Journal of Mathematics and Scientific Computing with Applications

E-ISSN : 2580-5754; P-ISSN : 2580-569X
Volume 7, Number 1, 2026
DOI: 10.53806/jmscowa.v7i1.1557
Page: 139–155

GANATCpair: GENERATIVE HARD NEGATIVE SAMPLING FOR IMPROVING DRUG-TARGET INTERACTION PREDICTION AGAINST SNAKE VENOM TARGETS

Said Thaufik Rizaldi[1], **Toto Haryanto**[1], **Fajar Sofyantoro**[4], **Wisnu Ananta Kusuma**[1,2,3]

[1]School of Data Science, Mathematics and Informatics, IPB University, Indonesia
[2]Bioinformatics Study Program, Faculty of Mathematics and Natural Sciences, IPB University, Indonesia
[3]Tropical Biopharmaca Research Center, IPB University
[4]Faculty of Biology, Universitas Gadjah Mada, Indonesia

Article Info

Article history:
Received : April 12, 2026
Revised : May 20, 2026
Accepted : July 05, 2026

Keywords:
Bioactive Compounds;
Drug-Target Interaction Prediction;
Generative Adversarial Network;
Hard Negative Sampling;
Snake Venom Targets.

ABSTRACT

Predicting drug-target interactions (DTIs) remains challenging because experimentally verified non-interaction data are rarely available for bioactive compounds and snake venom targets. This study proposes GANATCpair (Generative Adversarial Network with ATC Pairing), a GAN-based hard-negative sampling strategy for selecting unknown-status drug-target pairs. GANATCpair learns latent representations of DTI patterns and uses synthetic vectors as anchors to select informative unknown-status pairs near the positive decision region as putative negatives. The strategy was evaluated using the Snakebite Envenoming Medicines Database and four Yamanishi benchmark datasets across SVM, RF, XGBoost, and MLP classifiers. On the Snakebite dataset, GANATCpair improved AUC from 0.626-0.773 under random sampling to 0.916-0.952. The observed differences were supported by paired statistical analysis, and its computation was mainly driven by generative training and latent-space mapping. These findings suggest that GANATCpair strengthens hard-negative sampling and supports computational screening and drug–target prioritisation in snakebite envenoming.

Corresponding Author:
Wisnu Ananta Kusuma,
School of Data Science, Mathematics and Informatics
IPB University
Email: ananta@apps.ipb.ac.id

1. INTRODUCTION

Snakebite envenoming is a major socio-economic and public health problem, particularly in tropical and developing countries [1]. The effects of snakebite can cause serious complications, including haemorrhage, kidney failure, and paralysis, with approximately 2 million cases of envenoming and tens of thousands of deaths reported [2]. These clinical manifestations are largely mediated by toxic protein components of snake venom, including phospholipase A_2 (PLA_2) and snake venom metalloproteinases (SVMPs) [3]. Although antivenom remains the primary treatment, its application is limited by high production costs [4] and the extensive diversity of venom composition across snake species [5]. Therefore, a computational drug-target interaction prediction approach is needed to prioritize bioactive compounds with potential inhibitory activity against snake venom toxin targets.

Recent advances in computational drug discovery, particularly artificial intelligence and machine learning-based approaches, have improved the efficiency of prioritizing bioactive compounds with potential activity against snake venom toxins [6]. Drug-target interaction (DTI) prediction is increasingly used as virtual screening to narrow the search space of candidate compound target pairs and support drug repurposing [7], [8], [9]. Previous studies have investigated plant-derived compounds, including flavonoids, as potential inhibitors of snake venom metalloproteinases (SVMP) [3]. In addition, recent machine learning-integrated workflows combining stacked autoencoder-based deep neural networks (SAE-DNN), XGBoost (XGB) and Random Forest (RF), molecular docking, and molecular dynamics simulations have identified plant-derived compounds predicted to interact with and potentially inhibit snake venom phospholipase A_2 (PLA) [10]. However, supervised ML-DTI models require both positive and negative drug-target pairs for training, highlighting the need for a more informative negative sampling strategy.

Machine learning-based drug-target interaction (DTI) prediction is still limited by the lack of experimentally validated non-interaction data [7], [11]. Positive interactions are typically supported by experimental evidence. In contrast, true negative compound target pairs are rarely found in DTI databases [12]. As a practical response, random negative sampling is often used. This approach constructs supervised learning datasets by treating unobserved or unknown-status pairs as putative negatives. This method is common in snake venom-related DTI and peptide–protein interaction studies [10], [13], [14]. Accordingly, random sampling was selected as the controlled baseline because it permits direct comparison between unguided and GAN-guided pair selection while preserving matched sampling ratios and identical downstream evaluation settings.

However, random negative sampling can introduce false negatives and sampling bias because unobserved pairs may represent undiscovered positive interactions [7], [15]. Distinct methodological paradigms address the scarcity of experimentally confirmed negatives. Reliable-negative extraction prioritises candidates unlikely to belong to the positive class, whereas positive unlabeled learning estimates predictive risk or class information from positive and unlabeled observations. Hard-negative sampling instead selects challenging candidates near the decision boundary to provide stronger discriminative information than easily separable negatives [16], [17]. Although addressing the same data limitation, these paradigms differ in their objectives, assumptions, and roles within the modelling pipeline.

Recent GAN-based DTI studies indicate that generative learning has been increasingly explored to address data-related challenges in DTI modelling. GAN-generated synthetic samples have been used to balance imbalanced DTI datasets before classifier training [18], whereas VGAN-DTI integrates GANs, VAEs, and Multilayer Perceptron (MLP) to generate molecular candidates and improve interaction prediction [19]. However, these developments primarily focus on data balancing, molecular generation, or prediction enhancement, and have not specifically addressed the construction of informative putative negative samples from unknown-status compound-target pairs.

Adversarial negative sampling in contrastive learning has been formulated through adversarially trained negative samplers [20], trainable negative adversaries [21], and jointly learnable positive and negative samples [22]. Although their implementations differ, these approaches generally integrate adversarial negatives into contrastive objectives to improve the discriminative quality of learned representations. Their methodological focus therefore lies in optimising the representation space during model training rather than constructing a separate set of putative negative pairs for downstream supervised DTI prediction.

This study proposes GANATCpair, a generative adversarial network (GAN)-based hard-negative sampling strategy for selecting informative unknown-status compound-target pairs involving snake venom targets. Unlike adversarial negative sampling in contrastive learning, which uses adversarial negatives within representation-learning objectives, GANATCpair uses GAN-derived latent anchors to select existing unknown-status compound-target pairs before downstream supervised classifier training. GANATCpair also differs from positive-unlabeled learning, which directly incorporates positive and unlabeled observations into a specialised learning objective [23].

In contrast, GANATCpair selects a targeted subset of existing unknown-status pairs before downstream supervised classifier training. This design supports informative putative negative-pair construction without relying on random negative selection. The strategy was evaluated using a snakebite envenoming dataset from the Snakebite Envenoming Medicines Database [4] and the Yamanishi benchmark dataset [24]. The contributions of this study

are: (1) developing a GAN-guided strategy for constructing putative negative samples; (2) evaluating GANATCpair on snakebite-specific and benchmark DTI datasets; and (3) assessing its performance across Support Vector Machine (SVM), RF, XGBoost, and MLP under random 1:1 and 1:2 sampling baselines.

2. RESEARCH METHOD

This section outlines the research workflow for developing and evaluating DTI prediction models. As shown in Figure 1, the workflow comprises five main stages: (1) data acquisition; (2) the generation of putative negative samples from unknown-status pairs using GANATCpair; (3) the extraction of protein and compound features; (4) the training of predictive models; and (5) the evaluation of model performance across the experimental scenarios.

