Novel fuzzy approach to Antimicrobial Peptide Activity Prediction: A tale of limited and imbalanced data that models won't hear

Anonymous Author(s)

Abstract

Antimicrobial peptides have gained immense attention in recent years due to their potential for developing novel antibacterial medicines, next-generation anti-cancer treatment regimes, etc. Owing to the significant cost and time required for wet lab-based AMP screening, researchers have framed the task as an ML problem. However, traditional models rely on the unrealistic premise of large medical data availability to achieve significant performance levels; otherwise, they overfit, decreasing model precision. The collection of such labeled medical data is a challenging and expensive task in itself. The current study is the *first* to examine models in a real-world setting, training them on restricted and highly imbalanced data. A 'Fuzzy Intelligence' based model is proposed for short (< 30 aa) AMP activity prediction, and its ability to learn on limited and severely skewed high-dimensional space mapping is demonstrated over a set of experiments. The proposed model significantly outperforms state-of-the-art ML models trained on the same data. The findings demonstrate the model's efficacy as a potential method for *in silico* AMP activity prediction.

1 Introduction

Most organisms generate antimicrobial peptides as part of their innate immune response to pathogens. They are a significant resource for drug development since they represent a broad repertory of antimicrobial agents. To some extent, recent advances in peptide engineering and careful optimization have helped overcome AMP's toxicity to human cells and its unstable nature [1]. However, the high expense of peptide screening makes the AMP production process time-consuming and costly. Thus, *in silico* methods were developed. However, these require large volumes of costly labeled datasets to achieve significant levels of performance. Moreover, most deep learning and machine learning models require a balanced dataset to train. In actual practice, collecting a large number of positive samples is a difficult task in itself. The present work is the *first* to address these issues formulating Antimicrobial peptide prediction task as a 'fuzzy' problem. The proposed method outperforms other state-of-the-art ML models by a significant margin. *Second,* the model's other novelty lies in its capacity to attain high classification accuracy despite being trained on limited data samples, as proven experimentally, in contrast to previously proposed models that need large amounts of data that is difficult and costly to obtain. *Third*, another unique feature of the model lies in its robustness to high-class imbalance, commonly seen in real-world bioinformatics data, as demonstrated in the experiments performed. Most models, on the other hand, perform inconsistently as the imbalance in classes increases.

2 Proposed Methodology

Mapping the $n-$dimensional feature space. The generated feature vectors are mapped in the $n-$dimensional feature space. For each sample, the feature vector is passed to the Classifying Neurons (CLN), which perform the classification of the learned data using min-max hyperboxes [2]. In this

