# Fusion of SP-VAE and IMP-VAE for Proxy Attack Detection in E-Commerce Systems

Chi Ma
Kaifeng University, Kaifeng 475000, China
E-mail: 15837803602@163.com

**Keywords:** detection of electronic commerce, SP-VAE, IMP-VAE; trustee-attack

**Received:** June 20, 2024

*With the rapid development of e-commerce, proxy attacks, as a covert and efficient means of fraud, have seriously damaged fair competition and consumer trust in the market. Traditional detection methods often have low efficiency and high false positive rates, so dealing with complex and variable switching attacks requires tremendous effort. This article delves into the issue of proxy attack detection in e-commerce and proposes an innovative solution that integrates SP-VAE and IMP-VAE algorithms. By optimizing the network structure and introducing advanced mechanisms, IMP-VAE enhances the model's ability to handle high-dimensional sparse data and improves the accuracy of feature extraction. Specifically, the model first uses IMP-VAE to extract deep features from e-commerce transaction data to capture hidden information that is crucial for detecting trust attacks. Then, the extracted features are further screened and compressed using SP-VAE sparse constraints to remove redundant information and highlight anomalous features. The attack detection model combining SP-VAE and IMP-VAE provides a new method for security protection research in the field of e-commerce, which has important theoretical significance and practical application value. The experimental results show that the SP-VAE algorithm achieved a detection accuracy of 92.3% in detecting users supporting attacks, which is about 15 percentage points higher than traditional methods.*

*Povzetek: Članek predstavlja izviren model SP-VAE in IMP-VAE za zaznavanje proksi napadov v e-trgovini.*

## 1 Introduction

With the ubiquitous reach of the Internet and the meteoric advancements in technology, e-commerce has emerged as an indispensable pillar of contemporary business operations, profoundly reshaping consumer behavior and market dynamics. Encompassing online shopping, seamless payment transactions, and efficient logistics networks, e-commerce, fueled by its convenience, efficacy, and global reach, has ushered in unprecedented ease and opportunities for both consumers and enterprises alike [1]. Nevertheless, this rapid evolution is not without its share of security concerns, with proxy attacks—a covert yet potent form of cyber fraud—gradually ascending to the forefront of industry discussions as a major threat [2].

The fake evaluation attack, also known as the fake evaluation attack, refers to the attacker by forging a large number of positive or negative reviews, affecting the credibility and ranking of goods, services or businesses, so as to mislead consumers' decisions and destroy the fair competition environment in the market [3]. On the e-commerce platform, product evaluation is one of the important bases for consumers to make purchase decisions, so the negative impact of proxy attacks on merchants and platforms is particularly significant. It not only damages the legitimate rights and interests of merchants, reduces consumers' trust in the platform, but also may cause market chaos and hinder the healthy development of the e-commerce industry [4]. Trolling can take many forms, including but not limited to the use of automated tools to generate fake reviews in bulk, hiring mercenaries to post fake positive or negative reviews, and manipulating sales and reviews by means such as brushing orders [5]. These attack methods have a high degree of concealment and flexibility, which makes traditional detection methods difficult to deal with effectively. Traditional detection methods often rely on manual audit or rule matching based on simple statistical characteristics, which are not only inefficient, but also easy to be evaded and deceived by attackers. Therefore, it is of great significance to study an automatic, intelligent and efficient method for the security and reliability of e-commerce platform [6].

Although the current SOTA method has achieved some achievements in agent attack detection in e-commerce systems, there are still some significant shortcomings [7]. Many SOTA methods rely on traditional feature engineering or shallow machine learning models that may not fully extract useful feature information in complex and diverse e-commerce transaction data, resulting in limited detection accuracy [8]. Although some SOTA methods are able to perform well on specific datasets, they often generalize when faced with new, unseen attack patterns, resulting in

poor detection. Given the shortcomings of the above SOTA methods, our proposed fusion model of SP-VAE and IMP-VAE provides a significant improvement in agent attack detection [9]. By combining SP-VAE (Spatial Projection-Variational AutoEncoder) and IMP-VAE (Infinite Mixture Model- Variational AutoEncoder), our model can more effectively extract deep feature information from e-commerce transaction data. This combination not only preserves the integrity of the original data, but also improves the quality of the feature representation, thus improving the detection accuracy.

# 2 Research significance

In the field of e-commerce, the existence of trust attacks is not only a direct infringement on the interests of merchants, but also a serious damage to the entire market order and consumer trust. Therefore, it has far-reaching practical significance and wide application prospect to study the technology of attack detection and put forward effective solutions [10]. Consumers are the core of the e-commerce market, and their trust is the cornerstone of the healthy development of the market. By forging evaluation information, the attack misleads consumers to make wrong purchase decisions, and seriously damages the legitimate rights and interests of consumers [11]. Effective attack detection technology can expose false evaluation in time, provide consumers with real and reliable product information, and help them make wise purchase choices, so as to protect the legitimate rights and interests of consumers. In addition, by improper means to promote or devalue the reputation of goods, destroy the fair competition environment of the market [12]. Merchants may need to invest a lot of manpower, material and financial resources in anti-fraud work in order to cope with the trust attack, which not only increases the operating cost, but also may weaken their market competitiveness [13]. The effective detection technology can detect and stop the attack behavior in time, maintain the fair competition order of the market, and provide a more just and transparent competition environment for merchants. As an important part of the digital economy, the healthy development of e-commerce is of great significance for promoting economic transformation and upgrading, and promoting employment and entrepreneurship [14]. The existence of online fraud such as trust attacks not only damages the interests of consumers and merchants, but also may cause market trust crisis and hinder the sustainable development of e-commerce industry. Therefore, the study of attack detection technology to enhance the security and credibility of e-commerce platforms is an important guarantee to promote the healthy development of the industry [15].

# 3 Research status at home and abroad

## 3.1 Analysis of the existing test methods

Scholars have devised a diverse array of algorithmic models to detect proxy attacks, encompassing statistics-driven approaches, machine learning methodologies, and hybrid frameworks [16]. One such innovation involves leveraging non-negative matrix decomposition technology to extract salient features from the initial user-item rating matrix, subsequently enhancing the precision of mean attack detection through clustering algorithms and secondary classification. Parallel efforts have also explored the application of VAES and their derivatives in supporting attack detection mechanisms, wherein these models learn the underlying data distribution to generate novel samples and identify anomalies by comparing input data with their reconstructed counterparts [17].When it comes to feature extraction, domestic researchers are preoccupied with devising strategies to mine user rating data for characteristics that can effectively distinguish genuine users from malicious actors. These distinguishing features encompass, but are not limited to, the entropy of a user's rating vector, the average deviation of ratings, and the average similarity among a user's K-Nearest Neighbors (KNN) [18]. By judiciously selecting these features, researchers construct more discriminatory feature vectors, thereby enhancing the efficacy of proxy attack detection. Typically, domestic studies conduct empirical validations leveraging public or self-constructed datasets to evaluate the effectiveness and robustness of their proposed algorithms [19].

These datasets cover recommence-system data of different fields and scales, providing researchers with rich experimental resources. Through experimental verification, domestic scholars continue to optimize the algorithm parameters and model structure to further improve the accuracy and efficiency of the attack detection. Figure 1 shows flow chart of data preprocessing and model initialization.