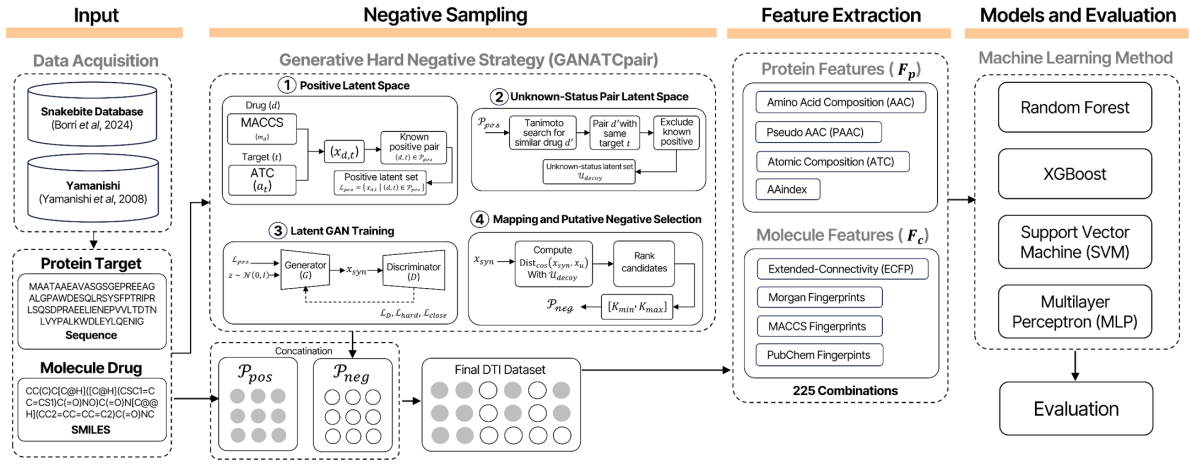

Figure 1. Workflow of the methodology

2.1. Data Acquisition

This study utilised data from the Snakebite Envenoming Medicines Database, which comprises over 324 entries of compounds and therapeutic agents tested against snake venom toxins [4]. The analysis was restricted to active small-molecule compounds with available Simplified Molecular Input Line Entry System (SMILES) identifiers in PubChem [25], as these representations were required for compound feature extraction. After excluding entries without retrievable SMILES and removing invalid or duplicated compound-target records, the final Snakebite dataset contained 102 positive compound-protein interaction pairs, involving 44 unique compounds and 42 unique protein targets.

Prior to model construction, the selected compound-target records were further screened for missing values, valid target protein sequences, and duplicated pairs. Records with missing target identifiers, duplicated interactions, or unprocessable protein sequences were excluded, while valid sequences were filtered to retain only standard amino acid residues. The Yamanishi benchmark dataset [24], available at http://web.kuicr.kyoto-u.ac.jp/supp/yoshi/drugtarget/, was then used to evaluate the general DTI prediction setting, with dataset details presented in Table 1. Both datasets were accessed on 12 November 2025.

Table 1: Details of the Yamanishi benchmark dataset

No	Data	Sample Size
1	Enzyme	2924 samples
2	Ion channel	1477 samples
3	G-protein-coupled receptors	636 samples
4	Nuclear receptor	91 samples

2.2. GANATCpair

GANATCpair is a hard-negative sampling strategy developed to select putative negative pairs from an unknown-status compound-target pair space for DTI prediction. This approach utilises the GAN mechanism [26] to learn the latent features of positive interaction pairs and generate synthetic latent vectors that guide the selection of unknown-status pairs. GANATCpair consists of four phases: (1) positive latent space construction; (2) unknown-status pair space construction; (3) latent-space GAN training; and (4) mapping to the unknown-status pair space. Following the mapping process, the selected putative negative pairs are combined with known positive pairs to construct the final supervised DTI dataset.

2.2.1 Positive Latent Space Construction

This step extracts features from each positive interaction pair. Compounds are represented using a SMILES-based MACCS fingerprint [27], whilst proteins are represented using atomic and bond composition (ATC) features [28] based on amino acid sequences. The MACCS vector (m_d) of the compound and the ATC vector (a_t) of the target protein are then concatenated to form the drug-target pair representation ($x_{d,t}$), as expressed in Equation 1.

$$x_{d,t} = [m_d \parallel a_t] \tag{1}$$

All pairs of positive interactions then form a set of positive latent vectors. If $\mathcal{P}_{pos}$ denotes a drug-target pair known to interact, its latent representation is formulated in Equation 2 in the next step.

$$\mathcal{L}_{pos} = \{x_{d,t} \mid (d,t) \in \mathcal{P}_{pos}\} \tag{2}$$

2.2.2 Unknown-Status Pair Space Construction

This step constructs the unknown-status pair space as a source of candidate putative negative samples. For each positive pair (d,t), compounds structurally related to d are identified using the Tanimoto coefficient [29]. This measure is appropriate for MACCS fingerprints because they are represented as binary molecular feature vectors, and the Tanimoto coefficient quantifies similarity based on the ratio of shared active features to the union of active features across the two compounds. If m_{d_i} and m_{d_j} denote the MACCS fingerprints of two compounds, their similarity score is calculated as shown in Equation 3.

$$\mathrm{Tan}(d_i, d_j) = \frac{m_{d_i}^T m_{d_j}}{\|m_{d_i}\|_1 + \left\|m_{d_j}\right\|_1 - m_{d_i}^T m_{d_j}} \tag{3}$$

The compound d' with the highest $\mathrm{Tan}(d,d')$ score is then paired with the same target, provided that the pair (d',t) has not yet been recorded as a positive interaction. Pairs that meet these criteria form the unknown-status pairs $\mathcal{U}_{decoy}$, which is defined in Equation (4) as a set of decoy candidates for the formation of putative negative samples.

$$\mathcal{U}_{decoy} = \left\{(d',t) \,\middle|\, (d,t) \in \mathcal{P}_{pos},\ d' \in \mathrm{TopK}_{d'}\big(\mathrm{Tan}(d,d')\big),\ (d',t) \notin \mathcal{P}_{pos}\right\} \tag{4}$$

$\mathrm{TopK}_{d'}(\mathrm{Tan}(d,d'))$ identifies the set of compounds d' with the highest Tanimoto scores relative to the target compound d. Consequently, the unknown-status pairs ($\mathcal{U}_{decoy}$) are drug-target pairs that have not yet been recorded as positive interactions but are derived from compounds closely related to known interaction pairs.

2.2.3 Latent-Space GAN Training

In this stage, GAN training [26] is carried out using a set of positive latent vectors L_{pos}. The generator G receives a random vector z to generate a synthetic latent vector x_{syn}, whilst the discriminator D distinguishes this vector from the positive latent vectors x_{real}. The process of generating synthetic vectors in Equation 5 serves as the search for hard negative candidates in the subsequent stage.

$$x_{syn} = G(z), \qquad z \sim \mathcal{N}(0, I) \tag{5}$$

The discriminator is trained to assign outputs close to 1 to positive latent vectors and outputs close to 0 to synthetic latent vectors. The discriminator objective is constructed using binary cross-entropy loss, as formulated in Equation (6).

$$\mathcal{L}_D = -\frac{1}{N}\sum_{i=1}^{N}\left[\log D\left(x_{real}^{(i)}\right) + \log\left(1 - D\left(x_{syn}^{(i)}\right)\right)\right] \tag{6}$$

Unlike the discriminator objective, the generator objective is defined using two complementary terms: the hard negativity loss $\mathcal{L}_{hard}$ in Equation (7) and the closeness loss $\mathcal{L}_{close}$ in Equation (8).

$$\mathcal{L}_{hard} = \frac{1}{N}\sum_{i=1}^{N}\left(D(x_{syn}^{(i)}) - 0.5\right)^2 \tag{7}$$

The hard negativity loss $\mathcal{L}_{hard}$ in Equation (7) is formulated as a mean squared error objective with a target discriminator response of 0.5. For each synthetic latent vector $x_{syn}^{(i)}$, let $s_i = D(x_{syn}^{(i)})$, where $s_i \in [0, 1]$. The target response is defined as $s_i^* = 0.5$, which corresponds to the midpoint of the discriminator output range. The error with respect to this target is $e_i = s_i - s_i^* = s_i - 0.5$. Since this error can be positive or negative, the squared error e_i^2 is used to penalize deviations from the target in either direction. Averaging this squared error over N synthetic latent vectors yields the mean squared error form of $\mathcal{L}_{hard}$ shown in Equation (7).

$$\mathcal{L}_{close} = \frac{1}{N}\sum_{i=1}^{N} \max\left(0, \min_{x_{real}\in\mathcal{B}_{pos}} \left\| x_{syn}^{(i)} - x_{real} \right\|_2 - \tau\right) \tag{8}$$

The closeness loss $\mathcal{L}_{close}$ in Equation (8) penalizes synthetic latent vectors that are farther than the threshold τ from the nearest positive latent vector. For each synthetic latent vector $x_{syn}^{(i)}$, the nearest-neighbour distance to the positive batch $\mathcal{B}_{pos}$ is defined as $d_i = \min_{x_{real}\in\mathcal{B}_{pos}} \left\| x_{syn}^{(i)} - x_{real} \right\|_2$, where $\mathcal{B}_{pos}$ denotes the batch of positive latent vectors used during GAN training. The penalty is then computed using the positive-part function $\max(0, d_i - \tau)$. Therefore, no closeness penalty is added when $d_i \leq \tau$, whereas a linear penalty is added when $d_i > \tau$. This formulation encourages the generated synthetic vectors to remain within a tolerated neighbourhood of the positive latent distribution. The two generator loss components are then combined in the generator objective function, as stated in Equation (9).