Submitted to NeurIPS 2021 AI for Science Workshop

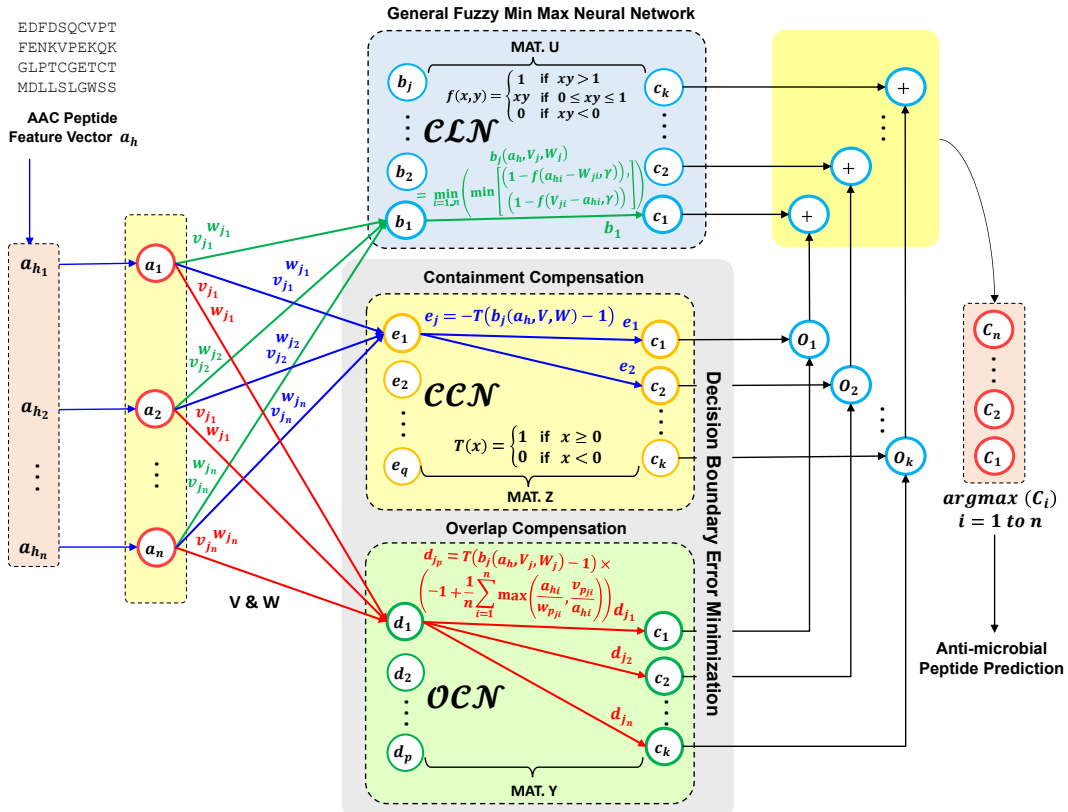

Figure 1: The proposed model architecture for anti-microbial peptide activity prediction

space, discontinuous decision boundaries in the form of 'Hyperboxes $\mathbf{H}$' are created during model training [3]. Defining this decision boundary requires a **min** coordinate $\mathbf{V_j} = (v_{j1}, v_{j2}, ..., v_{jn})$, and a **max** coordinate $\mathbf{W_j} = (w_{j1}, w_{j2}, ..., w_{jn})$. However, to reduce the number of model hyperparameters, a single parameter i.e., $\theta \in (0, 1)$ is defined which represents the hyperbox size. During training, for the first input, a point $\mathbf{H}$ is created. Subsequently, the model tries to accommodate remaining training samples $\{a_h, C_i\}$ in the previously constructed hyperboxes belonging to the same class C_i, provided the $\mathbf{H}_{size}$ does not exceeds a specified maximum limit ('θ') [2]. Else, a new $\mathbf{H}$ is added to $\mathcal{CLN}$, i.e., for a new training sample $\{a_h, C_i\}$, a hyperbox $\{b_j, C_j\}$ is found such that $C_j = C_i$ or $C_j = C_o$ which has the highest membership value and satisfying- (1) $\theta_{\max} \geq \frac{1}{n}\sum_{i=1}^{n}(\max(w_{ji}, a_{hi}) - \min(v_{ji}, a_{hi}))$, (2) b_j is not associated with any $\mathcal{OCN}/\mathcal{CCN}$ and (3) if $C_i = C_0$ or $C_j = C_0$ then $\mu_j > 0$. Following, the min-max coordinates of b_j are adjusted such that, $V_{ji}^{new} = \min(V_{ji}^{old}, a_{hi})$; $W_{ji}^{new} = \max(W_{ji}^{old}, a_{hi})$, where $i = 1, 2, ..., n$, and *third*, if $C_j = C_0$ and $C_i \neq C_0$ then $C_j = C_i$. Moreover, if no suitable b_j is present then a novel point hyperbox $\mathcal{H}$ for class C_i is created with $V_j = W_j = a_h$.

In CLNs, the neuron b_j represents hyperbox fuzzy set i.e., $= A_h, V_j, W_j, f(A_h, V_j, W_j)\ \forall\ (A_h \in I_n)$. In the middle layer of the classifier, the input nodes and the hyperbox nodes are connected together. These connections represent the min-max coordinates V and W of the $n-$ dimensional hyperbox fuzzy set [2]. During training, the neurons in the middle layer are created dynamically. Connection between the hyperbox node b_j to a class node C_j is represented by matrix U, where, $u_{ij} = 1$ if $b_j \in C_j$ else $u_{ij} = 0$. In CLN nodes, to compute the class memberships, activation function by [4] is used to assign membership value $= 1$ when the test sample falls within the hyperbox. In other cases, when the test sample lies outside $\mathbf{H}$, the model calculates the membership-value on the basis of its distance from the extreme coordinates of $\mathbf{H}$. Increasing the Fuzziness control parameter (γ) leads to more fuzzy classification, while decreasing it leads to a crisp classification.