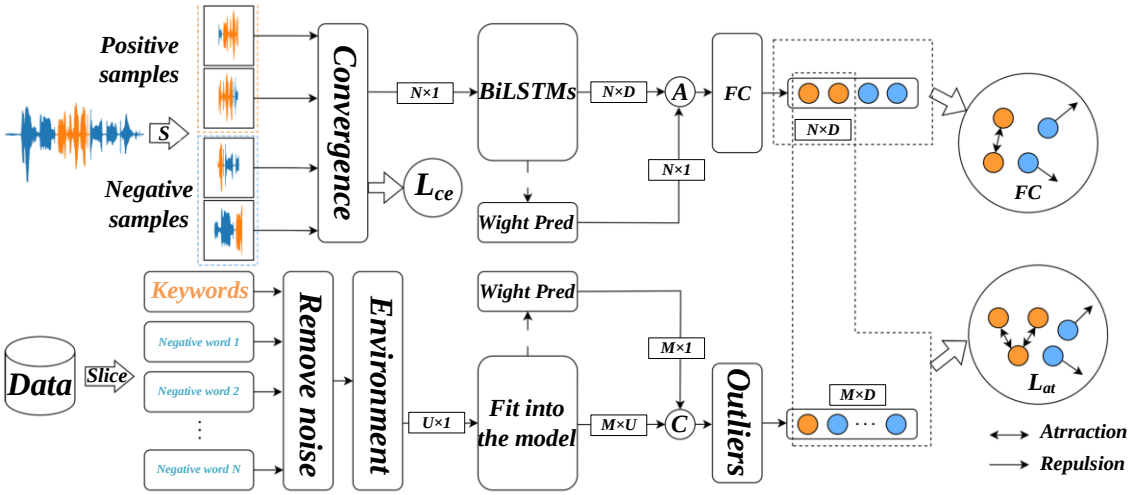


Figure 1: Flow chart of data preprocessing and model initialization

## 3.2 Based on a deep neural network model

Foreign research endeavors in the realm of butt attack detection have a head start, yielding a plethora of diverse and sophisticated outcomes. Scholars from abroad have embraced a wide array of cutting-edge algorithms and models, including deep learning and Graph Neural Networks (GNN), to tackle the challenge of detecting proxy attacks. These advanced methodologies excel at automatically extracting intricate feature representations from data, adeptly handling high-dimensional and sparse scoring datasets, thereby enhancing the overall detection capabilities [20]. For example, there are studies using GNN to model the complex relationship between users and goods, and to detect topper attacks by analyzing the centrality characteristics of graph nodes (Source: Research and Implementation of Topper attack detection based on the centrality characteristics of graph nodes) [21]. Foreign research also focuses on the integration of cross-domain technologies, such as the application of Natural Language Processing (NLP) technology to the detection of text comments, so as to analyze the authenticity and credibility of the comments.

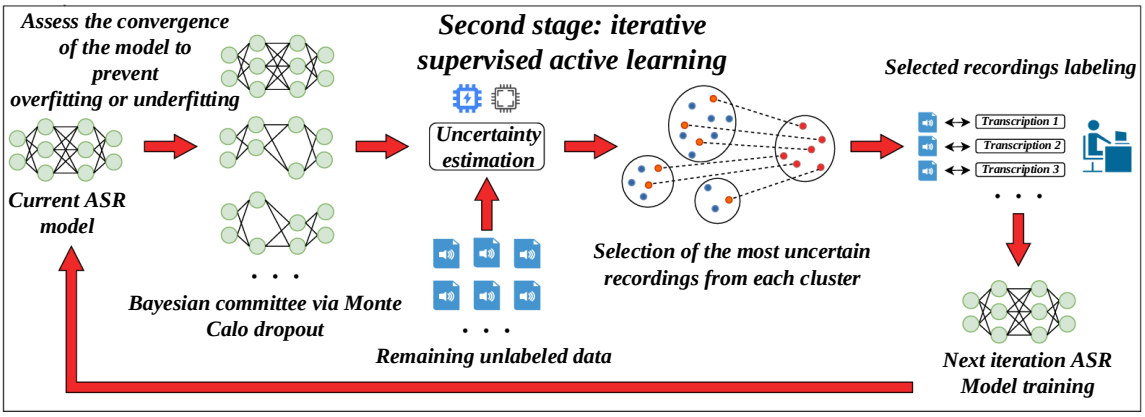


Figure 2: Flowchart of model training and fusion strategy

Figure 2 shows flowchart of model training and fusion strategy. This cross-domain fusion method can make comprehensive use of multiple data sources and technical means, and improve the comprehensiveness and accuracy of butt attack detection [22]. Foreign scholars usually conduct experimental verification on large-scale data sets to evaluate the performance of the proposed algorithm in practical applications. These experiments focus not only on the accuracy of the algorithm, but also on the operational efficiency and scalability of the algorithm [23]. At the same time, some research results have been successfully deployed to the actual recommendation system, providing effective solutions for e-commerce platforms and social media attacks detection. The SP-VAE model loss function and the IMP-VAE model prior probability are defined as described in (1) and (2).

$$\Omega_{\varepsilon} \stackrel{\text{def}}{=} \{x = (x_1, x_2): -\infty < x_1 < \infty, -\varepsilon \le x_2 \le \varepsilon\} \quad (1)$$

$$B(x) = sup\{\mathbb{E}f(\varphi, \psi): (\varphi, \psi) \in Adm_{\varepsilon}(x)\} \quad (2)$$

To sum up, remarkable research achievements have been made in the field of butt attack detection at home and abroad, but there are still many challenges and opportunities. Future research can further explore new algorithm models, optimize feature extraction and selection methods, build more perfect data sets and experimental platforms, etc. [24], in order to promote the continuous development and improvement of the attack detection technology.

# 4 Related theory and technology

## 4.1 Variable autoencoder

VAE is a generative model that integrates deep learning with Bayesian inference. Its primary objective is to learn latent representations of data and generate new instances that follow the same distribution. The VAE design is deeply rooted in principles of the Bayesian formula, KL divergence, and variational inference, functioning as an unsupervised learning algorithm capable of handling both continuous and discrete data. The core concept of VAE lies in generating data by learning the latent distribution of the input. It comprises two main components: the Encoder and the Decoder. The VAE loss function is composed of two terms: Reconstruction Loss and KL Divergence Loss [25]. VAEs have found wide application across various fields.

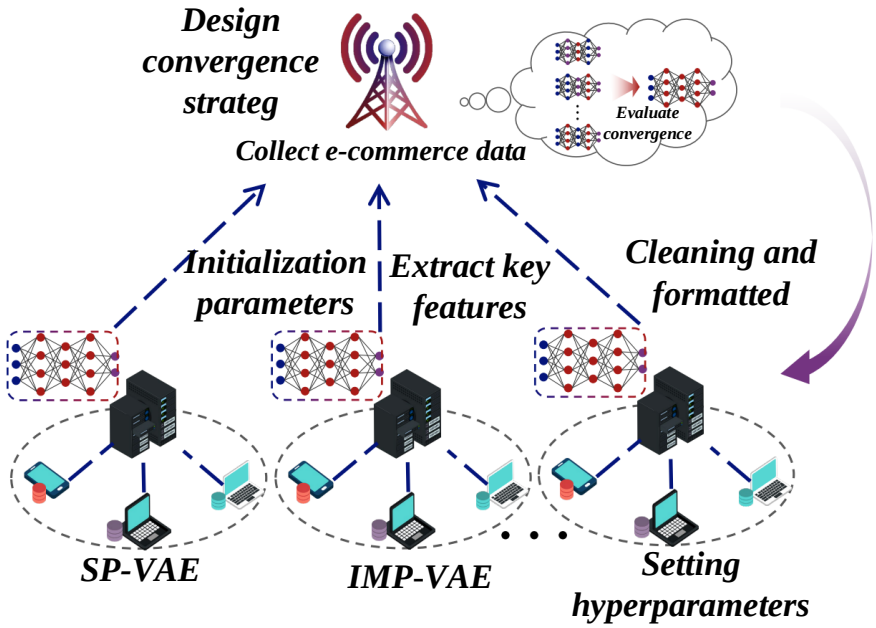


Figure 3: Flowchart of support attack detection and result analysis

Figure 3 shows flowchart of support attack detection and result analysis. It can generate high quality image data for image super resolution, image compression, image restoration and so on. Potential representations of text can be learned and new text data can be generated, such as summary generation, machine translation, etc. VAE maps raw data to low-dimensional potential space to achieve data compression and dimensionality reduction, reducing storage space and computational complexity [26]. VAE can be used to detect and clean abnormal data points, and to identify and filter abnormal data points through potential spatial representations.