$$\mathcal{L}_G = \mathcal{L}_{hard} + \lambda\mathcal{L}_{close} \tag{9}$$

The total generator objective $\mathcal{L}_G$ combines the hard negativity loss and the closeness loss. The parameter λ controls the contribution of $\mathcal{L}_{close}$ relative to $\mathcal{L}_{hard}$. In this formulation, $\mathcal{L}_{hard}$ encourages synthetic latent vectors to obtain intermediate discriminator responses, whereas $\mathcal{L}_{close}$ penalizes synthetic latent vectors that are farther than the tolerated distance τ from the positive latent batch. Thus, the joint objective favours latent anchors with intermediate discriminator responses while retaining proximity to the positive latent distribution, thereby guiding the selection of challenging unknown-status pairs that can provide informative constraints on the decision boundary of the downstream classifier.

For the main experiments, GAN training was performed for 150 epochs with a batch size of 32 and a 64-dimensional random vector z, while G and D were optimized using Adam with a learning rate of 1×10^{-4}. The closeness-loss weighting parameter λ in Equation (9) was set to the default value of 0.2 to balance $\mathcal{L}_{hard}$ and $\mathcal{L}_{close}$. To examine its influence around the default configuration, λ was evaluated at 0.1, 0.15, 0.2, 0.25, and 0.3, while all remaining GAN and experimental configurations were held constant to isolate its effect on negative-pair selection and downstream predictive performance. GAN behaviour was monitored through $\mathcal{L}_D$, $\mathcal{L}_G$, $\mathcal{L}_{hard}$, and $\mathcal{L}_{close}$. In the subsequent mapping stage, x_{syn} serves as a latent anchor for identifying corresponding unknown-status pairs, which are then used as putative negative samples in classifier training.

2.2.4 Mapping to the Unknown-Status Pair Space

The pairs in the unknown-status pair space $\mathcal{U}_{decoy}$ are transformed using the pair representation in Equation 1 to form the unknown-status latent space $\mathcal{L}_{unk}$, as expressed in Equation (10). This space serves as the candidate pool to be mapped to the synthetic latent vectors (x_{syn}) generated by the GAN training.

$$\mathcal{L}_{unk} = \{x_{d',t} \mid (d', t) \in \mathcal{U}_{decoy}\} \tag{10}$$

Each x_{syn} is then compared with candidates in $\mathcal{L}_{unk}$ using the cosine distance [30], as formulated in Equation (11). This measure is appropriate for latent-space mapping because x_{syn} and x_u are continuous high-dimensional pair representations, and cosine distance quantifies angular dissimilarity between vectors while reducing direct dependence on vector magnitude.

$$\text{Dist}_{cos}(x_{syn}, x_u) = 1 - \frac{x_{syn}^T x_u}{\|x_{syn}\|_2 \, \|x_u\|_2} \tag{11}$$

Candidates in $\mathcal{L}_{unk}$ are ranked in ascending order based on $\text{Dist}_{cos}(x_{syn}, x_u)$, where smaller distances indicate greater proximity to the GAN-derived synthetic latent anchor. In the main experiments, the ranking bounds were defined as $K_{min} = 3$ and $K_{max} = 30$, with K_{max} treated as the exclusive upper bound. These bounds specify a local ranking interval for candidate selection: K_{min} excludes the closest-ranked candidates, whereas K_{max} limits the search to a local neighbourhood of the anchor. This ranking interval preserves latent-space proximity while reducing dependence on the nearest candidates alone. The same ranking bounds were applied consistently across the Snakebite case study and the Yamanishi benchmark experiments to maintain an identical candidate-selection

rule across datasets. $\mathcal{P}_{neg}$ pairs are then selected from this interval, as stated in Equation (12). The pairs in $\mathcal{P}_{neg}$ are used as putative negative samples and are subsequently combined with the positive pairs to construct the DTI prediction dataset.

$$\mathcal{P}_{neg} = \{(d', t) \in \mathcal{U}_{decoy} \mid \text{Rank}(\text{Dist}_{cos}(x_{syn}, x_{d',t})) \in [K_{min}, K_{max})\} \tag{12}$$

The selected $\mathcal{P}_{neg}$ pairs were combined with known positive pairs to form the final DTI prediction dataset. For the Snakebite case study, the GANATCpair scenario comprised 204 compound-protein pairs, including 102 positive and 102 putative negative interactions, corresponding to a 1:1 positive-to-negative ratio. Class balance was therefore controlled during dataset construction by matching the number of putative negative pairs to the positive pairs. As baselines, Random 1:1 and Random 1:2 sampling were applied using equal and two-fold negative sampling ratios, respectively. The same sampling schemes were used for the Yamanishi benchmark datasets.

Algorithm 1: GANATCpair

Input: Known positive pairs $\mathcal{P}_{pos}$, compound set C, target set T, pair representation function $f(\cdot)$, number of putative negatives N_{neg}, ranking bounds K_{min} and K_{max}.
Output: Putative negative pair set $\mathcal{P}_{neg}$.
Initialize $\mathcal{L}_{pos} \leftarrow \emptyset$, $U \leftarrow \emptyset$, and $\mathcal{P}_{neg} \leftarrow \emptyset$.
for all $(c_i, t_j) \in \mathcal{P}_{pos}$ **do**
    $\mathcal{L}_{pos} \leftarrow \mathcal{L}_{pos} \cup \{f(c_i, t_j)\}$.
    Identify similar compounds $c_k \in C$ to c_i using the Tanimoto coefficient.
    for all selected c_k **do**
        if $(c_k, t_j) \notin \mathcal{P}_{pos}$ **then**
            $U \leftarrow U \cup \{(c_k, t_j)\}$.
        end if
    end for
end for
Construct $\mathcal{L}_{unknown} \leftarrow \{f(c, t) \mid (c, t) \in U\}$.
Train the GAN using $\mathcal{L}_{pos}$ as the real latent-vector distribution.
Generate synthetic latent vectors X_{syn} using the trained generator.
for all $x_{syn} \in X_{syn}$ **do**
    Compute $\text{Dist}_{cos}(x_{syn}, x_u)$ for each $x_u \in \mathcal{L}_{unknown}$.
    Rank candidates in $\mathcal{L}_{unknown}$ by ascending cosine distance.
    Select candidates with ranks $K_{min} \leq K < K_{max}$.
    Add the corresponding unknown-status pairs to $\mathcal{P}_{neg}$.
    if $|\mathcal{P}_{neg}| \geq N_{neg}$ **then**
        break
    end if
end for
Remove duplicate pairs from $\mathcal{P}_{neg}$.
Retain the first N_{neg} pairs in $\mathcal{P}_{neg}$.
return $\mathcal{P}_{neg}$.

The computational complexity of GANATCpair, following Algorithm 1, can be described according to its main stages: unknown-status candidate construction, GAN training, and latent-space mapping. Let n_p, n_c, m, and q denote the numbers of positive pairs, compounds, unknown-status candidates, and synthetic latent vectors, respectively. Let b be the fingerprint length used in Tanimoto computation, d the pair-representation dimension, E the number of GAN epochs, and θ_{GAN} the cost of one GAN update. Constructing the unknown-status candidate set requires $\mathcal{O}(n_p n_c b)$ time, while forming its latent representations requires $\mathcal{O}(md)$. GAN training requires $\mathcal{O}(En_p\theta_{\text{GAN}})$. During latent-space mapping, each synthetic vector is compared with m unknown-status candidates and the resulting distances are ranked, requiring $\mathcal{O}(md + m\log m)$ time per synthetic vector, or $\mathcal{O}(qm(d + \log m))$ for q synthetic vectors.

Thus, the overall time complexity can be expressed as $\mathcal{O}(n_p n_c b + md + En_p\theta_{\text{GAN}} + qm(d + \log m))$. The dominant space complexity is $\mathcal{O}(md)$, corresponding to the storage of unknown-status latent representations. Under the reported computational environment, the mean end-to-end runtime of the GANATCpair scenario construc-

tion across three runs was 0.029 minutes for Snakebite and ranged from 0.015 to 0.952 minutes across the four Yamanishi target groups. These measurements covered feature and candidate construction, GAN training, and latent-space mapping, excluding downstream classifier training.