Decision Boundary Error Minimization. Since the high dimensional feature space contains all the learned peptide features, a possible case of $\mathbf{H}$ overlap can occur. A reflex section is introduced which contains the Overlap Compensation Neurons ($\mathcal{OCN}$) and Containment Compensation Neurons ($\mathcal{CCN}$) which becomes active only when a case of hyperbox overlap and containment is encountered respectively [5]. If $V_{ki} < W_{ki} < V_{ji} < W_{ji}$ or $V_{ji} < W_{ji} < V_{ki} < W_{ki}$ is true for any $i \in 1, ..., n$, then hyperboxes (b_k, b_j) are isolated. If the isolation condition does not hold, the feature space must be checked for containment, i.e., if $V_{ki} < V_{ji} < W_{ji} < W_{ji}$ or $V_{ji} < V_{ki} < W_{ki} < W_{ji}$ is true for any $i \in 1, ..., n$, then Hyperboxes are contained. In such case, a $\mathcal{CCN}$ node is formed dynamically. If hyperboxes are not contained, then an $\mathcal{OCN}$ node is created. Moreover to check for hyperbox overlap, initially the value of δ^{old} is set as 1. A set of 4 cases are formed in for overlap in which, if $v_{ji} < v_{ki} < w_{ji} < w_{ki}$, then $\delta^{new} = \min(w_{ji} - v_{ki}, \delta^{old})$. Second, if $v_{ki} < v_{ji} < w_{ki} < w_{ji}$, then $\delta^{new} = \min(w_{ki} - v_{ji}, \delta^{old})$. Third, if $v_{ji} < v_{ki} \leq w_{ki} < w_{ji}$, then $\delta^{new} = \min(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{old})$. Fourth, $v_{ki} < v_{ji} \leq w_{ji} < w_{ki}$, then $\delta^{new} = \min(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{old})$. If overlaps exist (i.e., one of the above condition is true) and $(\delta_{new} - \delta_{old}) > 0$, then, $\Delta = i$ **else** $\Delta = -1$ [5]. For checking hyperbox containment, if overlap exists and is minimum along Δ dimension, the hyperboxes are contracted using the following given conditions i.e., if $v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta}$, then, $v_{k\Delta}^{new} = w_{j\Delta}^{new} = {}^{(w_{j\Delta}^{old} + v_{k\Delta}^{old})}/2$; second, if, $v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta}$, then $v_{k\Delta}^{new} = w_{j\Delta}^{new} = {}^{(w_{k\Delta}^{old} + v_{j\Delta}^{old})}/2$; third, $v_{k\Delta} < v_{j\Delta} \leq w_{j\Delta} < w_{k\Delta}$ and $w_{k\Delta} - v_{j\Delta} < w_{j\Delta} - v_{k\Delta}$, then $v_{j\Delta}^{new} = w_{k\Delta}^{old}$ else $w_{j\Delta}^{new} = v_{k\Delta}^{old}$; fourth, $v_{j\Delta} < v_{k\Delta} \leq w_{k\Delta} < w_{j\Delta}$ and $w_{k\Delta} - v_{j\Delta} < w_{j\Delta} - v_{k\Delta}$, then $w_{j\Delta}^{new} = v_{k\Delta}^{old}$ else $v_{j\Delta}^{new} = w_{k\Delta}^{old}$ [5]. The Reflex mechanism is biologically inspired from that of the human brain which takes control of the body in hazardous conditions unconsciously. The Reflex mechanism helps in obtaining more explainable class memberships with high accuracy.

3 Experimental Settings

Dataset. Dataset was taken from [1, 6] to access the performance of the proposed model. We calculated amino acid features based on its composition, following which, the redundant peptide sequence samples with > 0.99 similarity were removed from the dataset.