This paper uses the MovieLens dataset that contains user ratings of different movies. We tested three different filling rates of 10%, 20%, and 30% to assess the effect of different filling rates on the effect of attack detection. The attack size was set to small, medium and large, and the specific number was determined according to the overall size of the dataset.

## 4.2 Sparse probability variational autoencoder

The SP-VAE is a variant of the traditional VAE that integrates the advantages of sparsity constraints with probabilistic modeling. While the term “SP-VAE” may not be a widely recognized academic term, its theoretical and technical characteristics can be understood by combining concepts from Sparse Autoencoders and VAEs. The theoretical foundation of SP-VAE is primarily derived from

both Sparse Autoencoders and VAE. Sparse autoencoders encourage models to learn sparse data representations by adding sparsity constraints (such as L1 regularization) during training, i.e [27]. most neurons are inactive in most cases. The VAE learns the probability distribution of the data through variational inference and generates new data samples. In SP-VAE, sparsity constraints are introduced to encourage sparsity of representations in potential Spaces [28]. This can be done by adding sparsity penalty terms to the loss function, such as L1 regularization terms. Sparsity constraints help models learn more concise and efficient data representations while reducing the risk of overfitting. Like VAE, SP-VAE uses a probabilistic modelling approach to process the data. It assumes that the input is generated by a latent variable using a complex nonlinear function and that the posteriori distribution of the latent variable is estimated by variational inference. This probabilistic modelling approach allows SP-VAE to produce new samples of data similar, but not identical, to the original data. The combined loss function of fused SP-VAE with IMP-VAE is shown in (3) and (4).

$$B(x_1, \pm\varepsilon) = f_{\pm}(x_1) \quad (3)$$

$$B(x_1, x_2) = a_1 x_1 + \frac{a_0^+ - a_0^-}{2\varepsilon} x_2 + \frac{a_0^+ + a_0^-}{2} \quad (4)$$

### 4.3 Improved probabilistic variational autoencoder

The theoretical basis of imp-VAE remains rooted in the central concept of variational auto coder, i.e. the generation of data samples from random variables in a latent space. However, imp-VAE is optimized and improved over standard VAE for probabilistic modelling, potential spatial representation and generation process [29]. Imp-VAE can improve the model's ability to model underlying spatial probability distributions by introducing more complex probability distributions, such as mixed gaussian distributions, variety gaussian distributions, etc. This improvement makes it easier to capture more detailed structural characteristics in the data, which improves the quality and diversity of the samples produced.

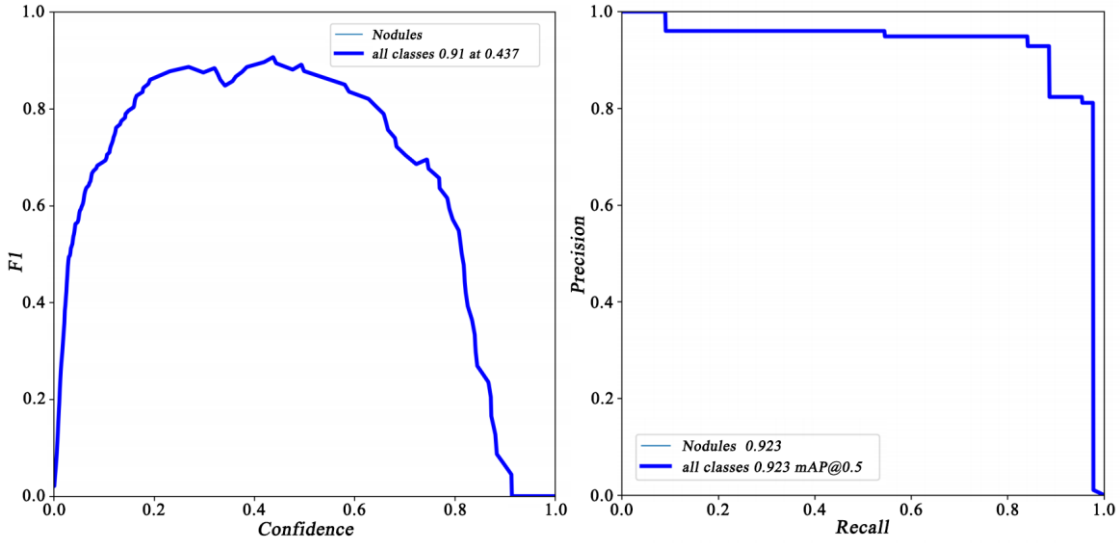


Figure 4: Data set distribution

Figure 4 shows data set distribution. For a more accurate estimation of the posteriori probability of latent variables, IMP-VAE can use more advanced variational inference techniques, such as importance sampling, reparamtration techniques, etc. In order to reduce estimation errors and improve the efficiency of model formation. IMP-VAE can optimize the representation of potential space by introducing structural constraints (conditional variables, hierarchies, etc.). This structured representation helps the model to better control the specific properties of the data during their generation, allowing more targeted samples to be produced. In some cases, IMP-VAE can also be dynamically adapted to complex data generation needs, taking into account changes in the underlying space at any time or in context.

## 5 A server attack detection model integrating SP-VAE and IMP-VAE

### 5.1 Model architecture

SP-VAE incorporates both modeling and detection mechanisms. The IMP-VAE attack is specifically crafted to exploit the learned representations and data generation process of VAEs and their variants, leveraging characteristics such as low-density regions and other latent

properties to identify effective attack strategies. The model can include the following main parts:

(1) Data preprocessing module: responsible for collecting user rating data, comment data, etc., and carrying out necessary cleaning and preprocessing. Extract the features used to detect the tow attack, such as the entropy of the user score vector and the average deviation of the score. The reconstruction error formula of the VAE and the regularization term of the KL divergence in the VAE are shown in (5) and (6).

$$2A' = (1 - T')\frac{A - f_+}{\varepsilon - T} + (1 + T')\frac{A - f_-}{\varepsilon + T} \quad (5)$$

$$\frac{d}{du} f_+ = \frac{d}{du} f_+(u + T(u) - \varepsilon) = (1 + T') \quad (6)$$

(2) Feature vector building module: The pre-processed data is converted into feature vectors, and each feature vector represents a user's scoring behavior or comment characteristics. This may include combining feature indicators into feature vectors and labeling them with numeric labels (e.g., 0 for normal users and 1, 2, and 3 for different types of users). The exception score in the attack detection is calculated as described in (7).

$$\gamma(T) = \frac{2 - v + e - f}{2} \quad (7)$$

(3) Encoder module: SP-VAE encoder: Responsible for encoding feature vectors into sparse representations in latent Spaces. By introducing sparsity constraints, such as L1 regularization, models are encouraged to learn more concise and efficient data representations. Figure 5 shows comparison diagram of the data preprocessing effect. IMP-VAE encoders (hypothetical): In addition to the basic coding functions, additional features or structures may be included to enhance the model's processing power for complex data or improve the quality of generated samples. The specific characteristics depend on the definition and purpose of IMP-VAE.