2.3. Feature Extraction

The feature combinations evaluated comprised 225 pairs of protein and compound features, formed from all possible subsets of four protein features and four compound features. The protein features used included AAindex [31], Atomic and Bond Composition (ATC) and Amino Acid Composition (AAC) [28], and Pseudo Amino Acid Composition (PAAC) [32]. These four features represent the characteristics of residues and the proportions of amino acids in the protein sequence.

The compound features used include the PubChem Fingerprint [33], the MACCS Fingerprint [27], the Morgan Fingerprint and the Extended-Connectivity Fingerprint (ECFP) [34]. These features capture molecular substructural and topological patterns. All resulting feature combinations were subsequently used as input in classification experiments to assess the impact of variations in protein compound representations on DTI prediction performance.

2.4. Predictive Models

The predictive models used to evaluate the predictive performance of drug-target interactions were SVM, MLP, RF, and XGBoost. SVM was used to form a non-linear decision boundary through the application of a Radial Basis Function (RBF) kernel [35]. MLP is applied to learn non-linear patterns through an arrangement of hidden neural layers [36]. RF is used as a decision tree-based ensemble method that is robust in handling high-dimensional data [37]. XGBoost was selected because it combines a stepwise boosting mechanism and regularisation to control model complexity [38].

Grid search was used to obtain the optimal configuration for each classification algorithm using the discrete candidate values listed in Table 2. Continuous-valued hyperparameters, including C, γ, learning rate (η), and dropout rate, were evaluated at these predefined grid values instead of being sampled from a continuous distribution.

Table 2: Hyperparameter Search Space

Hyperparameter	**Value**
C^a	0.1, 1, 10, 100
γ^a	scale, auto, 0.01, 0.1
class_weight^a	none, balanced
HL_0 Node^d	256, 512
HL_i Node^d	128, 256
Hidden layer^d	2
LR / η^c	0.01, 0.05, 0.1
Learning rate^d	0.01
Dropout rate^d	0.3, 0.5
Epochs^d	100
Batch size^d	32
$n^{b,c}_{\text{estimators}}$	100, 300, 500

Note. ^{a} SVM; ^{b} RF; ^{c} XGBoost; ^{d} MLP. HL0 Node denotes the number of neurons in the first hidden layer, while HL_i Node denotes the number of neurons in the subsequent hidden layer(s).

2.5. Evaluation Method

The evaluation used Stratified 5-Fold Cross-Validation to preserve class proportions in each fold and obtain more stable performance estimates, particularly when evaluating models on limited-size datasets [39]. The Area Under the Curve (AUC) via the Receiver Operating Characteristic (ROC) curve illustrates the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR), and is therefore used to assess the model's ability to distinguish between drug-target pairs that interact and those that do not [40]. The Precision Recall (PR) curve is used to illustrate the trade-off between precision and recall at various classification thresholds.

The accuracy metric is used to measure the proportion of drug-target pairs classified correctly relative to the entire test dataset [40], as formulated in Equation 13.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

The precision metric is used to measure the proportion of correct predictions of positive interactions out of all pairs predicted as positive [41], as expressed in Equation (14).

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

The recall metric measures the model's ability to identify positive interaction pairs from all available positive pairs [39], as expressed in Equation (15).

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

The F1-score metric is used as the harmonic mean of *precision* and *recall*, thereby providing a performance measure that takes into account the balance between the two [42], as expressed in Equation (16).

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{16}$$

An ablation study was conducted to examine the contribution of the main GANATCpair components, namely Tanimoto-based decoy space construction, cosine-distance ranking, and GAN-based latent mapping for putative negative selection. Four variants were evaluated: unconstrained decoy selection, Tanimoto-based decoy selection, Tanimoto-based decoy selection with cosine-distance ranking, and the full GANATCpair framework. Each variant was evaluated using AUC and F1-score, with F1-score computed as defined in Equation 16, to quantify the effect of each component on predictive performance.

Statistical significance was evaluated using a paired two-sided Wilcoxon signed-rank test [43] on fold-level AUC values [44]. The test was selected for paired comparisons without assuming normality in the distribution of differences [45]. Paired comparisons were conducted using aligned fold-level AUC estimates, with alignment defined by classifier and cross-validation fold for Snakebite and by target group, classifier, and cross-validation fold for Yamanishi. For each aligned pair, ΔAUC was calculated as the AUC obtained with GANATCpair minus that obtained with the corresponding random-sampling baseline.

Sensitivity to λ_{close} was assessed using ROC-AUC and F1-score, with the F1-score calculated according to Equation (16). For each evaluated value of λ_{close}, changes in ROC-AUC and F1-score were calculated relative to the default configuration $\lambda_{close} = 0.2$. The resulting differences were denoted as ΔAUC_λ and ΔF1_λ, respectively. Negative values represent lower performance than the default configuration, whereas positive values represent higher performance.

2.6. Computational Resources

All experiments were conducted using two computing resources. Initial experiments were carried out on a High Performance Computing (HPC) server with an NVIDIA GeForce RTX 3070 GPU and 12 GB of VRAM, belonging to the Computer Science Program Study at IPB University. For more computationally intensive scenarios, particularly feature combination evaluation, hyperparameter search, and model tuning, an NVIDIA DGX H100 HPC server with 80 GB of VRAM was used via full GPU allocation.

The research was implemented using Python 3.10. The development of GANATCpair utilised NumPy, pandas, RDKit, protlearn, scikit-learn and PyTorch, whilst the classification modelling and evaluation process utilised NumPy, pandas, TensorFlow, scikit-learn and XGBoost. To ensure the reproducibility of results, the random seed was fixed at 42 for main experiments, whilst the preprocessing stages and evaluation procedures were applied uniformly across all test scenarios.

3. RESULT AND ANALYSIS

The first evaluation was conducted on the Snakebite Envenoming Medicines Database to assess the effectiveness of the negative pairing strategy on the predictive performance of drug-target interactions. Three scenarios were compared random sampling at a 1:1 ratio, random sampling at a 1:2 ratio, and GANATCpair, using four classification models SVM, RF, XGBoost, and MLP. The average cross-validation performance for all scenarios is presented in Table 3.

Table 3: Average cross-validation performance of methods on the Snakebite database

Method	Performance Measure	Value Random Sampling		GANATCpair
		1:1	1:2	
SVM	**AUC**	**0.639 ± 0.022**	**0.691 ± 0.029**	**0.943 ± 0.008**
	Accuracy	0.613 ± 0.066	0.690 ± 0.022	0.886 ± 0.011
	Precision	0.614 ± 0.069	0.774 ± 0.208	0.899 ± 0.026
	Recall	0.618 ± 0.108	0.169 ± 0.052	0.871 ± 0.039
	F1-score	0.611 ± 0.077	0.263 ± 0.052	0.884 ± 0.013
RF	**AUC**	**0.685 ± 0.026**	**0.773 ± 0.020**	**0.952 ± 0.004**
	Accuracy	0.652 ± 0.068	0.740 ± 0.022	0.916 ± 0.036
	Precision	0.648 ± 0.063	0.660 ± 0.049	0.948 ± 0.030
	Recall	0.658 ± 0.096	0.475 ± 0.062	0.882 ± 0.070
	F1-score	0.652 ± 0.078	0.550 ± 0.044	0.912 ± 0.040
XGBoost	**AUC**	**0.651 ± 0.038**	**0.730 ± 0.033**	**0.936 ± 0.011**
	Accuracy	0.608 ± 0.052	0.738 ± 0.028	0.897 ± 0.042
	Precision	0.596 ± 0.043	0.620 ± 0.068	0.936 ± 0.037
	Recall	0.658 ± 0.106	0.568 ± 0.048	0.853 ± 0.060
	F1-score	0.623 ± 0.068	0.591 ± 0.045	0.892 ± 0.044
MLP	**AUC**	**0.626 ± 0.031**	**0.645 ± 0.016**	**0.916 ± 0.003**
	Accuracy	0.588 ± 0.031	0.646 ± 0.026	0.892 ± 0.026
	Precision	0.588 ± 0.037	0.488 ± 0.043	0.909 ± 0.014
	Recall	0.587 ± 0.068	0.552 ± 0.160	0.871 ± 0.068
	F1-score	0.586 ± 0.043	0.499 ± 0.085	0.888 ± 0.032

The results in Table 3 show that GANATCpair achieved higher performance values across the evaluated models and metrics. The AUC values in the GANATCpair scenario range from 0.916 to 0.952, compared with 0.626-0.685 for Random Sampling 1:1 and 0.645-0.773 for Random Sampling 1:2. For all three scenarios, negative-pair construction was performed before classifier evaluation, and the labelled dataset produced by each strategy was subsequently evaluated using the same stratified five-fold cross-validation protocol. This pattern suggests that differences in negative-pair construction are associated with the model's ability to distinguish between positive and negative interactions. The putative negative pairs selected through GANATCpair were associated with higher downstream discriminative performance than those obtained through random sampling.