Baseline Model Training. The work is implemented using Keras [7] with Tensorflow [8] as backend. Nvidia K80 GPU with 12GB RAM workbench was used for conducting the experiments. Various state-of-the-art ML classifiers are implemented on the same dataset to compare the classification results. 'Accuracy' was used as the metric for optimizing the hyperparameters used for training. 'zscore' was used as the normalization method which is calculated as $z = (x - u)/s$. 'yeo-johnson' transformation was applied while training the ML classifiers for comparison.

Evaluation Metrics. Accuracy along with Matthews Correlation Coefficient (MCC) is used as the evaluation metric. MCC $\in [-1, 1]$ accounts for true and false positives and negatives, and is widely considered as a fair metric that can be applied to datasets with imbalanced classes and limited training samples [9]. A perfect forecast has a coefficient of $+1$, an average random prediction has a coefficient of 0 and an inverse prediction has a coefficient of -1.

4 Experiments, Results and Discussion

Exp-01: Limited Data Subset Configuration. In the first set of experiments, the model is trained on limited data subset configurations, i.e., $n = 300, 200, 100$, where n is the number of samples. The obtained results are compared with state-of-the-art ML models trained on the same number of data samples (see Table 01). Here, accuracy and MCC were used as the evaluation metrics for performance comparison. Further, it is to be noted that the same data samples, feature selection, and pre-processing techniques are used while implementing the baseline models for a fair comparison of results. From Table 1, it is evident that the proposed model outperforms all other models for $n = 300$ and 200. This demonstrates the strong ability of the model to achieve high performance on limited datasets. Moreover, most ML models demonstrate inconsistent performance with variations in dataset size. Though Support Vector Machine achieves comparable accuracy (although a low MCC score than the proposed model) for $n = 300$, its performance drops sharply for $n = 200$, with an MCC reduction of $\approx 40\%$. In the case of stochastic gradient descent (SGD), the model has low performance (57% accuracy and a low MCC score of 0.15) on $n = 300$. However, for

Table 1: Comparative Results with state-of-the-art models on limited data subset configuration. Proposed model was tuned at Hyperparameters: i) $n = 300 : \theta = 0.40, \gamma = 2$, ii) $n = 200 : \theta = 0.38, \gamma = 2$, iii) $n = 100 : \theta = 0.42, \gamma = 2$

Model	$n =$ **300**		$n =$ **200**		$n =$ **100**	
	ACC	**MCC**	**ACC**	**MCC**	**ACC**	**MCC**
Linear Discriminant Analysis	0.47	−0.13	0.40	−0.19	0.50	0.00
Quadratic Discriminant Analysis	0.48	0.00	0.42	0.00	0.50	0.00
Stochastic Gradient Descent	0.57	0.15	0.55	0.07	**0.90**	**0.82**
Bagging Classifier	0.58	0.21	0.62	0.29	0.55	0.12
Random Forest	0.63	0.29	0.68	0.42	0.60	0.20
Decision Tree	0.65	0.31	0.65	0.35	0.55	0.10
K-Neighbors Classifier	0.68	0.37	0.65	0.33	0.70	0.40
Support Vector Machine	0.70	0.41	0.62	0.24	0.80	0.65
Proposed Model	**0.70**	**0.44**	**0.68**	**0.42**	0.75	0.58

Table 2: Comparative Results with state-of-the-art models on imbalanced data subset configuration (with $n = 200$). Proposed model was tuned at Hyperparameters: i) $\mathbf{Imb} = 0.40 : \theta = 0.85, \gamma = 2$, ii) $\mathbf{Imb} = 0.30 : \theta = 0.85, \gamma = 2$, iii) $\mathbf{Imb} = 0.20 : \theta = 0.10, \gamma = 2$.