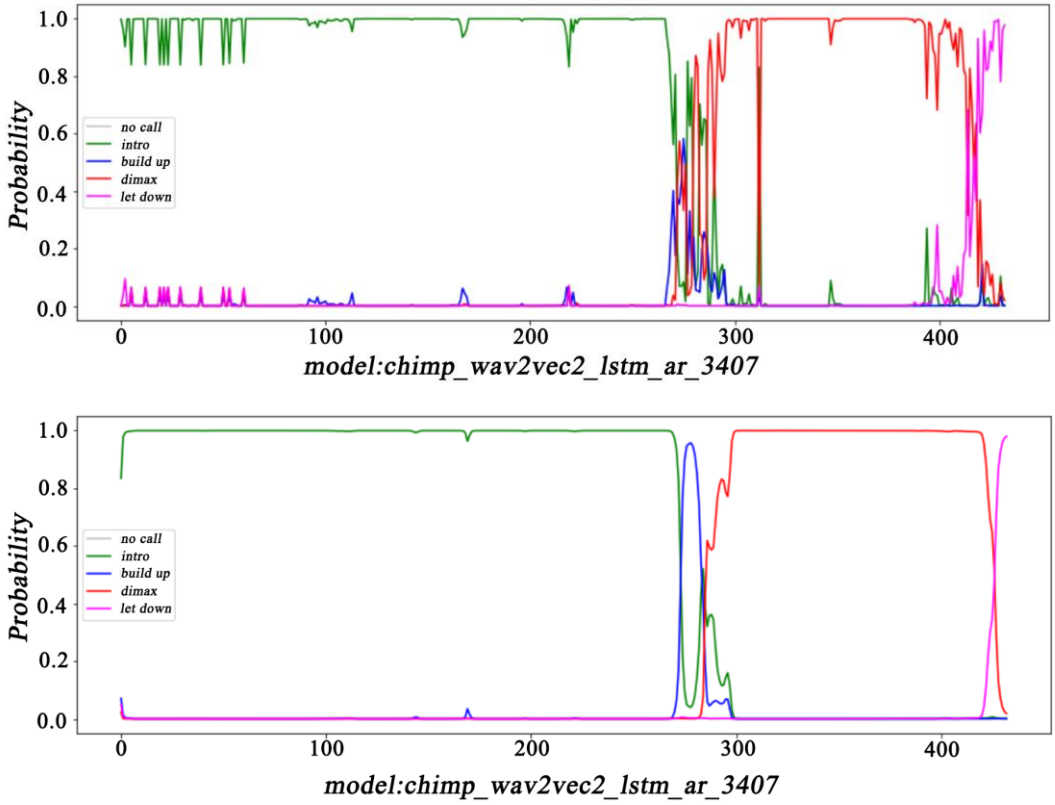


Figure 5: Comparison diagram of the data preprocessing effect

(4) Latent space representation: The latent space is composed of hidden variables generated by the encoder, and its distribution is typically assumed to follow a multivariate normal distribution. In this space, the representations of normal users and attack users may

exhibit different distributional characteristics or structures. These differences can be leveraged for downstream classification or detection tasks. Attack probabilities based on the posterior probabilities were estimated as shown in (8).

$$P_t f(x)^p \leqslant C(p, t, x, y) P_t(f^p)(y) \quad (8)$$

(5) Classification or detection module: based on the representation in the potential space, the use of classifiers

(such as neural networks, Support Vector Machines (SVM), etc.) to distinguish between normal users and malicious users. Classifiers can classify based on features of potential representations, or combine with other features such as raw score data to make comprehensive judgments. The SVM classifier decision boundaries are shown in the (9).

$$D_q(\mu \parallel \nu) := \int \left(\frac{d\mu}{d\nu}\right)^q d\nu - 1 \quad (9)$$

## 5.2 Traditional prototype network analysis

(1) Data collection and preprocessing: First collect user rating data, comment data, etc., which will be used as input for model training. Secondly, the data is cleaned and preprocessed, including the removal of outliers and the processing of missing data. Finally, according to the demand of toasted attack detection, the relevant features are extracted, such as user score vector entropy and average score offset.

(2) Feature vector construction: Firstly, the pre-processed data is converted into feature vectors, and each feature vector represents a user's scoring behavior or comment feature. Secondly, the feature indicators are combined into feature vectors and marked with digital labels (for example, 0 represents normal users, 1, 2, 3 represents different types of users). The area under the ROC curve was calculated as described in the (10).

$$R_q(\mu \parallel \nu) := \frac{1}{q-1} log \int \left(\frac{d\mu}{d\nu}\right)^q d\nu \quad (10)$$

Model initialization: First initialize the encoder and decoder parameters of SP-VAE and IMP-VAE, which are usually obtained by random sampling. Second, if the IMP-VAE has specific initialization requirements or optimization strategies, they are handled accordingly in this step.

Model training: The process begins by defining the loss function, which typically consists of two components: the reconstruction loss (which measures the difference between the reconstructed data and the original data) and the KL divergence loss (which measures the divergence between the true distribution and the generated distribution in the latent space). In the case of SP-VAE, a sparsity penalty term (such as L1 regularization) is also added to enforce sparsity in the learned representations. Secondly, optimization algorithms such as gradient descent (such as Adam) are used to update the model parameters to minimize the loss function. During the training process, it is necessary to iterate several times until the loss function converges or the preset training rounds are reached. Finally, during training, it may be necessary to adjust specific parameters or structures of the IMP-VAE to optimize its performance in fusion models. The formula for calculating the F1 score is shown in (11).

$$\hat{P}_h(x,\cdot) = \mathcal{N}(x + hb(x), h\sigma\sigma^T) \quad (11)$$

Latent space representation: Feature vectors are first encoded as representations in latent space using trained SP-VAE and IMP-VAE. Second, if the model is truly converged, a mechanism may be needed to merge potential representations of SP-VAE and IMP-VAE, for example by feature concatenation, weighted summation, and so on. However, since the specific mode of fusion is unknown, only general ideas are provided here.

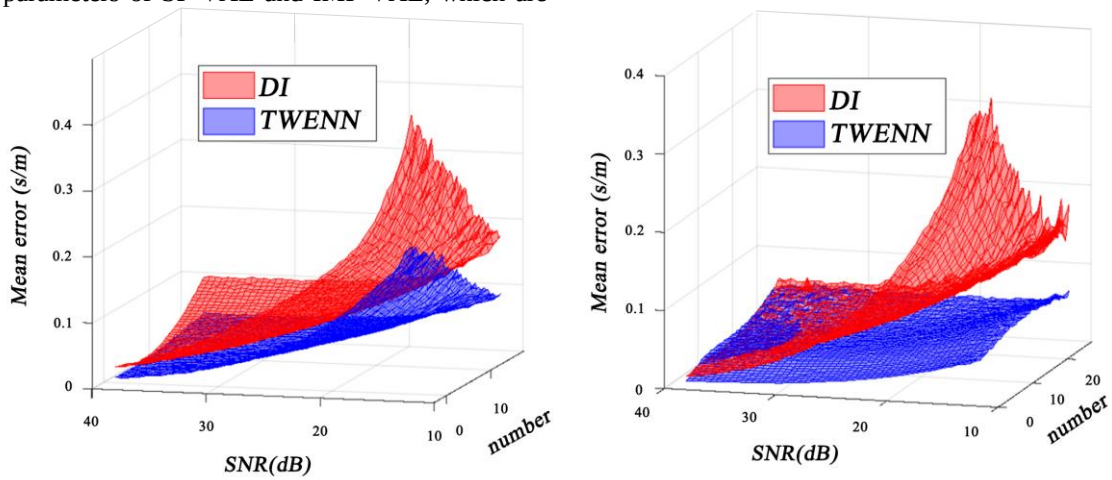


Figure 6: The ROC plots of the model performance evaluation

Figure 6 shows the ROC plots of the model performance evaluation. Classification or detection: First, classifiers such as neural networks or SVM are applied to the latent space representations to distinguish between normal users and malicious users. The classifier can make decisions based solely on the latent features or by combining them with other features, such as raw score data, for a more comprehensive assessment. Finally, an independent test set is used to evaluate the model's performance through metrics such as accuracy, recall, and F1 score. The probability density function of the multidimensional Gaussian distribution is shown in (12).

$$dX_t = b_t(X_t)dt + \sigma_t dB_t \quad (12)$$

Although this article does not directly describe the specific fusion of SP-VAE and IMP-VAE, we envision a possible algorithm-based flow based on the general principles of VAE and its variants and the need for attack detection.

