DEEP AUTOENCODER LEARNING FEATURES FOR CLASSIFYING PEPTIDES DATA AS RECOMBINANT BACTERIAL MATERIAL USING PTVPSO-DEEP ELM ALGORITHM

I Cholissodin[1,2], N Suciati[1*], D Herumurti[1] and V Yurina[3]
[1]Informatics Engineering, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh November Surabaya
[2]Informatics Engineering, Faculty of Computer Science, Universitas Brawijaya Malang
[3]Department Of Pharmacy, Faculty of Medicine, Universitas Brawijaya Malang

*Corresponding author: nanik@if.its.ac.id

Abstract- Codon codes directly in the form of vector or 3D molecular to support feature learning often impede machine learning reading for the epitope classification process in extracting feature values that are more easily understood by computers. The compute time required for molecular 3D is also quite significant, which is directed more to a good visualization art approach only because the DNA visualization method for determining epitope and non-epitope classes is more suitable for observation by human experts, not computers. In addition, the length of each codon code in the peptides varies greatly. Therefore, computers for such dataset processing require standardized visualization transformation results before the classification process is carried out. In this research, a 2-stage approach was carried out. First, the deep autoencoder approach was used as a pre-process to transform the learning features from multi-dimensional variations to 2D or 3D which is more powerful than PCA and kernel trick. Then, the classification stage was carried out using the PTVPSO-Deep ELM algorithm. PSO is used to optimize the transformation and classification results. The test results showed that the proposed method is able to produce better and more stable metric values.

Keywords— autoencoder-ptvpso-deep-elm, classification-peptides, codon, epitope, recombinant-bacterial.

I. INTRODUCTION

Determining the correct class related to the epitope is a very crucial part of the peptide in the successful production of recombinant bacterial material in bio-molecular technology [1–6]. The toughest challenge in classifying peptides is determining the filter process for extracting the correct features to overcome the difficulty of reading and using it directly by only utilizing codon codes which have highly varied dimensional lengths that make it difficult to analyze the distribution of data between epitope and non- epitope classes [7]. When feature learning such as peptide_length, chou_fasman, etc. only rely on the results from tools from various bio-informatics software, the results are very different. The value of the information obtained can also be assessed subjectively for each tool with the risk of being less consistent, and certainly quite a lot of information being unreadable or lost [8,9].

There is quite a lot of research on Autoencoder deep learning algorithms the default of which is capable of being flexible [14–21], and can be freely set regarding how long the dimensions of the transformation composition result are, not towards reduction or weighting or some kind of pruning of the number of features, but more focused on summarizing the features so that it is more guaranteed that no information is lost in the data. Among these autoencoders, few have been developed for processing data that has a variety of data dimensions at the layer, and are dominated by deep learning algorithms the time complexity of which is quite large, so they need to be developed further. In this research, the development of the Deep Autoencoder algorithm was carried out by utilizing the speed of computing time from the ELM algorithm to become Deep ELM as a process for transforming peptide data into 2D or 3D dimensions the reasons of which are the processing speed and ease of visualization of more consistent results. Then in the final stage, the classification process was carried out using the Parallel Time-Variants Particle Swarm Optimization Deep Extreme Learning Machine (PTVPSO-Deep ELM) where the ELM also has a process for optimization in determining input weights and biases to overcome local optimality and underfit in the optimized method [22–25], both for the data transformation process to produce feature learning and also for the training and testing process when the classification was carried out. The PSO method was chosen with the consideration that it converges more quickly and there is a process in it that is able to balance the process of exploration and exploitation of candidate solutions in the form of decentralized particles.

II. METHODS

A. Peptides Data as Recombinant Bacterial Material

Peptide based on Fig. 1 is a collection of codon codes from the DNA part, which in this case is from the individual SARS-CoV virus, from the B-Cell and SARS-CoV parts [11]. In a bio-

molecular context, peptides can be classified into 2 classes, i.e., epitopes and non-epitopes. In the epitope class, a relationship can be drawn with the materials for making materials from recombinant bacteries, which can be in the form of pieces of codon code with dimensions that vary in length [26,27].

