ISSN 2616-5775 Vol. 6, Issue 6: 16-25, DOI: 10.25236/AJCIS.2023.060603

Infinite Mixture Prototypical Variational Autoencoder for Shilling Attack Detection in Recommender Systems

Xinhao Wang[1,*]

[1]*School of Information Engineering, Nanjing University of Finance and Economics, Nanjing, China*
**Corresponding author*

Abstract: *With the rapid development of the Internet, recommendation systems have become increasingly important components of various e-commerce and social media platforms. However, recommendation systems also face some security issues, including the problem of shilling attacks. Shilling attacks refer to malicious users who use multiple fake accounts or provide fake reviews to influence the recommendation results and evaluations of the recommendation system, in order to obtain their own interests. The existence of shilling attacks not only affects users' shopping experiences, but also disrupts the normal operation of the recommendation system, affecting the platform's economic benefits.At present, some researchers have attempted to use prototype network methods to solve the problem of underattack detection with low attack filling rate and small attack scale, and have achieved good detection results. However, when faced with more complex underattack data, the effectiveness will deteriorate. We propose an attack detection method based on an infinite hybrid prototype network, and validate our method on real datasets to achieve good results in the face of more complex attacks.*

Keywords: *Recommendation system; Shilling attack detection; Variational autoencoder; Supervised Prototype network; Infinite mixture model*

1. Introduction

The traditional prototype network [1]is a supervised learning method, which is used to assign input data to predefined categories. It consists of two main components: prototype representation and distance measurement. Prototype representation is the central or representative point of each category, usually obtained by calculating the average value of samples for each category. Distance metrics are used to calculate the distance between input data and prototype representations, and commonly used distance metrics include Euclidean distance and Manhattan distance.

During the training phase, the prototype network uses labeled data to calculate the prototypes for each category, and uses distance metrics to allocate input data to the corresponding categories. During the testing phase, the distance between the input data and each prototype is calculated and then assigned to the category represented by the closest prototype.

However, traditional prototype networks have some drawbacks and shortcomings:

(1) Fixed number of prototypes: The number of prototypes in traditional prototype networks is usually predetermined and cannot be dynamically adjusted. This limits the adaptability of the model when dealing with different categories and numbers of samples.

(2) Fixed prototype distribution pattern: The traditional prototype network typically has a fixed prototype distribution pattern, where each prototype only represents a certain category, which limits the model's expressive ability when dealing with complex data distributions.

(3) Sensitivity to noisy data: The prototype of traditional prototype networks is initialized at certain positions in the sample space, and noisy data may cause the prototype to deviate from its correct position, affecting the performance of the model.

(4) Not suitable for large-scale data: As the data size increases, the computational and storage costs of traditional prototype networks also increase, making the model unsuitable for large-scale datasets.

How to solve these shortcomings of traditional prototype networks has become an urgent issue to be studied.

ISSN 2616-5775 Vol. 6, Issue 6: 16-25, DOI: 10.25236/AJCIS.2023.060603

In previous research, it was found that the main advantage of traditional prototype network models lies in detecting shilling attack users with lower attack fill rates and smaller attack scales. Especially when the fill rate is below 10% and the attack scale is less than 20%, traditional prototype network models have good detection performance. However, after further experiments, it was found that when the attack fill rate and attack scale exceed the above range. The detection effect of this method will deteriorate, and as the attack scale and fill rate increase, the effect will deteriorate more significantly.

In order to solve the above problems, this article draws inspiration from paper [2] and proposes a detection method based on infinite mixture prototype variational autoencoder (IMP-VAE). The difference between this method and the detection method based on supervised prototype network variational autoencoder is that the iterative prototype network classification module of SP-VAE uses traditional prototype network methods, while IMP-VAE uses prototype network methods based on Infinite Mixture Model (IMP).

2. Problem model

Assuming there are M users $U = \{U_1, U_2, \ldots, U_M\}$and N products $I = \{I_1, I_2, \ldots, I_N\}$ (such as products, services, etc.) on an e-commerce platform, and $R_{ui} = \{0,1,2,3,4,5\}$ indicates that user u has evaluated product i, and $R \in R^{M \times N}$ is the rating matrix. A positive feedback score of 1-5 indicates that the user has rated the product, while a score of 0 indicates that the user has not rated the product. Scoring is usually accompanied by other information, such as comments, rating time, etc. The goal of a recommendation system is to establish a model that aims to predict the scores of unknown items with positive feedback ratings and obtain the first k products of interested users. Table 1 shows an example of user rating data.