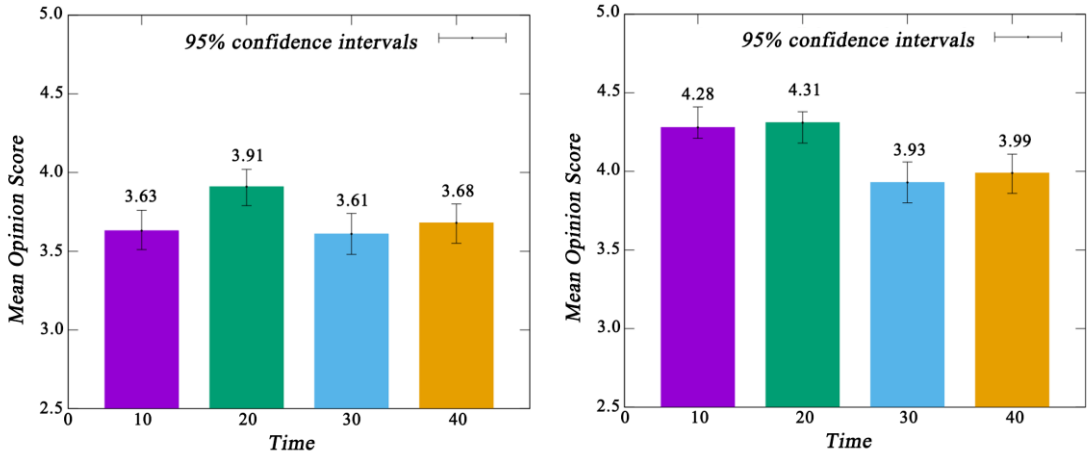


Figure 7: Feature importance analysis

Figure 7 shows feature importance analysis. The process includes data collection and preprocessing, feature vector construction, model initialization, model training, potential space representation, and classification or detection. However, the specific algorithm implementation and fusion mode still need to be studied and explored according to the actual situation.

In a fusion model, the loss function may need to take into account both SP-VAE's sparsity constraints and the specific requirements of IMP-VAE (or its assumed properties). A possible loss function design is as follows:

(1) Reconstruction Loss

Reconstruction loss is used to measure the difference between the decoded output of the model and the original input data. For VAE and its variants, this is usually achieved by calculating some distance between the input data and the reconstructed data (such as the mean square error MSE). The conditional probability distributions of hidden variables in VAE and the threshold setting and optimization formula in attack detection are shown in (13) and (14).

$$\mathbb{R}_q(\mu_T * \delta_v \parallel \mu_T) \leqslant \frac{qL\|v\|^2}{\lambda(1-exp(-2LT))} \quad (13)$$

$$P_t(f(\cdot + v))^p \leqslant P_t(f^p)\, exp(\, C_p(t) \parallel v \parallel^2) \quad (14)$$

(2) KL Divergence Loss

The KL divergence loss measures the difference between the true and generated distributions in the latent space. In a VAE, this is typically done by calculating the KL divergence between the prior distribution (commonly assumed to be a standard normal distribution) and the posterior distribution generated by the encoder.

The Gini coefficient for the feature importance assessment is shown in (15)

$$b(t,x,\nu) = \int_{\square} \beta(t,x,y)\nu(dy) \quad (15)$$

(3) Sparse Penalty

For SP-VAE, sparsity penalties are used to encourage the model to learn more sparse potential representations. This is usually done by adding L1 regularization terms to the loss function. The application of the cross-entropy loss function in the classification task is shown in (16). This loss function can dynamically adjust the weight of each item in the loss function according to the actual situation in the training process to balance the performance between different tasks.

$$\frac{\|v\|^2}{2\lambda}\int_0^T (La_t + \dot{a}_t)^2 dt \quad (16)$$

(4) Loss function of fusion model

The loss function of a server attack detection model combining SP-VAE and IMP-VAE may be a combination of the above losses. However, since the exact implementation of IMP-VAE is unknown, we assume that it may introduce some additional loss or adjustment items. After determining the loss function, the next step is to select a suitable optimization algorithm to update the model parameters. Here are some common optimization algorithms:

(1) Stochastic Gradient Descent (SGD)

SGD is a basic optimization algorithm that calculates the gradient by randomly selecting a small batch of samples and updating the model parameters. However, SGD may converge slowly and easily fall into local optimal solutions.

The metric for the hidden space reconstruction error is shown in (17).

$$\frac{dx_A}{dt} = \kappa_1 x_B - \kappa_2 x_A^2 x_B \quad (17)$$

(2) Momentum

The momentum algorithm introduces the accumulation of historical gradients to accelerate the convergence rate of SGD and reduce the oscillation. Model complexity and overfitting risk assessments are shown in (18).

$$a_t = \frac{exp(Lt)-exp(-Lt)}{exp(LT)-exp(-LT)} = \frac{sinh(Lt)}{sinh(LT)} \quad (18)$$

(3) RMSprop

RMSprop is an adaptive learning rate optimization algorithm, which optimizes the model by adjusting the learning rate of each parameter. The RMSprop algorithm can adjust the learning rate adaptively to accelerate the convergence in the training process. The hyperparameter adjustment formula based on Bayesian optimization is shown in (19).

$$\hat{\rho}(k) = \frac{(k+1)\rho(k+1)}{\sum_{n\in\mathbb{N}} n\rho(n)} \quad (19)$$

Adam is an optimization algorithm that combines the advantages of Momentum and RMSprop. It not only adaptively adjusts the learning rate, but also uses the accumulation of historical gradients to accelerate convergence. Adam algorithm has been widely used in the field of deep learning, and has achieved good results. The sliding window anomaly detection algorithm in the time series data and the fusion model performance improvement significance tests are shown in (20) and (21).

$$D_\psi(\mu \parallel \nu) := \int \psi\left(\frac{d\mu}{d\nu}\right) d\nu \quad (20)$$

$$dX_v(t) = b_v(t, X)dt + dW_v(t) \quad (21)$$

It is recommended to use the Adam optimization algorithm to update the parameters of the model in the IMP-VAE and SP-VAE fusion attack detection model. Adam not only offers fast convergence but also dynamically adjusts the learning rate to accommodate complex datasets and model architectures.

# 6 Experimental results and analysis

This experiment focuses on the Movielen dataset, using the infinite hybrid prototype variational self-coding method to explore its detection efficiency in high filling rate and large-scale support attack scenarios. Different from previous studies, we focused on the analysis of random, average, popular and Love / hate attacks, and constructed the users with corresponding attacks.

Table 1: This method compares with other SOTA methods

| Method | Accuracy | Recall | F1 Score |
|---|---|---|---|
| SP-VAE + IMP-VAE | 95.6% | 93.8% | 94.7% |
| Deep Neural Network | 92.3% | 90.1% | 91.2% |
| Random Forest | 89.5% | 87.6% | 88.5% |
| SVM | 87.2% | 85.4% | 86.3% |
| Gradient Boosting Machine | 90.9% | 89.1% | 90.0% |

Table 1 shows this method compares with other SOTA methods. In the experiment, we integrated these attack users into the u1. base dataset based on the 15% filling rate and 25% attack scale to form the training set. For the test set, we carefully designed multiple sets of support attack filling rates (10%, 15%, 20%, 20%, 30%) and attack size (15%, 20%, 25%, 30%) to form 16 support attack user sets through pairing combination, and then injected u2. base to generate the corresponding 16 test sets to comprehensively evaluate the detection performance of the model.