Model	**Imb = 0.40**		**Imb = 0.30**		**Imb = 0.20**	
	ACC	**MCC**	**ACC**	**MCC**	**ACC**	**MCC**
Decision Tree	0.53	−0.05	0.57	−0.09	0.88	0.63
K-Neighbors Classifier	0.55	−0.06	0.62	−0.19	0.85	0.53
Bagging Classifier	0.55	−0.19	0.70	0.18	0.88	0.59
Support Vector Machine	0.55	−0.19	0.55	−0.28	0.85	0.53
Quadratic Discriminant Analysis	0.57	0.11	0.55	−0.28	0.88	0.57
Random Forest	0.65	0.23	0.70	0.10	0.93	0.76
Linear Discriminant Analysis	0.78	0.53	0.57	−0.16	0.85	0.53
Stochastic Gradient Descent	0.80	**0.67**	0.47	−0.22	0.78	−0.08
Proposed Model	**0.83**	0.63	**0.70**	**0.32**	**0.93**	**0.76**

$n = 100$, the metric suddenly increases to 90% accuracy with a 0.82 MCC score. This is due to the model over-fitting on limited training data. Furthermore, it is seen that as the number of training samples varies, even the second-best performing model loses consistency, i.e., for $n = 300$, SVM is the second best-performed model, its performance decreases for $n = 200$ where the Random Forest takes its position and similarly for $n = 100$, SGD performs the best (which was ironically 7th best-performed model for $n = 300$). This illustrates that no single ML model exhibits robust and consistent performance while training on limited data. On the contrary, the proposed model has consistent performance, demonstrated by a fairly constant metric on all subset configurations of limited data, compared to other models that either over-fit or fail to classify the complex feature space on small training data.

Exp-02: Imbalanced Data Subset Configuration. Most antimicrobial peptide activity prediction datasets suffer from high-class imbalance (example, $75\% - 65\%$ of positive class distribution) since positive samples are difficult to obtain compared to the negative ones. Most deep learning and ML models exhibit a reduction in performance on high class-imbalance. To demonstrate the model's robustness to class imbalance, we compare the model performance on three data subset configurations: *first* with a low imbalance with Imb = 0.40 (i.e., 40% of all samples represent positives), *second* with high skewness of 0.30 and *third* with severely imbalanced class distribution with just 20% of all samples being positives. The obtained results (see Table 02) are compared with ML models trained on the same subset configuration. Here, accuracy along with MCC is considered a better evaluation metric than accuracy alone. Lack of consistency is also evident on imbalanced data configuration, with the SGD being the worst performed model for Imb= 0.30, while the same demonstrating second-best performance on the subset with Imb= 0.40. On the other hand, the proposed model shows consistent performance over all subset configurations.

5 Conclusion and Future Work

In this study, we present a fuzzy approach towards Antimicrobial peptide activity prediction. The study demonstrates the effectiveness and consistency of the proposed model compared to currently used state-of-the-art ML models. The presented model is capable of training on sparse datasets with high-class imbalance (as demonstrated through experiments). The proposed model significantly outperforms state-of-the-art ML models trained on the same data. The obtained results establish the model's effectiveness and quantify that fuzzy classifier-based models are more suited towards problems where the dataset is highly imbalanced or limited. Currently, the present study is focused on amino acid compositions that consider the entire peptide sequence as the feature. In the future, we plan to investigate the encoding percentage composition frequency of dipeptide, tripeptide, etc.

References

[1] Yan, J., Bhadra, P., Li, A., Sethiya, P., Qin, L., Tai, H. K., ... & Siu, S. W. (2020). Deep-AmPEP30: improve short antimicrobial peptides prediction with deep learning. Molecular Therapy-Nucleic Acids, 20, 882-894.

[2] Simpson, P. K. (1992). Fuzzy Min—MaX Neural NetWorks—Part 1: Classification. IEEE Trans. on Neural Networks, 3(5), 776-786.

[3] Alpern, B., & Carter, L. (1991). The hyperbox (pp. 133-139). IEEE.

[4] Gabrys, B., & Bargiela, A. (2000). General fuzzy min-max neural network for clustering and classification. IEEE transactions on neural networks, 11(3), 769-783.

[5] Nandedkar, A. V., & Biswas, P. K. (2007). A General Reflex Fuzzy Min-Max Neural Network. Eng. Lett., 14(1), 195-205.

[6] Dataset - AxPEP. (n.d.). Cbbio.Online. Retrieved September 26, 2021, from https://cbbio.online/AxPEP/?action=dataset

[7] Chollet, F. (2015). Keras. GitHub. https://github.com/fchollet/keras

[8] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

[9] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics, 21(1), 1-13.