The goal of shilling attack [3]is to promote or suppress a set of target items through multiple attack strategies, and to undermine the actual recommendation effectiveness of the recommendation system. The attack profile of each attacking user is divided into four parts: selected items, loaded items, unrated items, and target items. Specifically, selecting and filling in items are used to select and randomly select items for specific attack purposes, while the target item is the item that needs to be attacked. Meanwhile, for each shilling attack, attackers typically choose one or more attack strategies to construct each attack profile. The basic attack strategies include random attack, average attack, popular attack, segmented attack, sampling attack, and Love/hate attack.

Table 1: Example of user rating data

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8
U_1	2	0	4	3	0	2	4	4
U_2	4	4	0	3	2	3	5	0
U_3	4	3	4	2	3	1	0	0
U_4	5	4	5	0	2	1	4	4
U_5	0	4	4	3	2	0	4	5
U_6	0	0	3	0	4	2	5	0
$Attack_1$	1	1	1	0	1	1	0	5
$Attack_2$	5	5	1	5	0	5	5	0
$Attack_3$	3	0	5	3	3	2	4	4

In this article, the goal of detecting shilling attack is to discover attackers who use different attack strategies. In a recommendation system, user behavior traces, user rating time, and rating strategies all indirectly reflect whether the user is an attacking user. Therefore, statistical features are usually defined for each user to represent their characteristics. For user u, a k-dimensional dense feature vector $X_u \in R^K$can be generated to construct its predefined behavior attributes, and X is used to represent the feature matrix of all users. Based on these preliminary knowledge, the attack detection task can be defined as follows:

Given the training set X_{train} and its label set Y_{train}, the purpose of detecting shilling attack is to detect attacker in the test set X_{test}, where $X = X_{train} \cup X_{test}$,$Y_{train} = \{0,1,2,3\ldots,n\}_{j=n}^{|Y_{train}|}$(where 0 represents normal users, 1, 2, 3, n, etc. represent different types of shilling attack, and j represents the number of shilling attack types)

ISSN 2616-5775 Vol. 6, Issue 6: 16-25, DOI: 10.25236/AJCIS.2023.060603

3. Method

This section will introduce the IMP-VAE(Infinite Mixture Prototypical Variational Autoencoder) shilling attack detection algorithm. Figure 1 shows the framework of the proposed model, where N and S are the classifications of normal users and shilling attackers. It consists of two modules: 1) Variational autoencoder embedding module, used to obtain robust representations of manually annotated features. 2) Use an infinite mixture prototype model to learn classifiers for shilling attack detection.