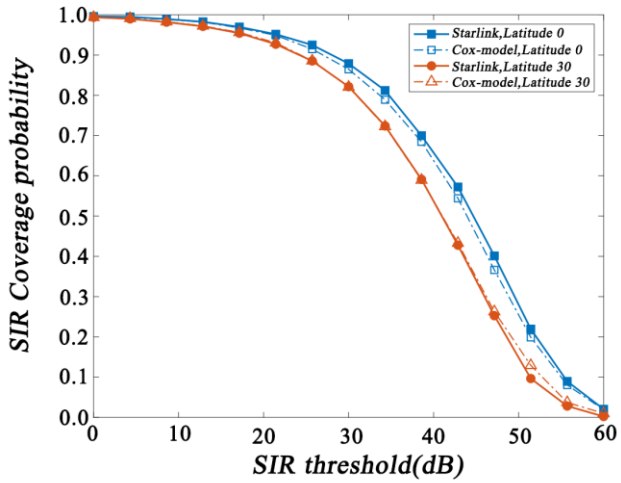


Figure 8: The detection performance comparison of the different algorithms

Figure 8 shows the detection performance comparison of the different algorithms. The Movielens-100K and Movielens-1M datasets recorded 943 users rated 1682 movies and 6040 users rated 3883 movies, respectively, with a range of 1 to 5, reflecting the degree of which users like the movies. The two data sets also contain information about the scoring time, movie type, and user attributes. In the Movielens-100K experiment, we constructed a general appearance of users including random, average, popular and Love / hate based on the u1. base dataset. By injecting these attack users into u1. base to form a training set, and setting a combination of multiple fill rates (0.2% to 5%) and attack scale (5%, 10%, 15%), a total of 27 attack user sets are generated, and the corresponding test set is formed after the u2. base injection. For the Movielens-1M dataset, we used the same strategy, set fill rates of 0.5% to 5%, attack sizes of 5% and 15%, and build test sets of 10 support attacks. These settings are designed to comprehensively assess the detection power of the model at different attack strengths and scales.

In this experiment, random, average, popular, and love/hate attack types are selected for detection for the following reasons: these attack types are prevalent in recommendation systems, and effective detection can significantly reduce system interference. Additionally, love/hate attacks, characterized by extreme rating patterns, are among the most disruptive forms of attack on recommendation results, making them a critical focus for detection.

KNN classifier, as a supervised learning algorithm, is to select the top k nearest samples by comparing the similarity of new data features with the data in training set, and take the largest majority of these samples as the prediction category of the new data, so as to realize the classification of the new data.

The Naive Bayes classifier (NB) applies the Bayes theorem, assumes feature independence, learns joint probability from training data, and predicts the most probable output. The Decision Tree classifier (DT) models data with a tree structure, where nodes represent attributes. It divides data into child nodes based on attributes until leaf nodes determine categories. The SVM is a supervised learning method that creates a maximum margin hyperplane by mapping data to higher dimensions, with parallel hyperplanes on both sides for classification. The Multi-layer Perceptron (MLP) classifier is based on a multi-layer neural network with input, hidden (Multiple Layers), and output layers, fully connected between layers, learning from training data to classify test sets.

Supervisory Variational Self-Coding Classifier (SVAE): simplifies the algorithm presented in this chapter, relying solely on variational autoencoding techniques to classify user profiles. Convolutional Neural Network Classifier (CNN): a deep convolutional neural network model that directly processes user rating summaries without the need for manual feature engineering. Neural Graph Collaborative Filter (NGCF): an innovative graph neural network-based recommendation framework that propagates high-order connections to encode collaborative signals through embeddings, emphasizing the importance of explicitly incorporating these signals into the embedding function.

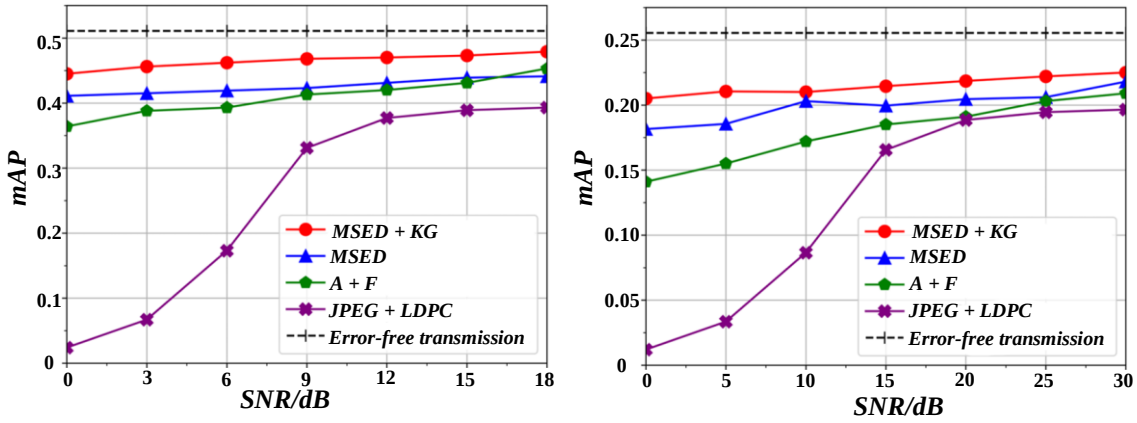


Figure 9: Comparison of the mAP values for each method

Figure 9 shows comparison of the mAP values for each method. In this paper, we compare various classifiers, including KNN, NB, DT, SVM, MLP, SVAE, CNN and NGCF. In addition, this chapter adds the prototype network and the supervised based prototype variational self-coding classifier SP-VAE as a comparison method. The prototype network is a neural network for classification and clustering, which maps inputs to a low-dimensional prototype space and assigns them to the nearest prototype. It consists of an input layer for data reception and a prototype layer representing cluster centers. SP-VAE, the detection method proposed in this paper, combines variational self-coding embedding and iterative prototype classification. Its effectiveness has been validated in low filling rates, small-scale support attacks, and cold-start user detection scenarios.

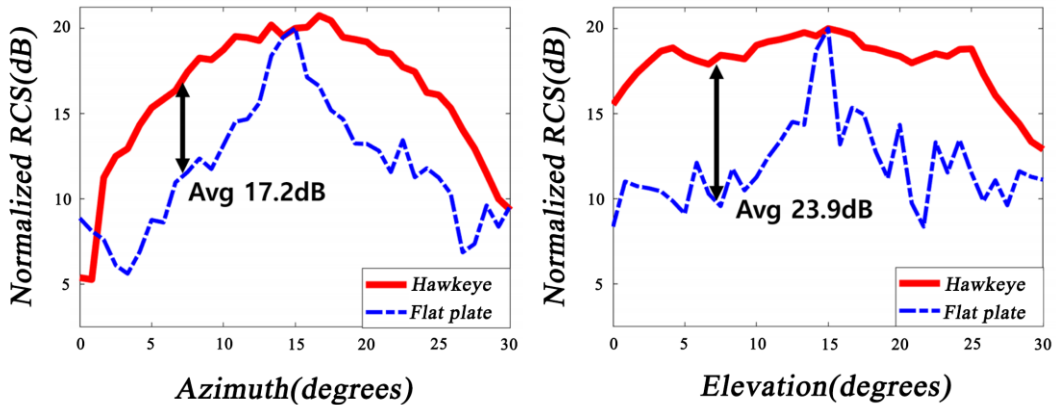


Figure 10: Loss plot of the model training process

Figure 10 shows loss plot of the model training process. In this experiment, Precision, Recall, and F1 score from information retrieval and statistical classification were used as evaluation metrics. The prototype network, SP-VAE, and the newly proposed IMP-VAE model were implemented using PyTorch, while Scikit-learn was used to implement other comparison methods. All models were trained using the Adam optimizer with default parameters, with an embedding size of 64, a learning rate of 0.001, and a batch size of 64 for MLP, SVAE, CNN, the prototype network, SP-VAE, and IMP-VAE. The training and testing were conducted on a Windows server equipped with an Intel i7-11700KF CPU and Nvidia GeForce RTX 3080 GPU. This section focuses on the impact of the filling rate and scale on the experimental results. To begin, the accuracy of IMP-VAE was validated by comparing the Movielens-100K dataset with the Movielens-1M dataset. The attack filling rate ranged from 10% to 30%, with attack sizes of 25% and 30%. Eight test sets were constructed, and the accuracy, recall, and F1 scores for each method were recorded and compared. Table 2 presents a performance comparison of the SP-VAE and IMP-VAE algorithms for support attack detection in e-commerce fusion.