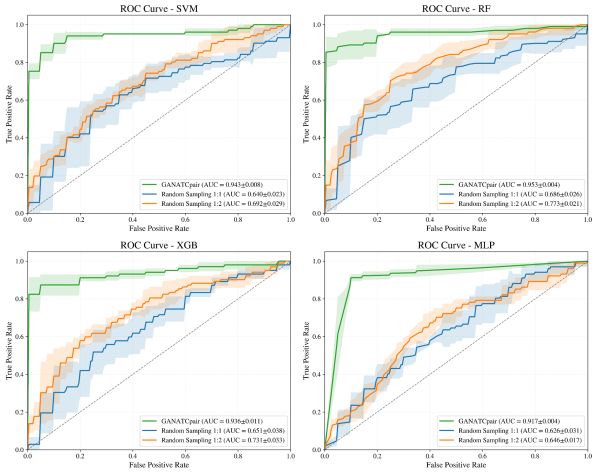

Figure 2. ROC curves of various models on the snakebite database

The Random Forest (RF) with GANATCpair delivered the best overall performance, with an AUC of 0.952 ± 0.004, accuracy of 0.916 ± 0.036, precision of 0.948 ± 0.030, recall of 0.882 ± 0.070, and an F1-score of 0.912 ± 0.040. These values indicate that RF not only possesses high class discrimination capability but also maintains a good balance between positive prediction accuracy and interaction detection sensitivity. High performance was also

achieved by SVM and XGBoost, with AUCs of 0.943 ± 0.008 and 0.936 ± 0.011, respectively, whilst the MLP achieved an AUC of 0.916 ± 0.003. Although the MLP produced the lowest performance among the GANATCpair models, its results were still superior to all random sampling scenarios.

The differences between the scenarios become increasingly apparent in the Random Sampling 1:2 setting, particularly for SVM. In this scenario, SVM achieves relatively high precision, but this is accompanied by a low recall of 0.169 ± 0.052 and an F1-score of 0.263 ± 0.052. Because recall and F1-score are threshold-dependent metrics, these results should be interpreted in relation to the fixed classification threshold used consistently across all sampling scenarios and classifiers. The SVM AUC under Random Sampling 1:2 increased from 0.639 to 0.691 relative to Random Sampling 1:1, indicating a difference between threshold-dependent metrics and threshold-independent discrimination. In comparison, GANATCpair yields higher SVM recall and F1-score values of 0.871 ± 0.039 and 0.884 ± 0.013, respectively. Similar F1-score improvements are also observed for RF, XGBoost, and MLP, suggesting that the negative-pair construction strategy influences the balance between positive detection and classification precision across models [17], [46].

The quantitative results in Table 3 are supported by the ROC curves shown in Figure 3. The GANATCpair curve is consistently closer to the top-left corner across all models, indicating better class discrimination across various decision thresholds. In the best validation fold visualised, the GANATCpair AUC values reached approximately 0.941 for SVM, 0.948 for RF, 0.935 for XGBoost, and 0.911 for MLP. In contrast, the random sampling curve lies closer to the diagonal line, particularly for SVM and MLP, indicating weaker class separation. This difference aligns with the mean AUC values in Table 3 and demonstrates that the performance advantage of GANATCpair is consistently evident in both the statistical summary and the visualisation across classification thresholds.

The Precision-Recall (PR) curve in Figure 3 provides a more detailed picture of the models' ability to maintain the quality of positive class predictions. GANATCpair exhibits a more stable curve across a wide range of recall values, with the highest average precision achieved by RF at 0.964, followed by SVM at 0.957, XGBoost at 0.956, and MLP at 0.885. Conversely, the PR curve for random sampling tends to decline more rapidly as recall increases. This pattern indicates that random negative pairs result in models that are more prone to a decline in precision when the positive detection coverage is expanded. Thus, the advantage of GANATCpair is evident not only from the improvement in ROC-AUC but also from its ability to maintain the quality of positive predictions more consistently.

The distribution of true positives and classification errors in Table 4 reinforces these findings. In the GANATCpair scenario, RF produced a high number of true positives and true negatives, accompanied by low classification errors, namely only 3 false positives and 13 false negatives. XGBoost exhibits a similar pattern with 87 true positives, 95 true negatives, 7 false positives, and 15 false negatives. This distribution is consistent with the high precision, recall, and F1-score values reported in Table 3, whilst also explaining the stability of the PR curve in Figure 3.

Table 4: Confusion Matrix Results on the Snakebite Dataset

Sampling	**Model**	**True Positive (TP)**	**True Negative (TN)**	**False Positive (FP)**	**False Negative (FN)**
GANATCpair	RF	89	99	3	13
	MLP	77	84	18	25
	XGBoost	87	95	7	15
	SVM	66	75	27	36
Random	RF	60	69	33	42
Sampling 1:1	MLP	60	61	41	42
	XGBoost	67	61	41	35
	SVM	51	72	30	51
Random	RF	43	179	25	59
Sampling 1:2	MLP	41	153	51	61
	XGBoost	54	170	34	48

The random sampling scenario resulted in a higher error rate. In the RF with a 1:1 ratio, the number of false positives and false negatives increased to 33 and 42, whilst at a 1:2 ratio the model tended to produce higher true negatives but this was accompanied by an increase in false negatives. This pattern indicates a tendency for the model to predict the negative class more frequently when the negative pair distribution is formed randomly, thereby reducing sensitivity to positive interactions. These findings are consistent with the low recall in some

models in Table 3, particularly the SVM under 1:2 random sampling, and confirm that the improved performance of GANATCpair is reflected not only in aggregate metrics but also in a more controlled distribution of prediction errors.

The results in Table 3, Figure 2, Figure 3, and Table 4 indicate that the negative pairing strategy plays a crucial role in the evaluation of the DTI model on the Snakebite Envenoming Medicines Database. Under the evaluated cross-validation setting, GANATCpair was associated with higher performance than the two random sampling scenarios in terms of predictive performance, stability of positive predictions, and classification error distribution. These findings suggest that negative pairs generated via a generative approach have the potential to provide more relevant classification challenges, thereby enabling the model to learn interaction patterns more accurately. These results provide preliminary evidence that GANATCpair can support downstream DTI prediction for bioactive compounds and snake venom targets under the present experimental setting.

The ablation results indicate that the contribution of each GANATCpair component varies across classifiers, as summarized in Table 5. The Tanimoto Decoy variant generally improved AUC and F1-score compared with Unconstrained Decoy, suggesting that Tanimoto-based decoy space construction provides a more informative candidate space for putative negative selection. However, the Tanimoto + Cosine variant did not provide consistent performance gains when GAN-based latent mapping was not included, indicating that cosine-distance ranking alone was insufficient to identify robust negative samples. The Full GANATCpair achieved the highest performance across all classifiers, with AUC values of 0.943, 0.953, 0.936, and 0.917 for SVM, RF, XGB, and MLP, respectively. These findings show that the best performance was obtained when Tanimoto-based decoy construction, cosine-distance ranking, and GAN-based latent mapping were integrated.