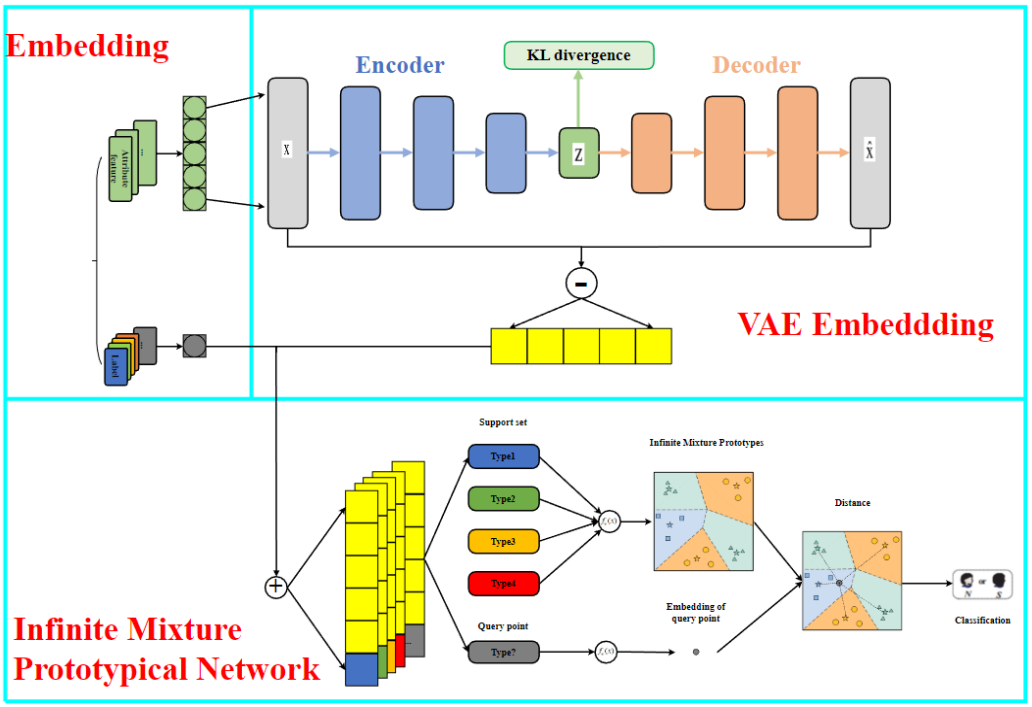

Figure 1: IMP-VAE shilling attack detection algorithm

3.1 Variational autoencoder embedding

Variational Autoencoder (VAE) is a generative model and a self-encoder. Unlike traditional autoencoders, VAE learns not a set of eigenvalues, but a potential probability distribution. VAE is a model based on Bayesian inference and deep learning, which can compress data during the learning process while preserving key features of the data, thus effectively restoring the distribution of the original data when generating it. The core idea of VAE is to represent the original data x as a distribution $p(x|z)$ of a potential variable z, while optimizing the encoder and decoder neural networks during the learning process, so that the encoder can map the original data x to the distribution of z, and the decoder can decode the distribution of z into data x similar to the original data. Meanwhile, in order to make the model more robust, VAE also restricts the distribution of potential variables z within a certain range.

This study formulates the representation of the training set X_{train} from the perspective of variational autoencoders. VAE combines Bayesian inference and simple neural networks to learn robust representations of training sets [4]lll. By optimizing the neural network parameters in the encoding and decoding steps, continuous random variables can be used to optimize VAE through backpropagation. Given the potential hidden variable z randomly sampled from a prior distribution $p\,\theta\,(z)$, VAE can be represented as an encoder $enc(x) = q_{\phi}(z|x)$ and a decoder $dec(x) = p_{\theta}(x|z)$, where $x \in X_{train}$. Using variational reasoning, it is necessary to maximize the lower bound of evidence (ELBO) [5], which is an expected lower bound for the potential variable z of VAE and the training data x. It is composed of reconstruction error and regularization term. Reconstruction error refers to the error between the reconstructed data generated by the decoder and the original data for a given potential variable z; The regularization term is used to constrain the distance between the prior distribution and the posterior distribution of the potential variable z. ELBO can be expressed as:

ISSN 2616-5775 Vol. 6, Issue 6: 16-25, DOI: 10.25236/AJCIS.2023.060603

$$\begin{aligned} ELBO &= E_{q_\phi(z|x)}[\log p_\theta(x|z) - \log q_\phi(z|x)] \\ &= E_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL\log(q_\phi(z|x) \| p_\theta(z)) \\ &= -L_{VAE} \end{aligned} \tag{1}$$

Where is the Kullback Leibler (KL) divergence, $p_\theta(z) = N(0,1)$, l_{VAE} is the loss function that needs to be minimized in the VAE model in this study. Specifically, the process of developing encoders and decoders can be developed through neural networks. In this case, VAE is usually used to transform the input vector into a low dimensional representation in the middle and downstream models in the unsupervised learning domain $z \in Z$. In order to apply VAE to the field of supervised learning, build a shared forward propagation model$\phi: X \to Z$, a decoding function $r: Z \to X$ and a prediction function $d: Z \to Y$, where d is downstream of the prediction task, and then calculate a new loss function:

$$L_1 = \frac{1}{|X_{train}|}\sum_{j=1}^{|X_{train}|}[l_{VAE}(\tilde{x}_j, x_j) + l_d(\tilde{y}_j, y_j)] \tag{2}$$

Where $\tilde{x}_j$ is the reconstruction vector of user u_j, and $\tilde{y}_j$ is the predicted label of user u_j. Through construction, we can use $l_{VAE}(\tilde{x}_j, x_j)$ to get the reconstructed loss function, and then train and predict the loss by optimizing the binary cross entropy loss, that is $l_d(\tilde{y}_j, y_j)$:

$$l_d(\tilde{y}_j, y_j) = -y_j \log(\tilde{y}_j) + (1 - y_j)\log(1 - \tilde{y}_j) \tag{3}$$

3.2 Infinite mixture model

The infinite mixture model [6] is a probabilistic generative model, which can be applied to clustering, dimension reduction, density estimation and other tasks. Its basic idea is to view a dataset as consisting of an infinite number of potential distributions, which are randomly sampled from a prior distribution. Specifically, the infinite mixture model sets the parameters of a finite hybrid model as random variables, which allows the model to automatically learn information such as the number of clusters and cluster centers.