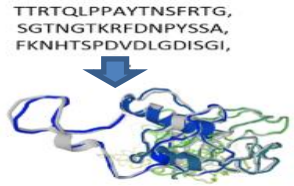

Fig. 1. Illustration of codon codes visualization in 3D Bio-Molecular form

B. Deep Autoencoder

Autoencoder is part of an artificial neural network algorithm that can use backpropagation or only feed forward techniques to obtain learning results that can be focused on feature extraction. There are two stages in it, i.e., encoding and decoding according Fig. 2(b). The encoding process can be used for data dimension reduction such as PCA [10], while the decoding process can be used to check the quality of the return to the original data at the input layer, for example for data compression [30,31]. In addition, Autoencoder is not limited to a few purposes as an independent method, but can be more widely combined with other techniques in various fields.

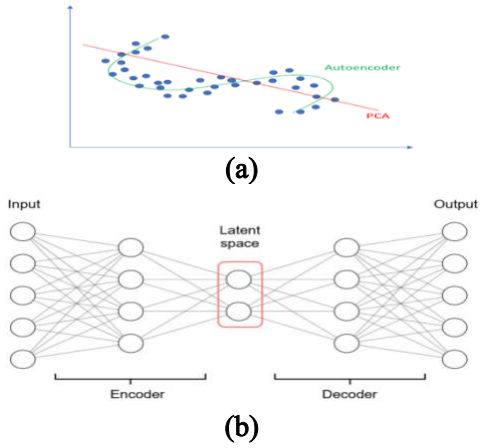

Fig. 2. (a) PCA vs ELM, (b) Deep Autoencoder illustration [32]

If there are many or more than one layer in the hidden layer, then it can be said to be a Deep Autoencoder, which can later be merged with a process for example for classification or according to the context needed in building an AI ecosystem. The main advantage of this method is the standardization of the process and ease of control for optimizing results which adopts the Artificial Neural Network (ANN) concept by utilizing the summary results of feature values in the hidden layer. The data values in the input layer are reflected in the target, which means that they are also used in the output layer [33].

C. Propose Machine learning of PTVPSO-Deep ELM Algorithm

The in-depth reasons for making this contribution are based on at least four things, i.e., ease of setup and standardization of feature transformations with varying dimensions from peptides for feature learning that are easier to interpret for various purposes, for example for dimension reduction, stretching feature values in the specific or flexible target layer, and visualization as well as for pre-processing before being entered into the machine learning algorithm. Then the speed of the transformation process is highly necessitated in managing input data with different dimensional lengths. Therefore, in the two cases as the basis for this initial process, a deep autoencoder algorithm based on the ELM algorithm is used. The third basis is the need for speed in the peptide classification process. In this process, Deep ELM is used. Because it connects or combines with the previous process, it is also called Dual ELM. The most crucial foundation is the need to optimize the results in the pre-preprocessing process or at least optimize the classification process. This stage uses the PTVPSO algorithm. This algorithm has been proven to be very fast in achieving global solution candidate expansion results with the addition of time-variant techniques to it. The gradual and detailed steps for the Autoencoder-PTVPSO Deep ELM algorithm can be presented in the following order based on Fig. 4.

Stages of Autoencoder PTVPSO-Deep ELM Algorithm.

1. For preprocessing to get feature learning utilize transformations based Deep Autoencoder algorithm
   a. If using or not using PTVPSO Ecosystem of the entire architecture, then the particle representation can be observed in Fig. 3.

Input Weight AE	Bias AE	Input Weight ELM	Bias ELM
..	..	..	..