Table 5: Ablation Study

Model	Metric	Unconstrained Decoy	Tanimoto Decoy	Tanimoto + Cosine	Full GANATCpair
SVM	AUC	0.661 ± 0.080	0.717 ± 0.064	0.663 ± 0.080	0.943 ± 0.008
	F1-score	0.604 ± 0.080	0.662 ± 0.040	0.612 ± 0.070	0.884 ± 0.014
RF	AUC	0.722 ± 0.090	0.768 ± 0.032	0.683 ± 0.074	0.953 ± 0.004
	F1-score	0.673 ± 0.083	0.707 ± 0.018	0.659 ± 0.079	0.912 ± 0.040
XGBoost	AUC	0.696 ± 0.092	0.703 ± 0.050	0.635 ± 0.066	0.936 ± 0.011
	F1-score	0.627 ± 0.054	0.673 ± 0.070	0.605 ± 0.086	0.892 ± 0.044
MLP	AUC	0.609 ± 0.065	0.651 ± 0.069	0.575 ± 0.066	0.917 ± 0.004
	F1-score	0.582 ± 0.046	0.599 ± 0.082	0.602 ± 0.061	0.888 ± 0.033

GANATCpair was further assessed beyond the snakebite context using the Yamanishi benchmark dataset, which comprises four target groups: Enzyme (E), G-protein-coupled receptor (GPCR), Ion Channel (IC), and Nuclear Receptor (NR). Figure 4 compares the AUC values obtained using Random Sampling 1:1, Random Sampling 1:2, and GANATCpair across four classification models. GANATCpair achieved the highest mean AUC across all evaluated model dataset combinations, although the magnitude of improvement varied according to the classifier and target group.

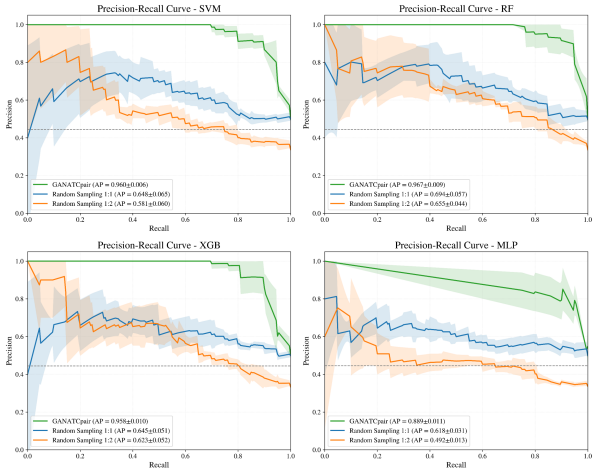

Figure 3. Precision-Recall curves of various models on the snakebite database

Tree-based models, RF and XGBoost, produced the most stable results across the entire dataset when combined with GANATCpair. RF achieved an AUC of 0.979 on Enzyme, 0.967 on GPCR, 0.979 on IC, and 0.978 on NR. XGBoost demonstrated comparable performance, with AUCs of 0.991, 0.973, 0.983, and 0.976 on the same sequence of datasets. This consistency may reflect the compatibility between tree-based learning algorithms and the heterogeneous tabular representation used in this study, where molecular fingerprints and protein sequence-derived features can be partitioned through hierarchical decision rules to capture non-linear drug-target patterns [14], [47]. These results indicate that GANATCpair is capable of maintaining the quality of informative negative pairs across various target distributions, including the NR class, which generally has a more limited dataset and tends to be challenging for random sampling scenarios.

Notably, the random sampling baselines also achieved competitive AUC values in several Yamanishi settings, particularly for tree-based models on the Enzyme and Ion Channel datasets. For RF on Enzyme, Random Sampling 1:1 and Random Sampling 1:2 achieved AUC values of 0.939 and 0.938, respectively, compared with 0.979 for GANATCpair. Similarly, XGBoost achieved AUC values of 0.933 and 0.943 under random sampling on Enzyme and 0.917 and 0.937 on Ion Channel, compared with 0.991 and 0.983 for GANATCpair. The performance of random sampling in these settings provides important context for interpreting the magnitude of improvement. The consistently higher AUC achieved by GANATCpair therefore indicates additional gains beyond these conventional baseline levels.

The advantage of GANATCpair is also evident in SVM and MLP. SVM achieved an AUC of 0.937 for Enzyme, 0.756 for GPCR, 0.881 for IC, and 0.682 for NR. Although its performance was lower than that of RF and XGBoost, all these values remained above those of the two random sampling scenarios. Meanwhile, the MLP showed a substantial improvement, particularly on GPCR and NR, with AUCs of 0.826 and 0.842, respectively. Although SVM and MLP can model non-linear relationships through kernel functions and multilayer architectures, respectively, their lower and less consistent AUC values may reflect greater vulnerability to spurious correlations in limited or heterogeneous DTI feature spaces involving putative negative samples [47]. These results indicate that MLP and SVM still benefit from targeted putative negative-pair construction, although their lower AUC values compared with RF and XGBoost suggest greater sensitivity to feature distribution, sample construction, and model configuration.

Table 6: Wilcoxon Signed-Rank Test Results for AUC Improvement

Dataset/Evaluation	**Random**	**GANATCpair**	**ΔAUC**	***p*-value**
Snakebite (Random 1:1)	0.5970	0.8799	0.2829	1.907×10^{-6}
Snakebite (Random 1:2)	0.6707	0.8799	0.2093	1.907×10^{-6}
Yamanishi (Random 1:1)	0.7909	0.8912	0.1003	8.152×10^{-14}
Yamanishi (Random 1:2)	0.7747	0.8912	0.1165	2.606×10^{-14}

Comparison across datasets shows that the largest performance gains occurred in the NR group, particularly for RF and XGBoost. For RF, the AUC increased from 0.857 with 1:1 random sampling and 0.796 with 1:2 random sampling to 0.978 with GANATCpair. XGBoost showed a similar pattern, rising from 0.835 and 0.785 to 0.976. These differences indicate that GANATCpair remains capable of constructing putative negative pairs that support model learning even on datasets with characteristics that are more difficult to distinguish. Conversely, the performance of SVM on NR remains relatively lower compared to other models, indicating that the effectiveness of the sampling strategy is still influenced by the model's capacity to utilise the resulting data structure.

The statistical significance of the observed performance differences was evaluated using the paired two-sided Wilcoxon signed-rank test, with GANATCpair compared against each random sampling baseline. As reported in Table 6, all comparisons yielded p-values below 0.05, indicating that the AUC improvements were statistically significant. Practical significance was assessed using the magnitude of ΔAUC, which was 0.2829 and 0.2093 in the Snakebite evaluation and 0.1003 and 0.1165 in the Yamanishi benchmark. These effect magnitudes indicate that GANATCpair provided practically meaningful improvements in discriminative performance, particularly in the Snakebite case study where the random sampling baselines produced substantially lower AUC values. Thus, the Wilcoxon test supports the consistency of the paired improvements, whereas the ΔAUC values provide evidence of their practical relevance.

The influence of λ on predictive performance was examined across the evaluated parameter range. Changes in AUC and F1-score were expressed relative to the default configuration $\lambda = 0.2$ for the Snakebite and Yamanishi NR datasets, as presented in Figure 5. Reducing λ to 0.1 or 0.15 generally resulted in lower AUC and F1-score across the evaluated datasets, while $\lambda = 0.25$ did not provide consistent gains over the default configuration. Increasing λ to 0.3 produced marginal improvements on the Snakebite dataset but was accompanied by larger performance reductions on Yamanishi NR. Across the evaluated parameter range, $\lambda = 0.2$ provided the most balanced cross-dataset performance, supporting its use as the default configuration in the main experiments.

Overall, these findings indicate that GANATCpair provides a more informative negative-pair construction strategy than conventional random sampling in both the Snakebite and Yamanishi evaluations. The consistent performance gains across multiple classifiers suggest that mapping GAN-derived latent representations to unknown-status pairs can strengthen the discriminative structure of the training data. The Snakebite evaluation provides a focused, application-specific preliminary assessment, while the Yamanishi benchmark supports the broader methodological utility of GANATCpair. The present experiments specifically evaluate GAN-guided pair selection against unguided random selection under matched downstream learning conditions. Accordingly, the results are limited to the evaluated random-sampling baselines and do not imply superiority over reliable-negative extraction [15] or positive unlabeled learning [23], which differ in their objectives, assumptions, and roles within the modelling pipeline.

