Clustering-based undersampling in class-imbalanced data," Inf. Sci. (Ny)., vol. 409–410, pp. 17–26, 2017.

[20] C. F. Tsai, W. C. Lin, Y. H. Hu, and G. T. Yao, "Under-sampling class imbalanced datasets by combining clustering analysis and instance selection," Inf. Sci. (Ny)., vol. 477, pp. 47–54, 2019.

[21] S. Wang and X. Yao, "Diversity analysis on imbalanced data sets by using ensemble models," 2009 IEEE Symp. Comput. Intell. Data Mining, CIDM 2009 - Proc., pp. 324–331, 2009.

[22] Y. Qian, Y. Liang, M. Li, G. Feng, and X. Shi, "A resampling ensemble algorithm for classification of imbalance problems," Neurocomputing, vol. 143, pp. 57–67, 2014.

[23] E. Raff, "JSAT : Java Statistical Analysis Tool , a Library for Machine Learning," J. Mach. Learn. Res., vol. 18, pp. 1–5, 2017.