In infinite mixture models, the Dirichlet process is usually used as a prior distribution. Dirichlet process can be seen as randomly extracting a distribution from infinite polynomial distributions, where each polynomial distribution corresponds to a category, and its probability mass function can be expressed as:

$$G = DP(\alpha, G_0) \tag{4}$$

Where G_0 represents the basic distribution, α Represents a proportional column parameter, and DP represents the Dirichlet process. Specifically, if a θ is extracted from the Dirichlet distribution, according to the definition of probability prediction, for any subset S, there are:

$$P(\theta(S)) = Gamma(\alpha)\frac{G_0(S)}{\Gamma(\alpha)}\prod_{i \in S}\theta_i \tag{5}$$

Where Gamma represents the gamma function, and $\theta(S)$ represents the projection of the vector parameter θ on the set S. It can be seen that the randomness of the Dirichlet process lies in randomly extracting a distribution from an infinite number of polynomial distributions, which can be represented by an infinite number of parameter vectors, and therefore can be applied to infinite clustering tasks.

Infinite mixture models typically use methods such as Gibbs sampling and variational inference for inference and learning. Among them, the Gibbs sampling method requires a lot of calculations, but it is the most commonly used method for non-parametric models; Variational inference methods are relatively fast, but require certain assumptions about distribution, and may require more manual design for complex models. This will result in the advantages of infinite mixture self-adaptability being limited by the difficulties in implementing and calculating Gibbs sampling and variational reasoning in infinite mixture models.

To address the above issues, Kulis and Jordan proposed the DP means [7] method, which is a

ISSN 2616-5775 Vol. 6, Issue 6: 16-25, DOI: 10.25236/AJCIS.2023.060603

deterministic, non-parametric Bayesian hard clustering algorithm suitable for the Dirichlet process. DP means iterate data points and calculate the minimum distance from each point to all existing clustering means. If this distance is greater than the threshold λ, Then create a new cluster with a mean equal to that point. Its optimization is similar to the reconstruction error target of k-means, coupled with the penalty of generating clustering.

3.3 Infinite Mixture Model Prototype Network Classification

In this section, we will mainly introduce the prototype network classification module based on the infinite mixture model.

Firstly, it is necessary to generate a prototype vector for each category, which is used to represent the prototype representation of the feature vectors of all samples under that category. For each category k, the prototype c_k represents the average of the eigenvectors of all samples defined in that category, namely:

$$c_k = \frac{1}{N_k}\sum_{i=1}^{N_k} x_i^k \tag{6}$$

Where x_i^k represents the feature vector of the i-th sample in the category k, and N_i represents the number of samples in category k. Next, allocate sample points, calculate their distance from all prototypes for each sample point, select the closest prototype as the category to which the point belongs, and update the count of prototypes to which the point belongs. Calculate the distance between sample point x and prototype c_k:

$$d(x,c_k) = ||x - c_k||^2 \tag{7}$$

Where c_k represents the prototype representation of category k. Calculate the probability of assigning sample point x to prototype c_k:

$$p_k(x) = \frac{n_k}{n_k + \alpha}\exp(-\frac{\beta}{2n_k + 2\alpha}d(x,k)) \tag{8}$$

Where n_k is the number of sample points in the category represented by the prototype c_k, and α and β are the hyperparameter of the model, which control the number and density of the prototype. For each prototype, if the number of sample points in the category it represents exceeds a threshold λ, the prototype is split into two prototypes, and the sample points of that category are allocated to the two prototypes. The calculation method for λ is as follows:

$$\lambda = 2\sigma\log(\frac{\alpha}{(1+\rho/\sigma)^{d/2}}) \tag{9}$$

Where ρ is the measure of the standard deviation of the basic distribution of clustering, and σ is the limit that approaches 0. Finally, delete the redundant prototypes. When there are no sample points in the category represented by a prototype, delete the prototype and merge the two categories where the deleted prototype is located into one category.