Fig. 3. Illustration of the representation of the particle form PTVPSO

   b. The codon code data configuration, which initially had varying lengths, was made standard so that all code feature lengths were the same length, through the addition of adaptive column features to make relevant preparations according to the needs of the transformation process with the Deep Autoencoder.
   c. Making visualization results (a) in 2D/3D dimensions.
2. Training and testing using PTVPSO-Deep ELM algorithm
   a. The training process uses 80% of the datasets from B-Cell and SARS-CoV

$$\hat{Y} = \left(H = \frac{1}{\left(1+\exp\left(-\left(X_{train}.W^{T} + ones\left(N_{train},1\right)b\right)\right)\right)} \right) \left(\left(\left(H \right)^{T}.\left(H \right) \right)^{-1}.H^{T} \right) Y \quad (1)$$

   b. The testing process uses 20% of the dataset.

$$\hat{Y} = \left(\frac{1}{\left(1+\exp\left(-\left(X_{test}.W^{T} + ones\left(N_{test},1\right)b\right)\right)\right)} \right) \left(\left(\left(H \right)^{T}.\left(H \right) \right)^{-1}.H^{T} \right) Y \quad (2)$$

   c. Metric value evaluation.

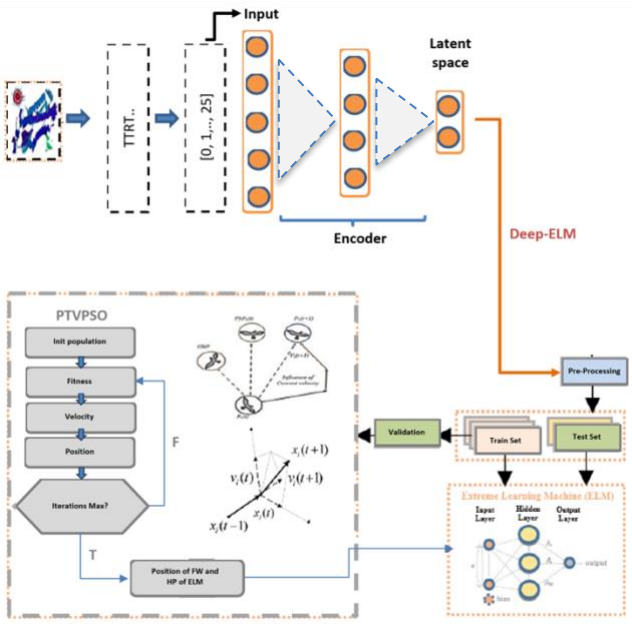

Fig 4. Scheme of Autoencoder PTVPSO-Deep ELM Algorithm

III. FINDINGS AND DISCUSSION

In the system test implementation scenario, 2 stages were carried out. The first stage was to get the results of the feature transformation into uniform 2D/3D dimensional lengths. Then a display of the results of the data visualization was created. In the second stage, the results of stage 1 were used for the training and testing process using the PTVPSO-Deep ELM algorithm. In the 1st stage of analysis, there was an additional comparison of the results of the PCA algorithm with the Deep Autoencoder algorithm to check which one is better. Test results based on Fig. 5 show that the Deep Autoencoder algorithm is much more representative to balance and centralized distribution of dataset in performing feature extraction compared to PCA. This can be observed in terms of visualization results of data distribution between classes which do not overlap each other and the classes are in regions that are further away from different classes, but closer to each other between the same classes and the patterns formed by each class are very clear.

Then the results of the 2nd stage metric values are shown in Table 1 which displays the advantages of the PTVPSO-Deep ELM algorithm with or without the Deep Autoencoder algorithm. An interesting thing to discuss can be seen from the architectural form of the entire algorithm ecosystem used. For example, in the first sub-ecosystem regarding the Deep Autoencoder algorithm, there is a way to determine the ideal input weights and biases for the amount of data that is balanced or unbalanced or for long and short intervals of codon codes so that they need to be given a filtering process, or make the treatment the same until the process to enter the machine learning. All of these things are given specific treatment or treated the same without optimization or all of them should be subjected to particle encoding using optimization techniques from the PTVPSO algorithm.