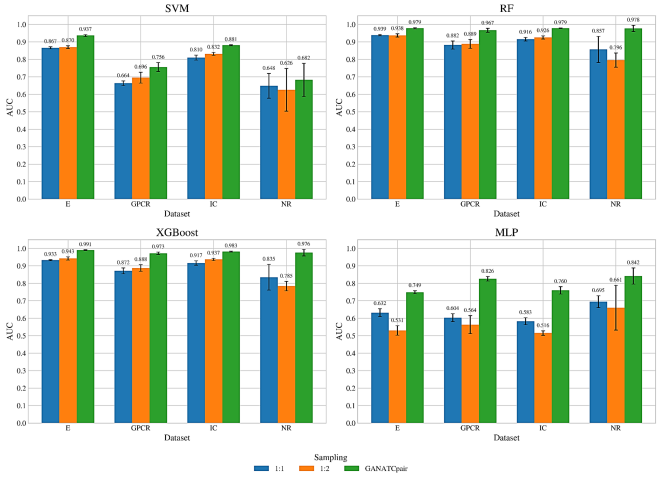

Figure 4. Comparative AUC value of sampling strategies on the Yamanishi benchmark dataset

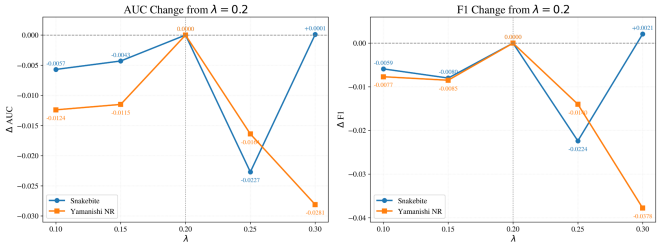

Figure 5. Sensitivity of AUC and F1-score to λ

The limited size of the Snakebite dataset reflects the scarcity of curated biological interaction records in snakebite-related DTI research. Given its compact scale, fold-level performance estimates may be sensitive to data partitioning, as reflected by the variability reported in Table 3. In the latent-space construction stage, this compact positive set was considered when interpreting the GAN-derived representations, since the generated anchors were used only to guide the selection of real unknown-status pairs rather than being directly added as classifier training samples. This variability was considered alongside evaluations involving four classifiers with different learning characteristics. The consistent direction of improvement across the four classifiers, its recurrence across the Yamanishi target groups in Figure 4, and the paired statistical results in Table 6 indicate that the observed performance pattern was reproduced across different modelling and dataset settings. Given the compact size of the Snakebite dataset, the magnitude of improvement is interpreted in the context of the applied fold composition and putative-negative class construction.

In hard-negative construction, proximity to known positive interaction patterns is used to obtain informative candidates near the classification boundary, consistent with hard-negative mining and contrastive representation learning [16], [48]. In GANATCpair, the selected pairs are not taken from P_{pos}, but are retrieved from the unknown-status pair space using GAN-derived latent anchors. Therefore, proximity between selected putative negatives and the positive latent region should not be interpreted as direct overlap with known positive interactions, but as an intended consequence of selecting challenging candidates in the latent space. Because unknown-status pairs may contain both true non-interactions and interactions that have not yet been experimentally reported, these candidates are treated as putative negatives rather than experimentally validated non-interactions [23]. Further assessment through molecular docking, molecular dynamics simulations, or experimental assays would help clarify their biological relevance.

Although the present study focuses on DTI prediction, the GAN-guided latent mapping formulation provides a pair-selection perspective that may be relevant to related pairwise interaction prediction tasks with observed positive links and unknown-status candidate links, such as protein protein interaction, peptide protein interaction, and drug drug interaction prediction. In such tasks, the representation of entity pairs, the construction of the candidate space, and the validation criteria would need to be defined according to the target domain. This suggests that broader applications of the proposed framework may be explored in future work.

4. CONCLUSION

This study shows that GANATCpair improved downstream DTI prediction performance under the evaluated random-sampling baseline settings by facilitating the selection of informative putative negative samples from unknown-status drug-target pairs. On the compact Snakebite dataset, GANATCpair yielded AUC values of 0.916-0.952, compared with 0.626-0.773 under random negative sampling across SVM, RF, XGBoost, and MLP.

GANATCpair uses GAN-derived latent vectors to guide the selection of existing unknown-status drug-target pairs without directly incorporating synthetic vectors into classifier training. This approach provides a targeted strategy for constructing putative negative samples and may support more discriminative computational screening and drug–target prioritisation for snakebite envenoming. Nevertheless, the selected pairs represent putative negatives rather than experimentally validated non-interactions. Future studies should therefore assess their biological relevance through molecular docking, molecular dynamics simulations, or experimental assays and evaluate GANATCpair using larger, more diverse DTI datasets and related pairwise tasks.

ACKNOWLEDGMENTS

This research was funded by the Indonesian Education Scholarship, the Centre for Higher Education Financing and Assessment under the Ministry of Higher Education, Science and Technology, and the Education Fund Management Agency under the Ministry of Finance of the Republic of Indonesia (ID 202327092129). The author would like to express gratitude to Prof. Dr.Eng. Wisnu Ananta Kusuma, S.T., M.T. for facilitating and supporting the use of the NVIDIA DGX H100 computing server. Appreciation is also to the Computer Science Study Program, IPB University, for supporting access to the NVIDIA GeForce RTX 3070 computing server used in conducting the experiments for this research.

REFERENCES

[1] World Health Organization, "Who expert committee on biological standardization, sixty-seventh report," World Health Organization, Geneva, Switzerland, Tech. Rep. 1004, 2017, https://iris.who.int/handle/10665/255657.

[2] R. Minghui, M. N. Malecela, E. Cooke, and B. Abela-Ridder, "Who's snakebite envenoming strategy for prevention and control," *The Lancet Global Health*, vol. 7, no. 7, pp. e837–e838, Jul. 2019, https://doi.org/10.1016/S2214-109X(19)30225-6.

[3] J. M. Gutiérrez *et al.*, "The search for natural and synthetic inhibitors that would complement antivenoms as therapeutics for snakebite envenoming," *Toxins*, vol. 13, no. 7, pp. 1–30, 2021, https://doi.org/10.3390/toxins13070451.

[4] J. Borri, J. M. Gutiérrez, C. Knudsen, A. G. Habib, M. Goldstein, and A. Tuttle, "Landscape of toxin-neutralizing therapeutics for snakebite envenoming (2015–2022): Setting the stage for an r&d agenda," *PLoS Neglected Tropical Diseases*, vol. 18, no. 3, p. e0012052, Mar. 2024, https://doi.org/10.1371/journal.pntd.0012052.

[5] J. M. Gutiérrez, N. R. Casewell, and A. H. Laustsen, "Progress and challenges in the field of snakebite envenoming therapeutics," *Annual Review of Pharmacology and Toxicology*, vol. 65, no. 1, pp. 465–485, Jan. 2025, https://doi.org/10.1146/annurev-pharmtox-022024-033544.

[6] P. G. Ojeda, D. Ramírez, J. Alzate-Morales, J. Caballero, Q. Kaas, and W. González, "Computational studies of snake venom toxins," *Toxins*, vol. 10, no. 1, pp. 1–24, 2018, https://doi.org/10.3390/toxins10010008.

[7] A. Vefghi, Z. Rahmati, and M. Akbari, "Drug-target interaction/affinity prediction: Deep learning models and advances review," *Computers in Biology and Medicine*, vol. 196, p. 110438, Sep. 2025, https://doi.org/10.1016/j.compbiomed.2025.110438.

[8] L. Xu, X. Ru, and R. Song, "Application of machine learning for drug–target interaction prediction," *Frontiers in Genetics*, vol. 12, p. 680117, 2021, https://doi.org/10.3389/fgene.2021.680117.

[9] W. Shi, H. Yang, L. Xie, X.-X. Yin, and Y. Zhang, "A review of machine learning-based methods for predicting drug–target interactions," *Health Information Science and Systems*, vol. 12, no. 1, p. 30, Apr. 2024, https://doi.org/10.1007/s13755-024-00287-6.

[10] W. A. Kusuma *et al.*, "In silico targeting of snake venom phospholipase a2 using phytochemicals from cyanthillium cinereum: An integrated machine learning and molecular simulation study," *Journal of Applied Pharmaceutical Science*, vol. 16, no. 03, pp. 375–391, 2026, https://doi.org/10.7324/JAPS.2026.271166.

[11] X. Ru, L. Xu, W. Han, and Q. Zou, "In silico methods for drug-target interaction prediction," *Cell Reports Methods*, vol. 5, no. 10, p. 101184, 2025, https://doi.org/10.1016/j.crmeth.2025.101184.

[12] C. Lin, S. Ni, Y. Liang, X. Zeng, and X. Liu, "Learning to predict drug target interaction from missing not at random labels," *IEEE Transactions on Nanobioscience*, vol. 18, no. 3, pp. 353–359, Jul. 2019, https://doi.org/10.1109/TNB.2019.2909293.

[13] J. Adhiva *et al.*, "Proventl: A transfer-learning framework for predicting peptide–protein interactions derived from snake venom for cancer therapeutics," *Journal of Computer-Aided Molecular Design*, vol. 40, no. 1, p. 90, Apr. 2026, https://doi.org/10.1007/s10822-026-00801-w.