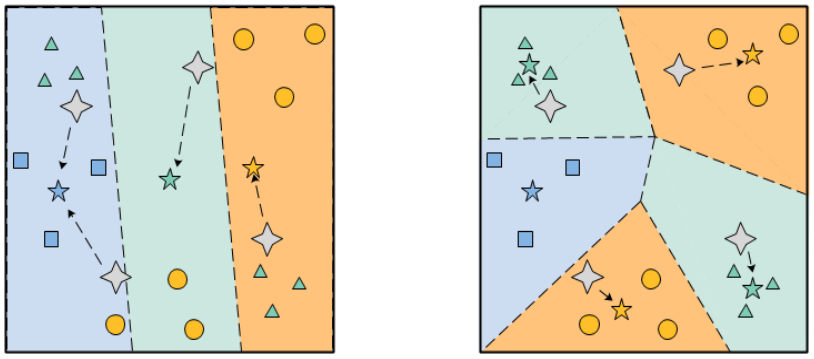

Figure 2: Traditional prototype network and the infinite mixture prototype network

ISSN 2616-5775 Vol. 6, Issue 6: 16-25, DOI: 10.25236/AJCIS.2023.060603

Figure 2 shows the classification of the traditional prototype network and the infinite mixture prototype network. Triangle, circle and square points represent various types of sample points, pentagram of different colors represent prototype points of each type, and four pointed stars represent samples that need to be classified. By comparing the two sides, it can be found that the traditional prototype network has one fixed prototype point for each type, while the infinite mixture prototype network can have multiple prototype points for each type of sample. In the case shown in the above figure, the classification method of the infinite mixture prototype network is more reasonable.

4. Experiment and Analysis

4.1 Experimental dataset

The dataset used in this experiment is Movielns-100k .The main focus of this article is to investigate the detection performance of IMP-VAE in dealing with high attack fill rates and large attack scales.

In this chapter, three types of attacks were selected for experimental processing: random attacks, average attacks, and Love/hate attacks. Based on these three attack methods, different attack profiles of attack users were constructed. In this experiment, these different types of attack users were injected into u1 base with a 15% fill rate and a 25% attack scale to form a training set for this experiment. When dealing with the test set, this experiment also set multiple different orders of magnitude of padding rates and attack scales for shilling attacks. The fill rates of the shilling attacks set in this experiment are 10%, 15%, 20%, and 30%, respectively, and the attack scales set are 15%, 20%, 25%, and 30%, respectively. The attack fill rate and attack scale were paired to construct a set of 16 attack users, which were injected into u2 base to generate 16 test sets.

4.2 Baseline

K-Nearest Neighbors Classifier (KNN): KNN classifier is a supervised learning with the first k nearest neighbors. Firstly, extract the features of the new data and compare them with each data feature in the test set. Then, k nearest (similar) data feature labels are extracted from the training set, and the most frequently occurring classification among the k nearest data is used as a new data category.

Naive Bayes classifier (NB): It is a classification algorithm based on Bayesian theorem and independent assumption of feature conditions. For a given training dataset, it first learns the joint probability distribution of input and output, and then based on this model, uses Bayesian theorem to find the output with the highest probability.

Decision Tree Classifier (DT): The decision tree classifier uses a tree structure to construct a classification model. Each node represents an attribute. According to the division of this attribute, it will enter the child nodes of the node until the leaf node. Each leaf node represents a certain category, thus achieving the purpose of classification.

Support Vector Machine Classifier (SVM): SVM classifier is a supervised learning method. It maps vectors to a higher dimensional space and establishes a maximum spacing hyperplane in that space. Two parallel hyperplane are established on both sides of the hyperplane separating data. Separating hyperplane maximizes the distance between two parallel hyperplane.

Multi-layer perceptron Classifier (MLP): It is a classifier based on multi-layer neural network, including three layers: input layer, hidden layer and output layer. There can be multiple hidden layers, and different layers of the MLP neural network are fully connected. Any neuron in the upper layer is connected to all neurons in the lower layer. Train the classifier by reading the training set, and then classify the test set.

Supervised Variational Autocoding Classifier (SVAE): It is a simplified version of the algorithm in this article that only uses variational autocoding to classify different user profiles.