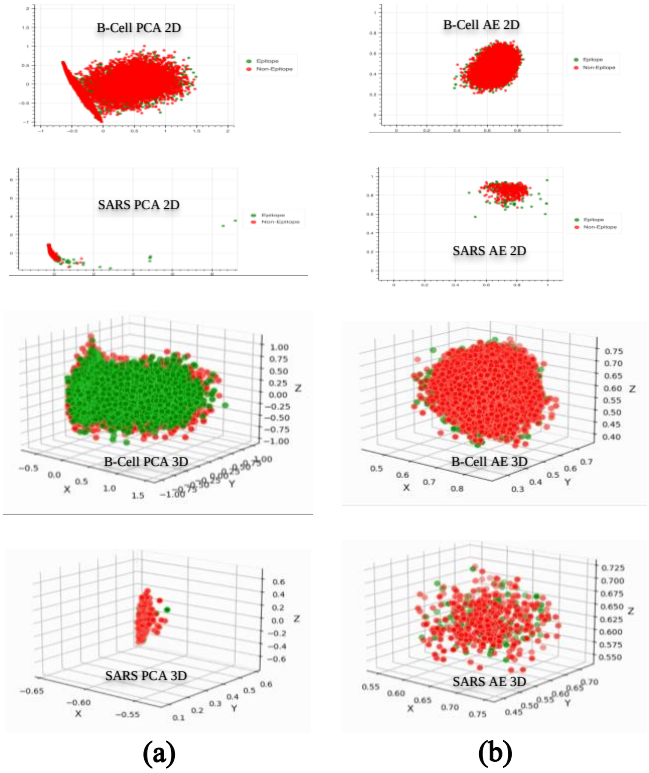

Fig. 5. Mapping Result of (a) PCA 2D and 3D, (b) AE 2D and 3D

Table 1. Result of data variation of test

Data Test	Accuracy with PCA utilize Scikit-Learn	Accuracy with Deep Autoencoder without Optimization
Trial 1st	0.817	0.856
Trial 2nd	0.885	0.875
Trial 3rd	0.817	0.856
Trial 4th	0.835	0.846
Trial 5th	0.835	0.856
Avg.	0.838	0.858

IV. CONCLUSION

This research has been able to make a new contribution by applying the Autoencoder PTVPSO-Deep ELM Algorithm, which is able to achieve more higher and stable based on average accuracy 85.78%. The implementation step begins with feature extraction using the ELM-based Deep Autoencoder algorithm, then continues with the training and testing process using the PTVPSO-Deep ELM algorithm. The future development plan is to add and compare the results of the 2D/3D Deep Autoencoder transformation to the 3D bio-molecular visual results for the reason of comparing the processing time and the metric value results. In the future, the features can be considered by combining features from codon codes with the results of several features from labs or bioinformatics tools such as peptide length, chou_fasman, etc. The epitope results can also be added to the intersection process with peptides from SARS-CoV2 for both the B-Cell and SARS-CoV datasets based on the best model obtained in each test scenario.

REFERENCES

[1] Huang, J.-H., Wen, M., Tang, L.-J., Xie, H.-L., Fu, L., Liang, Y.-Z., & Lu, H.-M. (2014). Using random forest to classify linear B-cell epitopes based on amino acid properties and molecular features. Biochimie, 103, 1–6. https://doi.org/10.1016/j.biochi.2014.03.016

[2] Rubinstein, N. D., Mayrose, I., & Pupko, T. (2009). A machine-learning approach for predicting B-cell epitopes. Molecular Immunology, 46(5), 840–847. https://doi.org/10.1016/j.molimm.2008.09.009

[3] Zheng, W., Zhang, C., Hanlon, M., Ruan, J., & Gao, J. (2014). An ensemble method for prediction of conformational B-cell epitopes from antigen sequences. Computational Biology and Chemistry, 49, 51–58. https://doi.org/10.1016/j.compbiolchem.2014.02.002

[4] Ozger, Z. B., & Cihan, P. (2022). A novel ensemble fuzzy classification model in SARS-CoV-2 B-cell epitope identification for development of protein-based vaccine. Applied Soft Computing, 116, 108280. https://doi.org/10.1016/j.asoc.2021.108280

[5] Soria-Guerra, R. E., Nieto-Gomez, R., Govea-Alonso, D. O., & Rosales-Mendoza, S. (2015). An overview of bioinformatics tools for epitope prediction: Implications on vaccine development. Journal of Biomedical Informatics, 53, 405–414. https://doi.org/10.1016/j.jbi.2014.11.003