A t-SNE Feature Visualization Maps

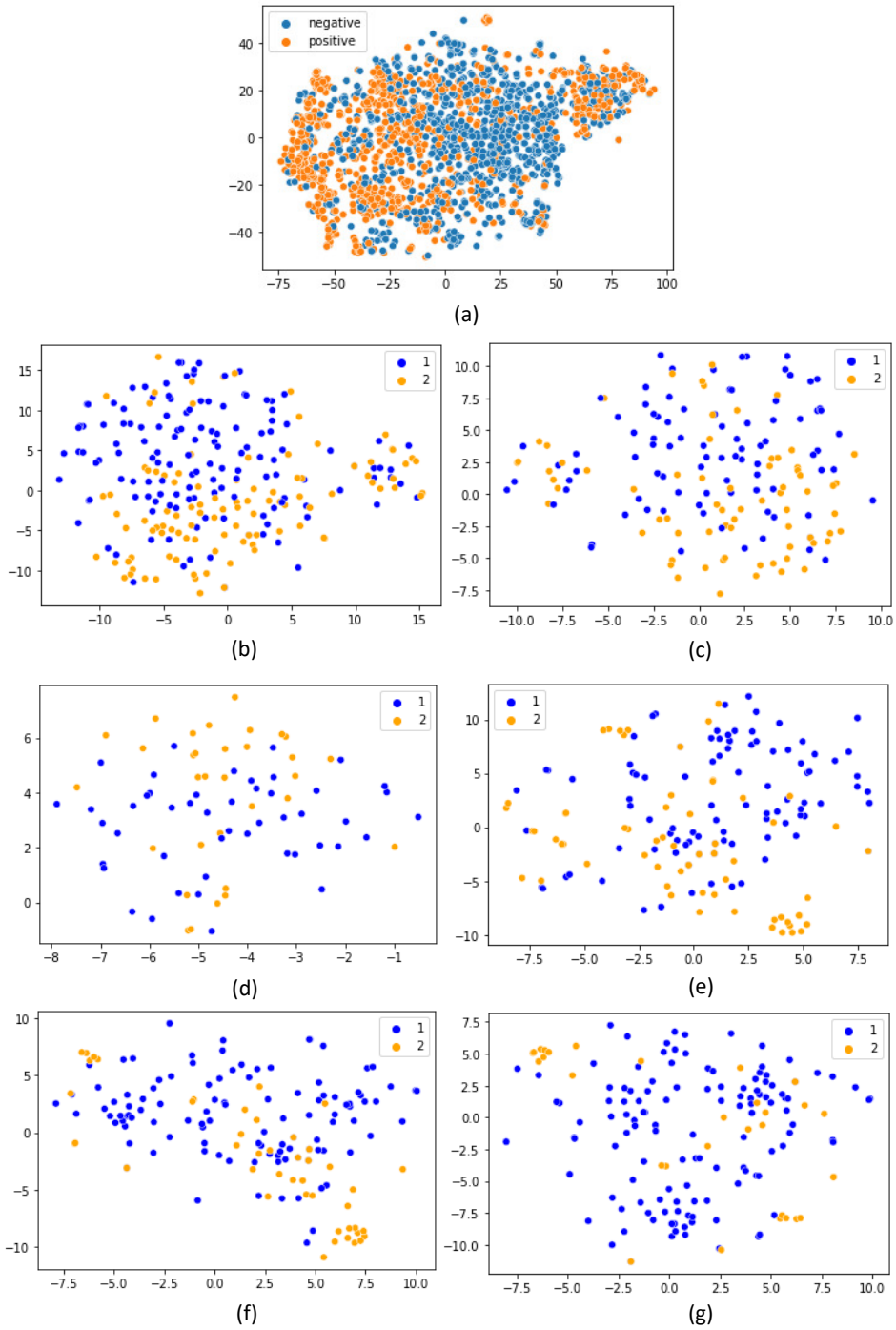

Figure 2: t-SNE feature visualizations for (a) complete dataset; limited training samples (b) $n = 300$ (c) $n = 200$ (d) $n = 100$; imbalanced subset samples (e) Imb $= 0.40$ (f) Imb $= 0.30$ (g) Imb $= 0.20$