Convolutional Neural Networks Classifier [8](CNN): The CNN classifier uses the deep convolutional neural network method, which only uses user rating profiles and does not use manually designed features.

Prototype network: The prototype network is a neural network model used for pattern recognition and classification tasks. Its core idea is to map the input space to a low dimensional prototype space

ISSN 2616-5775 Vol. 6, Issue 6: 16-25, DOI: 10.25236/AJCIS.2023.060603

and classify the input data onto the nearest prototype in the prototype space. Prototype networks are suitable for clustering and feature extraction of data. The prototype network consists of two layers: input layer and prototype layer. The input layer receives input vectors from the dataset, and the prototype layer consists of several prototype vectors, each representing a cluster center. Each input vector will be assigned to the cluster center represented by the nearest prototype vector.

Supervised Prototypical Variational Autoencoder classifier [9] (SP-VAE): The SP-VAE method is the method proposed in our previous research. This method mainly consists of two modules, namely the variational self-coding embedding module and the iterative prototype classification module. We have demonstrated in previous experiments that this method has good performance in detecting shilling attack with low fill rate and small attack scale, especially when dealing with cold start attack users, this method has good detection accuracy.

4.3 Evaluation parameters

In order to objectively and accurately evaluate the experimental results of the algorithm, three evaluation indicators widely used for information retrieval and statistical classification were selected in this experiment, namely Precision, Recall, and F1 [10].

Precision and Recall are closely related. Precision mainly focuses on the accuracy of model predictions, that is, the accuracy of the model's judgment on positive samples, while Recall mainly focuses on the recall of the model on positive samples, that is, the proportion of positive samples predicted by the model to the true positive samples. The calculation formula is as follows:

$$f_{pre} = \frac{TP}{TP + FP} \tag{10}$$

$$f_{rec} = \frac{TP}{TP + FN} \tag{11}$$

Among them, TP is the true positive, FP is the false positive, and FN is the false negative.

Precision and Recall are contradictory indicators, therefore, when using these two indicators for model evaluation, it is necessary to weigh these two indicators. F-1 is a comprehensive evaluation that combines the two indicators of Precision and Recall, and is a method that balances the consideration of Precision and Recall. The calculation formula is as follows:

$$F_1 = \frac{2 f_{pre} \times f_{rec}}{f_{pre} + f_{rec}} \tag{12}$$

4.4 Experimental result

In this section, we mainly elaborate on the impact of two parameters on the experimental results: the fill rate of shilling attack and the scale of shilling attack.

The first experiment verifies the accuracy of the IMP-VAE method proposed in this article when the fill rate is high and the attack scale is large. Several comparative experiments were designed. This experiment uses the Movielens-100K dataset, mainly adjusting the attack fill rate and attack scale parameters. The range of attack fill rates is 10%, 15%, 20%, and 30%, and the attack scale is set to 25% and 30%. This experiment was conducted on 8 test sets, recording accuracy, recall, and F1 values. The comparison method described in the previous section was compared with the method proposed in this article.

Table 2 lists the evaluation results in terms of Precision, Recall, and F1 values, with the best results highlighted in bold. It can be seen from Tables 2 that:

(1) As the fill rate and attack scale increase, the detection performance of traditional prototype networks and supervised prototype variational self-coding based shilling attack detection methods gradually decreases. Among these two methods, traditional prototype networks are the most affected in terms of detection, and their performance is the worst among the three. The SP-VAE method has an effect between the two, which is also affected by the increase in fill rate and attack scale. This also proves that traditional prototype network methods have the problem of poor performance when facing

ISSN 2616-5775 Vol. 6, Issue 6: 16-25, DOI: 10.25236/AJCIS.2023.060603

large-scale data. Therefore, traditional prototype network methods and SP-VAE algorithms are more suitable for small sample learning and smaller datasets.