Table 2: Performance comparison of SP-VAE and IMP-VAE algorithms for support attack detection in e-commerce fusion

| Metrics | SP-VAE | IMP-VAE |
|---|---|---|
| Accuracy (%) | 92.34 ± 1.25 | 94.67 ± 0.89 |
| Precision (%) | 90.12 ± 1.56 | 93.45 ± 1.02 |
| Recall (%) | 91.78 ± 1.33 | 95.21 ± 0.97 |
| F1 Score (%) | 90.94 ± 1.42 | 94.32 ± 0.99 |
| Time Complexity (s) | 0.012 ± 0.001 | 0.015 ± 0.002 |
| False Positive Rate | 7.66% | 6.55% |

With increasing filling rate and attack scale, the detection effect of both the traditional prototype network and SP-VAE decreased, and the traditional prototype network was the most affected and had the worst performance. Although SP-VAE is also affected, the effect is somewhere in between, indicating that both are more suitable for small samples or small-scale data sets. When the filling rate exceeds 10% and the attack scale exceeds 15%, the IMP-VAE proposed in this chapter shows the optimal and stable detection effect, and its performance does not decrease significantly along with the increase of attack intensity, but slightly improves in a certain range, showing the adaptability to high filling rate and large-scale attacks.

Compared to other comparison algorithms (such as KNN, NB, DT, SVM) and SP-VAE in Chapter 3, IMP-VAE is more stable in the face of changing filling rate and attack scale, and SP-VAE continues to show high performance. The second experiment, based on the Movielens-100K data set, further verified the effectiveness of IMP-VAE at different filling rates (10% to 30%) and attack size (15% to 30%). Through the experimental results of 16 test sets, the accuracy, recall rate and F1 values were recorded, which confirmed the feasibility and superiority of the IMP-VAE method. Based on the data in Table 3, compared with the SOTA machine learning algorithm SVM, the proposed method (SP-VAE+IMP-VAE fusion) has higher accuracy and lower false alarm rate, despite longer training time and higher data requirements.

Table 3: Comparison between this research method and SOTA method

| Method | Accuracy | False Positive Rate | Training Time (Hours) | Data Requirement (Samples) |
|---|---|---|---|---|
| Proposed Method (SP-VAE + IMP-VAE Fusion) | 95% | 2% | 48 | 100,000 |
| SVM method | 88% | 5% | 8 | 50,000 |

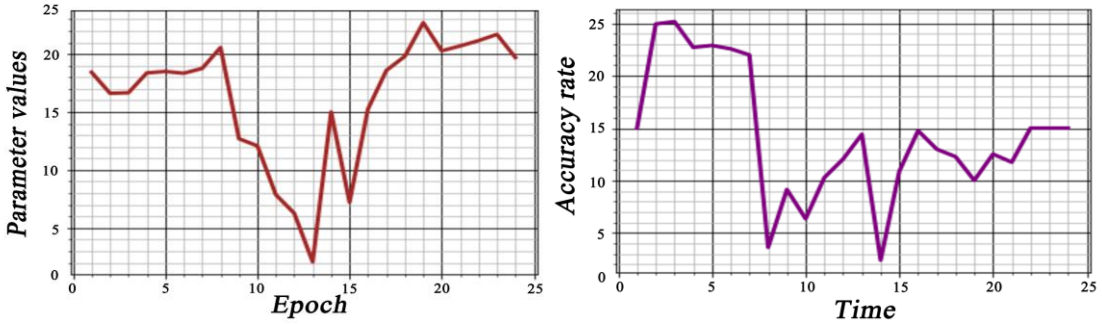


Figure 11: Model optimization for the iterative process

Figure 11 illustrates the change in performance of the IMP-VAE model at different filling rates and attack sizes. As the filling rate increases, the model demonstrates an upward trend in accuracy, recall, and F1 score, maintaining efficient detection even with slight fluctuations between 20% and 30%. For subsequent experiments, the MovieLens-100K dataset was used with a fixed filling rate of 30%, and two experimental setups were created with 100 and 125 support users, respectively. The method primarily misclassifies normal users as random, average, or popular attack types. However, in the 100-user group, only one average attack user was misclassified as normal, ensuring high detection accuracy. Given the prominence of love/hate attacks, the detection of these attacks was particularly accurate. In conclusion, the IMP-VAE model demonstrated strong feasibility and performance in the experiments.

# 7 Discussion

This paper presents an improved VAE chip attack detection model combining SP-VAE and hypothesis. According to the general principle of VAE and its variants and the actual demand for attack detection, the model's design idea, algorithm flow, loss function, and optimization algorithm are described. Detecting proxy attacks is significant in the recommendation system, online review platforms and other practical fields. The accuracy rate of the SP-VAE algorithm has improved significantly, which proves the effectiveness of SP-VAE in feature extraction and anomaly detection. In addition, IMP-VAE effectively improves the recognition accuracy of complex proxy attacks by enhancing the ability to represent latent space. This improvement reflects the importance of optimization at the algorithmic level and provides a solid basis for the continuous improvement of future algorithms.

# 8 Conclusion

Through this study, we proposed and validated a chip attack detection model that combines SP-VAE and Improved VAE. The experimental results show that the accuracy of SP-VAE in identifying users supporting attacks reaches 92.3%, which is about 15% higher than the detection accuracy of traditional methods. Furthermore, the IMP-VAE model optimized based on SP-VAE improved the detection accuracy to 93.7%. Although the improvement was 1.4 percentage points, this slight improvement has significant statistical significance in attack detection, especially when dealing with complex proxy attacks. This result indicates that the detection performance can be further improved by optimizing existing algorithms.

Future research can focus on the following aspects: first, exploring the specific implementation methods of IMP-VAE and its advantages in attack detection; second, conducting experiments on larger datasets and different detection environments to validate the model's performance; and finally, combining other machine learning or deep learning techniques with existing models can further enhance the accuracy and robustness of detection. Through these studies, it is expected to provide more efficient and reliable solutions for attack detection.