[14] W. A. Kusuma *et al.*, "Prediction of the interaction between calloselasma rhodostoma venom-derived peptides and cancer-associated hub proteins: A computational study," *Heliyon*, vol. 9, no. 11, p. e21149, 2023, https://doi.org/10.1016/j.heliyon.2023.e21149.

[15] M. M. Sharifabad, R. Sheikhpour, and S. Gharaghani, "Drug-target interaction prediction using reliable negative samples and effective feature selection methods," *Journal of Pharmacological and Toxicological Methods*, vol. 116, p. 107191, Jul. 2022, https://doi.org/10.1016/j.vascn.2022.107191.

[16] J. D. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," in *9th International Conference on Learning Representations (ICLR)*, Austria, 2021.

[17] Z. Yang *et al.*, "Does negative sampling matter? a review with insights into its theory and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5692–5711, 2024, https://doi.org/10.1109/TPAMI.2024.3371473.

[18] M. A. Talukder, M. Kazi, and A. Alazab, "Predicting drug-target interactions using machine learning with improved data balancing and feature engineering," *Scientific Reports*, vol. 15, no. 1, p. 19495, Jun. 2025, https://doi.org/10.1038/s41598-025-03932-6.

[19] R. R. Kotkondawar, S. R. Sutar, A. W. Kiwelekar, V. J. Kadam, and S. M. Jadhav, "A generative framework for enhancing drug target interaction prediction in drug discovery," *Scientific Reports*, vol. 15, no. 1, p. 35588, Oct. 2025, https://doi.org/10.1038/s41598-025-01589-9.

[20] A. J. Bose, H. Ling, and Y. Cao, "Adversarial contrastive estimation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 1021–1032, https://doi.org/10.18653/v1/P18-1094.

[21] Q. Hu, X. Wang, W. Hu, and G.-J. Qi, "Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 1074–1083, https://doi.org/10.1109/CVPR46437.2021.00113.

[22] X. Wang, Y. Huang, D. Zeng, and G.-J. Qi, "Caco: Both positive and negative samples are directly learnable via cooperative-adversarial contrastive learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10 718–10 730, Sep. 2023, https://doi.org/10.1109/TPAMI.2023.3262608.

[23] J. Bekker and J. Davis, "Learning from positive and unlabeled data: A survey," *Machine Learning*, vol. 109, no. 4, pp. 719–760, 2020, https://doi.org/10.1007/s10994-020-05877-5.

[24] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanehisa, "Prediction of drug–target interaction networks from the integration of chemical and genomic spaces," *Bioinformatics*, vol. 24, no. 13, pp. i232–i240, Jul. 2008, https://doi.org/10.1093/bioinformatics/btn162.

[25] S. Kim *et al.*, "Pubchem 2025 update," *Nucleic Acids Research*, vol. 53, no. D1, pp. D1516–D1525, Jan. 2025, https://doi.org/10.1093/nar/gkae1059.

[26] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, 2014, pp. 2672–2680.

[27] H. Kuwahara and X. Gao, "Analysis of the effects of related fingerprints on molecular similarity using an eigenvalue entropy approach," *Journal of Cheminformatics*, vol. 13, no. 1, p. 27, Dec. 2021, https://doi.org/10.1186/s13321-021-00506-2.

[28] R. Kumar *et al.*, "An in silico platform for predicting, screening and designing of antihypertensive peptides," *Scientific Reports*, vol. 5, no. 1, p. 12512, 2015, https://doi.org/10.1038/srep12512.

[29] D. J. Rogers and T. T. Tanimoto, "A computer program for classifying plants," *Science*, vol. 132, no. 3434, pp. 1115–1118, Oct. 1960, https://doi.org/10.1126/science.132.3434.1115.

[30] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975, https://doi.org/10.1145/361219.361220.

[31] Z. Chen *et al.*, "ifeature: A python package and web server for features extraction and selection from protein and peptide sequences," *Bioinformatics*, vol. 34, no. 14, pp. 2499–2502, Jul. 2018, https://doi.org/10.1093/bioinformatics/bty140.

[32] K. Chou, "Prediction of protein cellular attributes using pseudo-amino acid composition," *Proteins: Structure, Function, and Bioinformatics*, vol. 43, no. 3, pp. 246–255, May 2001, https://doi.org/10.1002/prot.1035.

[33] S. Kim *et al.*, "Pubchem substance and compound databases," *Nucleic Acids Research*, vol. 44, no. D1, pp. D1202–D1213, Jan. 2016, https://doi.org/10.1093/nar/gkv951.

[34] A. Capecchi, D. Probst, and J.-L. Reymond, "One molecular fingerprint to rule them all: Drugs, biomolecules, and the metabolome," *Journal of Cheminformatics*, vol. 12, no. 1, p. 43, Dec. 2020, https://doi.org/10.1186/s13321-020-00445-4.

[35] J. P. Correia, L. R. da Silva, and R. Silva, "Multifractal analysis and support vector machine for the classification of coronaviruses and sars-cov-2 variants," *Scientific Reports*, vol. 15, no. 1, p. 15041, Apr. 2025, https://doi.org/10.1038/s41598-025-98366-5.

[36] X. Liu, Z. Tang, and J. Wei, "Multi-layer perceptron model integrating multi-head attention and gating mechanism for global navigation satellite system positioning error estimation," *Remote Sensing*, vol. 17, no. 2, 2025, https://doi.org/10.3390/rs17020301.

[37] R. Couronné, P. Probst, and A. Boulesteix, "Random forest versus logistic regression: A large-scale benchmark experiment," *BMC Bioinformatics*, vol. 19, no. 1, p. 270, Dec. 2018, https://doi.org/10.1186/s12859-018-2264-5.

[38] C. Huang, X. Zhu, M. Lu, Y. Zhang, and S. Yang, "Xgboost algorithm optimized by simulated annealing genetic algorithm for permeability prediction modeling of carbonate reservoirs," *Scientific Reports*, vol. 15, no. 1, p. 14882, Apr. 2025, https://doi.org/10.1038/s41598-025-99627-z.

[39] N. Mathai, Y. Chen, and J. Kirchmair, "Validation strategies for target prediction methods," *Briefings in Bioinformatics*, vol. 21, no. 3, pp. 791–802, May 2020, https://doi.org/10.1093/bib/bbz026.

[40] D. Chicco, "Ten quick tips for machine learning in computational biology," *BioData Mining*, vol. 10, no. 1, p. 35, Dec. 2017, https://doi.org/10.1186/s13040-017-0155-3.

[41] K. Sachdev and M. K. Gupta, "A comprehensive review of feature based methods for drug target interaction prediction," *Journal of Biomedical Informatics*, vol. 93, p. 103159, 2019, https://doi.org/10.1016/j.jbi.2019.103159.

[42] S. Szeghalmy and A. Fazekas, "A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning," *Sensors*, vol. 23, no. 4, p. 2333, Feb. 2023, https://doi.org/10.3390/s23042333.

[43] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics*, S. Kotz and N. L. Johnson, Eds. New York, NY: Springer, 1992, pp. 196–202, https://doi.org/10.1007/978-1-4612-4380-9_16.

[44] P. K. Singh, R. Sarkar, and M. Nasipuri, "Significance of non-parametric statistical tests for comparison of classifiers over multiple datasets," *International Journal of Computational Science and Mathematics*, vol. 7, no. 5, p. 410, 2016, https://doi.org/10.1504/IJCSM.2016.080073.

[45] O. Thas, J. C. W. Rayner, and D. J. Best, "Tests for symmetry based on the one-sample wilcoxon signed rank statistic," *Communications in Statistics – Simulation and Computation*, vol. 34, no. 4, pp. 957–973, Oct. 2005, https://doi.org/10.1080/03610910500308354.

[46] H. Xuan, A. Stylianou, X. Liu, and R. Pless, "Hard negative examples are hard, but useful," in *Computer Vision – ECCV 2020: 16th European Conference*. Glasgow, UK: Springer, 2020, pp. 126–142, https://doi.org/10.1007/978-3-030-58568-6_8.

[47] L. Erlina *et al.*, "Virtual screening of indonesian herbal compounds as covid-19 supportive therapy: Machine learning and pharmacophore modeling approaches," *BMC Complementary Medicine and Therapies*, vol. 22, no. 1, p. 207, Dec. 2022, https://doi.org/10.1186/s12906-022-03686-y.

[48] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823, https://doi.org/10.1109/CVPR.2015.7298682.