Table 2: The detection performance of each method under 25% and 30% attack scale

Measure	Method	Filler Size							
		10%	15%	20%	30%	10%	15%	20%	30%
Precision	KNN	0.883	0.919	0.913	0.861	0.902	0.895	0.880	0.853
	NB	0.705	0.759	0.760	0.759	0.759	0.759	0.777	0.796
	DT	0.913	0.881	0.889	0.859	0.913	0.903	0.905	0.835
	SVM	0.896	0.898	0.885	0.891	0.929	0.887	0.855	0.845
	MLP	0.911	0.921	0.922	0.925	0.912	0.910	0.915	0.917
	SVAE	0.863	0.901	0.908	0.922	0.864	0.865	0.894	0.916
	CNN	0.910	0.923	0.930	0.928	0.914	0.915	0.920	0.922
	Prototypes	0.921	0.913	0.911	0.902	0.892	0.890	0.878	0.867
	SP-VAE	0.967	0.966	0.945	0.922	0.932	0.931	0.918	0.908
	IMP-VAE	**0.989**	**0.990**	**0.991**	**0.989**	**0.985**	**0.987**	**0.990**	**0.988**
Recall	KNN	0.929	0.910	0.904	0.887	0.895	0.880	0.860	0.847
	NB	0.839	0.840	0.842	0.840	0.841	0.841	0.865	0.891
	DT	0.940	0.923	0.928	0.911	0.920	0.905	0.914	0.856
	SVM	0.907	0.870	0.821	0.826	0.923	0.887	0.876	0.815
	MLP	0.913	0.915	0.923	0.925	0.894	0.902	0.911	0.920
	SVAE	0.898	0.910	0.917	0.921	0.860	0.867	0.882	0.907
	CNN	0.915	0.920	0.927	0.926	0.901	0.903	0.910	0.911
	Prototypes	0.923	0.917	0.915	0.909	0.887	0.883	0.876	0.868
	SP-VAE	0.960	0.956	0.949	0.931	0.930	0.928	0.923	0.907
	IMP-VAE	**0.986**	**0.989**	**0.991**	**0.991**	**0.987**	**0.991**	**0.992**	**0.991**
F1	KNN	0.905	0.914	0.908	0.874	0.898	0.997	0.870	0.850
	NB	0.766	0.797	0.799	0.797	0.798	0.798	0.818	0.841
	DT	0.926	0.902	0.908	0.884	0.916	0.904	0.909	0.845
	SVM	0.901	0.884	0.852	0.857	0.926	0.887	0.865	0.830
	MLP	0.912	0.918	0.923	0.925	0.903	0.906	0.913	0.919
	SVAE	0.880	0.905	0.912	0.922	0.862	0.866	0.888	0.911
	CNN	0.913	0.922	0.929	0.927	0.907	0.909	0.915	0.916
	Prototypes	0.922	0.915	0.913	0.905	0.889	0.886	0.877	0.867
	SP-VAE	0.963	0.961	0.947	0.926	0.931	0.929	0.920	0.907
	IMP-VAE	**0.987**	**0.989**	**0.991**	**0.990**	**0.986**	**0.989**	**0.991**	**0.989**

(2) After the filling rate reaches over 10% and the attack scale reaches over 15%, the variational autocoding detection based on infinite mixture prototype proposed in this article always maintains the optimal performance among the three methods. At the same time, the detection performance of IMP-VAE does not decrease with the increase of filling rate and attack scale, but shows a certain upward trend, When the attack fill rate range is between 20% and 30%, it slightly decreases but still maintains a high detection effect. Therefore, the IMP-VAE algorithm proposed in this article is feasible in the face of high fill rate and large-scale attacks.

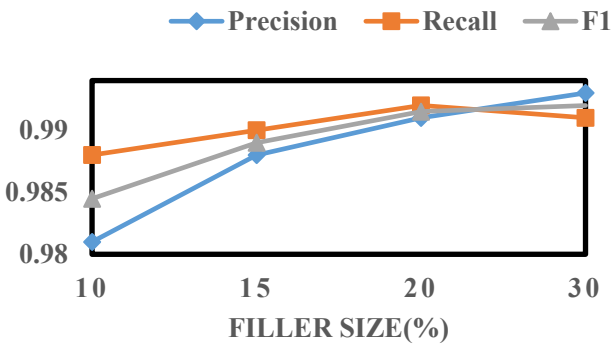

Figure 3: Detection performance under different fill rates when the attack scale is 15%