[6] Nascimento, I. P., & Leite, L. C. C. (2012). Recombinant vaccines and the development of new vaccine strategies. Brazilian Journal of Medical and Biological Research, 45(12), 1102–1111. https://doi.org/10.1590/S0100-879X2012007500142

[7] Nanni, L., & Lumini, A. (2011). A new encoding technique for peptide classification. Expert Systems with Applications, 38(4), 3185–3191. https://doi.org/10.1016/j.eswa.2010.09.005

[8] Yurina, V., & Adianingsih, O. R. (2022). Predicting epitopes for vaccine development using bioinformatics tools. Therapeutic Advances in Vaccines and Immunotherapy, 10, 25151355221100218. https://doi.org/10.1177/25151355221100218

[9] Negahdaripour, M., Nezafat, N., Eslami, M., Ghoshoon, M. B., Shoolian, E., Najafipour, S., Morowvat, M. H., Dehshahri, A., Erfani, N., & Ghasemi, Y. (2018). Structural vaccinology considerations for in silico designing of a multi-epitope vaccine. Infection, Genetics and Evolution, 58, 96–109. https://doi.org/10.1016/j.meegid.2017.12.008

[10] Kabir, M. F., Chen, T., & Ludwig, S. A. (2023). A performance analysis of dimensionality reduction algorithms in machine learning models for cancer prediction. Healthcare Analytics, 3, 100125. https://doi.org/10.1016/j.health.2022.100125

[11] Cihan, P., & Ozger, Z. B. (2022). A new approach for determining SARS-CoV-2 epitopes using machine learning-based in silico methods. Computational Biology and Chemistry, 98, 107688. https://doi.org/10.1016/j.compbiolchem.2022.107688

[12] Nie, F., Zhu, W., & Li, X. (2020). Decision Tree SVM: An extension of linear SVM for non-linear classification. Neurocomputing, 401, 153–159. https://doi.org/10.1016/j.neucom.2019.10.051

[13] Zaki, A., Métwalli, A., Aly, M. H., & Badawi, W. K. (2022). Enhanced feature selection method based on regularization and kernel trick for 5G applications and beyond. Alexandria Engineering Journal, 61(12), 11589–11600. https://doi.org/10.1016/j.aej.2022.05.024

[14] Rashid, N., Hossain, M. A. F., Ali, M., Islam Sukanya, M., Mahmud, T., & Fattah, S. A. (2021). AutoCovNet: Unsupervised feature learning using autoencoder and feature merging for detection of COVID-19 from chest X-ray images. Biocybernetics and Biomedical Engineering, 41(4), 1685–1701. https://doi.org/10.1016/j.bbe.2021.09.004

[15] Hedayati, R., Khedmati, M., & Taghipour-Gorjikolaie, M. (2021). Deep feature extraction method based on ensemble of convolutional auto encoders: Application to Alzheimer's disease diagnosis. Biomedical Signal Processing and Control, 66, 102397. https://doi.org/10.1016/j.bspc.2020.102397

[16] Namazi, M., Karimi-Jafari, M. H., Qassemi, F., & Ghasemi, J. B. (2023). Chapter 6—Autoencoders in generative modeling, feature extraction, regression, and classification. In J. B. Ghasemi (Ed.), Machine Learning and Pattern Recognition Methods in Chemistry from Multivariate and Data Driven Modeling (pp. 119–136). Elsevier. https://doi.org/10.1016/B978-0-323-90408-7.00007-1

[17] Li, P., Pei, Y., & Li, J. (2023). A comprehensive survey on design and application of autoencoder in deep learning. Applied Soft Computing, 138, 110176. https://doi.org/10.1016/j.asoc.2023.110176

[18] Ryu, S., Jeon, B., Seo, H., Lee, M., Shin, J.-W., & Yu, Y. (2023). Development of deep autoencoder-based anomaly detection system for HANARO. Nuclear Engineering and Technology, 55(2), 475–483. https://doi.org/10.1016/j.net.2022.10.009