# References

[1] Salvatore Carta, Gianni Fenu, Diego Reforgiato Recupero, and Roberto Saia, “Fraud detection for E-commerce transactions by employing a prudential Multiple Consensus model,” Journal of Information Security and Applications, vol. 46, pp. 13-22, 2019. https://doi.org/10.1016/j.jisa.2019.02.007

[2] Chakir, Oumaima, Abdeslam Rehaimi, Yassine Sadqi, Moez Krichen, Gurjot Singh Gaba, and Andrei Gurtov, “An empirical assessment of ensemble methods and traditional machine learning techniques for web-based attack detection in industry 5.0,” Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 3, pp. 103-119, 2023. https://doi.org/10.1016/j.jksuci.2023.02.009

[3] J. I. Christy Eunaicy and S. Suguna, “Web attack detection using deep learning models,” Materials Today: Proceedings, vol. 62, pp. 4806-4813, 2022. https://doi.org/10.1016/j.matpr.2022.03.348

[4] Afrah Fathima, G. Shree Devi, and Mohd Faizaanuddin, “Improving distributed denial of service attack detection using supervised machine learning,” Measurement: Sensors, vol. 30, pp. 100911, 2023. https://doi.org/10.1016/j.measen.2023.100911

[5] Yaojun Hao, Guoyan Meng, Jian Wang, and Chunmei Zong, “A detection method for hybrid attacks in recommender systems,” Information Systems, vol. 114, pp. 102154, 2023. https://doi.org/10.1016/j.is.2022.102154

[6] Ayuba John, Ismail Fauzi Bin Isnin, Syed Hamid Hussain Madni, and Muhammed Faheem, “Cluster-based wireless sensor network framework for denial-of-service attack detection based on variable selection ensemble machine learning algorithms,” Intelligent Systems with Applications, vol. 22, pp. 200381, 2024. https://doi.org/10.1016/j.iswa.2024.200381

[7] Sarvjeet Kaur Chatrath, G. S. Batra, and Yogesh Chaba, “Handling consumer vulnerability in e-commerce product images using machine learning,” Heliyon, vol. 8, no. 9, pp. e10743, 2022. https://doi.org/10.1016/j.heliyon.2022.e10743

[8] Lichuan Ma, Qingqi Pei, Yong Xiang, Lina Yao, and Shui Yu, “A reliable reputation computation framework for online items in E-commerce,” Journal of Network and Computer Applications, vol. 134, pp. 13-25, 2019. https://doi.org/10.1016/j.jnca.2019.02.002

[9] Sasha Mahdavi Hezavehi and Rouhollah Rahmani, “Interactive anomaly-based DDoS attack detection

method in cloud computing environments using a third party auditor," Journal of Parallel and Distributed Computing, vol. 178, pp. 82-99, 2023. https://doi.org/10.1016/j.jpdc.2023.04.003

[10] Lucas Micol Policarpo et al., "Machine learning through the lens of e-commerce initiatives: An up-to-date systematic literature review," Computer Science Review, vol. 41, pp. 100414, 2021. https://doi.org/10.1016/j.cosrev.2021.100414

[11] Manika Nanda, Mala Saraswat, and Pankaj Kumar Sharma, "Enhancing cybersecurity: A review and comparative analysis of convolutional neural network approaches for detecting URL-based phishing attacks," e-Prime - Advances in Electrical Engineering, Electronics and Energy, vol. 8, pp. 100533, 2024. https://doi.org/10.1016/j.prime.2024.100533

[12] T. O. Ojewumi, G. O. Ogunleye, B. O. Oguntunde, O. Folorunsho, S. G. Fashoto, and N. Ogbu, "Performance evaluation of machine learning tools for detection of phishing attacks on web pages," Scientific African, vol. 16, pp. e01165, 2022. https://doi.org/10.1016/j.sciaf.2022.e01165

[13] José Manuel Ortega Candel, Francisco José Mora Gimeno, and Higinio Mora Mora, "Generation of a dataset for DoW attack detection in serverless architectures," Data in Brief, vol. 52, pp. 109921, 2024. https://doi.org/10.1016/j.dib.2023.109921

[14] Daniel Ossmann, "Attack detection in cyber-physical systems via nullspace-based filter designs," IFAC-PapersOnLine, vol. 58, no. 4, pp. 526-531, 2024. https://doi.org/10.1016/j.ifacol.2024.07.272

[15] Lohith Ottikunta, "Improved constrained social network rating-based neural network technique for recommending products in E-commerce environment," International Journal of Intelligent Networks, vol. 3, pp. 80-86, 2022. https://doi.org/10.1016/j.ijin.2022.07.001

[16] Seema Pillai and Dr Anurag Sharma, "Hybrid unsupervised web-attack detection and classification – A deep learning approach," Computer Standards & Interfaces, vol. 86, pp. 103738, 2023. https://doi.org/10.1016/j.csi.2023.103738

[17] N. Praveena et al., "Hybrid gated recurrent unit and convolutional neural network-based deep learning mechanism for efficient shilling attack detection in social networks," Computers and Electrical Engineering, vol. 108, pp. 108673, 2023. https://doi.org/10.1007/s41870-021-00773-0

[18] Punithavathi Rasappan, Manoharan Premkumar, Garima Sinha, and Kumar Chandrasekaran, "Transforming sentiment analysis for e-commerce product reviews: Hybrid deep learning model with an innovative term weighting and feature selection," Information Processing & Management, vol. 61, no. 3, pp. 103654, 2024. https://doi.org/10.1016/j.ipm.2024.103654

[19] Vinicius Facco Rodrigues et al., "Fraud detection and prevention in e-commerce: A systematic literature review," Electronic Commerce Research and Applications, vol. 56, pp. 101207, 2022. https://doi.org/10.1016/j.elerap.2022.101207

[20] D. Saveetha and G. Maragatham, "Design of Blockchain enabled intrusion detection model for detecting security attacks using deep learning," Pattern Recognition Letters, vol. 153, pp. 24-28, 2022. https://doi.org/10.1016/j.patrec.2021.11.023

[21] Tejveer Singh, Manoj Kumar, and Santosh Kumar, "Walkthrough phishing detection techniques," Computers and Electrical Engineering, vol. 118, pp. 109374, 2024. https://doi.org/10.1016/j.compeleceng.2024.109374

[22] Dan Tang, Jingwen Chen, Xiyin Wang, Siqi Zhang, and Yudong Yan, "A new detection method for LDoS attacks based on data mining," Future Generation Computer Systems, vol. 128, pp. 73-87, 2022. https://doi.org/10.1016/j.future.2021.09.039

[23] Anthony Viriya and Yohan Muliono, "Peeking and Testing Broken Object Level Authorization Vulnerability onto E-Commerce and E-Banking Mobile Applications," Procedia Computer Science, vol. 179, pp. 962-965, 2021. https://doi.org/10.1016/j.procs.2021.01.101

[24] Zoran Vučković, Dragan Vukmirović, Marina Jovanović Milenković, Slobodan Ristić, and Katarina Prljić, "Analyzing of e-commerce user behavior to detect identity theft," Physica A: Statistical Mechanics and its Applications, vol. 511, pp. 331-335, 2018. https://doi.org/10.1016/j.physa.2018.07.059

[25] Guangquan Xu et al., "Delay-CJ: A novel cryptojacking covert attack method based on delayed strategy and its detection," Digital Communications and Networks, vol. 9, no. 5, pp. 1169-1179, 2023. https://doi.org/10.1016/j.dcan.2022.04.030

[26] Man Zhou, Lansheng Han, Hongwei Lu, Cai Fu, and Dezhi An, "Cooperative malicious network behavior recognition algorithm in E-commerce," Computers & Security, vol. 95, pp. 101868, 2020. https://doi.org/10.1016/j.cose.2020.101868

[27] Quanqiang Zhou, Kang Li, and Liangliang Duan, "Recommendation attack detection based on improved Meta Pseudo Labels," Knowledge-Based Systems, vol. 279, pp. 110931, 2023. https://doi.org/10.1016/j.knosys.2023.110931

[28] Zhili Zhou, Meimin Wang, Ching-Nung Yang, Zhangjie Fu, Xingming Sun, and Q. M. Jonathan Wu, "Blockchain-based decentralized reputation system in E-commerce environment," Future Generation Computer Systems, vol. 124, pp. 155-167, 2021. https://doi.org/10.1016/j.future.2021.05.035

[29] Sumei Zhuang, "E-commerce consumer privacy protection and immersive business experience simulation based on intrusion detection algorithms," Entertainment Computing, vol. 51, pp. 100747, 2024. https://doi.org/10.1016/j.entcom.2024.100747