ISSN 2616-5775 Vol. 6, Issue 6: 16-25, DOI: 10.25236/AJCIS.2023.060603

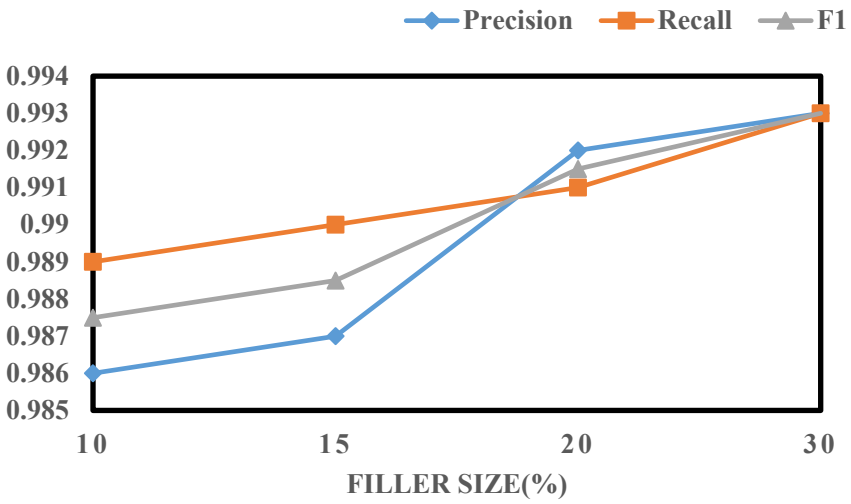

Figure 4: Detection performance under different fill rates when the attack scale is 20%

In Figures 3 and 4 the changes in accuracy, recall, and F1 values are described under different fill rate sizes and attack scale sizes. As shown in the following four figures, as the filling rate increases, the IMP-VAE model shows a basic upward trend in accuracy, recall, and F1. Although there is a slight decrease in the filling rate range of 20% to 30%, it still maintains good detection performance.

5. Conclusions

This article analyzes a series of problems in traditional prototype networks, and then proposes a prototype network method based on an infinite mixture model to replace the iterative prototype classification module based on supervised prototype variational self-coding, to form the new model proposed in this article, namely the shilling attack detection method based on infinite mixture prototype variational self-coding. This method aims to solve attack scenarios with larger attack scales and higher attack fill rates. This method is not a single prototype representation when constructing prototypes for each type of user, but a threshold construction is designed so that each type of prototype is represented as a set of equal number of prototype points. This processing method greatly improves the detection accuracy when detecting large-scale and high fill rate attack users. Experiments on real-world datasets have also confirmed the effectiveness of the IMP-VAE model proposed in this article.

Acknowledgments

This work was supported by the Postgraduate Research & Practice Innovation Program of Jiangsu Province of China under Grant KYCX21_1540.

References

[1] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 4080-4090.
[2] Allen K, Shelhamer E, Shin H, et al. Infinite mixture prototypes for few-shot learning [C]// International Conference on Machine Learning. PMLR, 2019: 232-241.
[3] Vivekanandan K. A Study on Shilling Attack Identification in SAN using Collaborative Filtering Method based Recommender Systems[C]//2021 International Conference on Computer Communication and Informatics (ICCCI). IEEE, 2021: 1-5
[4] Lopez R, Regier J, Jordan M I, et al. Information Constraints on Auto-Encoding Variational Bayes [J]. arXiv e-prints, 2018: arXiv: 1805.08672.
[5] Kingma D P, Welling M. Auto-encoding variational bayes [J]. arXiv preprint arXiv:1312.6114, 2013.
[6] Hjort N L, Holmes C, P Müller, et al. Bayesian Nonparametrics [M]. Cambridge University Press, 2010.

ISSN 2616-5775 Vol. 6, Issue 6: 16-25, DOI: 10.25236/AJCIS.2023.060603

[7] Kulis B, Jordan M I. Revisiting k-means: new algorithms via Bayesian nonparametrics [C]// Proceedings of the 29th International Coference on International Conference on Machine Learning. 2012: 1131-1138.
[8] Chua L O, Roska T. The CNN paradigm [J]. Circuits & Systems I Fundamental Theory & Applications IEEE Transactions on, 1993, 40(3):147-156.
[9] Wang X, Zhao H, Wang Y, et al. Supervised Prototypical Variational Autoencoder for Shilling Attack Detection in Recommender Systems[C]//Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II. Singapore: Springer Nature Singapore, 2023: 231-245.
[10] Mobasher B, Burke R. A Survey of Collaborative Recommendation and the Robustness of Model-Based Algorithms [J]. Bulletin of the Technical Committee on Data Engineering, 2009, 31(2): 3-13.