[19] Novoa-Paradela, D., Fontenla-Romero, O., & Guijarro-Berdiñas, B. (2023). Fast deep autoencoder for federated learning. Pattern Recognition, 143, 109805. https://doi.org/10.1016/j.patcog.2023.109805

[20] Tissera, M. D., & McDonnell, M. D. (2016). Deep extreme learning machines: Supervised autoencoding architecture for classification. Neurocomputing, 174, 42–49. https://doi.org/10.1016/j.neucom.2015.03.110

[21] Sun, K., Zhang, J., Zhang, C., & Hu, J. (2017). Generalized extreme learning machine autoencoder and a new deep neural network. Neurocomputing, 230, 374–381. https://doi.org/10.1016/j.neucom.2016.12.027

[22] Chegni, A. M., Ghavami, B., & Eftekhari, M. (2022). A GPU-based accelerated ELM and deep-ELM training algorithms for traditional and deep neural networks classifiers. Intelligent Systems with Applications, 15, 200098. https://doi.org/10.1016/j.iswa.2022.200098

[23] Ouyang, T. (2021). Feature learning for stacked ELM via low-rank matrix factorization. Neurocomputing, 448, 82–93. https://doi.org/10.1016/j.neucom.2021.03.110

[24] Xia, Y., Yi, W., & Zhang, D. (2022). Coupled extreme learning machine and particle swarm optimization variant for projectile aerodynamic identification. Engineering Applications of Artificial Intelligence, 114, 105100. https://doi.org/10.1016/j.engappai.2022.105100

[25] Bardhan, A., Samui, P., Ghosh, K., Gandomi, A. H., & Bhattacharyya, S. (2021). ELM-based adaptive neuro swarm intelligence techniques for predicting the California bearing ratio of soils in soaked conditions. Applied Soft Computing, 110, 107595. https://doi.org/10.1016/j.asoc.2021.107595

[26] Sinha, N. J., Kloxin, C. J., Saven, J. G., Jensen, G. V., Kelman, Z., & Pochan, D. J. (2021). Recombinant expression of computationally designed peptide-bundlemers in Escherichia coli. Journal of Biotechnology, 330, 57–60. https://doi.org/10.1016/j.jbiotec.2021.03.004

[27] Gaglione, R., Pane, K., Dell'Olmo, E., Cafaro, V., Pizzo, E., Olivieri, G., Notomista, E., & Arciello, A. (2019). Cost-effective production of recombinant peptides in Escherichia coli. New Biotechnology, 51, 39–48. https://doi.org/10.1016/j.nbt.2019.02.004

[28] Yurina, V. (2018). Live Bacterial Vectors—A Promising DNA Vaccine Delivery System. Medical Sciences, 6(2), 27. https://doi.org/10.3390/medsci6020027

[29] Yurina, V., Rahayu Adianingsih, O., & Widodo, N. (2023). Oral and intranasal immunization with food-grade recombinant Lactococcus lactis expressing high conserved region of SARS-CoV-2 spike protein triggers mice's immunity responses. Vaccine: X, 13, 100265. https://doi.org/10.1016/j.jvacx.2023.100265

[30] Fraihat, S., & Al-Betar, M. A. (2023). A novel lossy image compression algorithm using multi-models stacked AutoEncoders. Array, 19, 100314. https://doi.org/10.1016/j.array.2023.100314

[31] Das, L., Garg, D., & Srinivasan, B. (2020). NeuralCompression: A machine learning approach to compress high frequency measurements in smart grid. Applied Energy, 257, 113966. https://doi.org/10.1016/j.apenergy.2019.113966

[32] Subray, S., Tschimben, S., & Gifford, K. (2021). Towards Enhancing Spectrum Sensing: Signal Classification Using Autoencoders. IEEE Access, 9, 82288–82299. https://doi.org/10.1109/ACCESS.2021.3087113

[33] Abrar, S., & Samad, M. D. (2022). Perturbation of deep autoencoder weights for model compression and classification of tabular data. Neural Networks, 156, 160–169. https://doi.org/10.1016/j.neunet.2022.09.020