Digital Object Identifier

A Nearest Neighbor Similarity Method for Adversarial Samples Detection

YI REN, XIA XIE, HAI JIN, (Fellow, IEEE), and HANHUA CHEN
National Engineering Research Center for Big Data Technology and System Services Computing Technology and System Lab Cluster and Grid Computing Lab
School of Computer Science and Technology Huazhong University of Science and Technology, Wuhan, 430074, China

Corresponding author: Xia Xie (shelicy@hust.edu.cn)

ABSTRACT The k nearest neighbor algorithm is a non-parametric supervised classification algorithm with no training process. Because the process of the classification algorithm does not have a complicated iterative training process compared with other machine learning algorithms, it is simple and transparent, and its performance is quite well in practice. However, with the rapid development of deep learning, especially the development of generative model such as GAN, people can now use the distribution of data to artificially create a variety of synthetic datas, adversarial samples which results the final misclassification. It has brought a huge impact to the current traditional machine learning algorithms. Adversarial samples have a great impact on the performance of classification. At present, the research on the robustness of k nearest neighbor algorithm is limited. In this paper, we analyze the representation of data samples in the different feature space, including the data sample dimension reduction method such as GEM and LFDA, and the deep learning method to transform the original feature space to low dimension and abstract represention. Experiments show that the representation of data samples in different feature spaces will have different robust performance to adversarial samples, the improved subspace nearest neighbor similiarity algorithm are used to detect the potential adversarial sample among correctly classified true samples. These can effectively detect the misclassified threatened instance and ensure the reliability of the classifiers.

INDEX TERMS nearest neighbor search, adversarial samples, data mining

I. INTRODUCTION

WE are living in a world that a variety of data are produceed every day. It is ubiquitous that people have to process these big data to obtain the patern they want. The k Nearest Neighbor algorithm (KNN) is one of the data mining algorithms widely used in many fields that can automatically classify unlabeled sample x. It searchs the most similar k neighbors of the unlabeled input sample x in the training data set feature space and assigns the label which the most present among neighbors to this sample x [1]. At present, there has been a lot of work to optimize the efficiency and accuracy and their varients, e.g., weigh different neighbors specific weights [2] and dynamic k nearest neighbors in replace of fixed k [3]. However, KNN is a data-driven model with no prior hypothetical modeling process and distinguishes sample internal relationships based on the spatial feature distance between samples, the distribution and multi-modality of data samples will have a large impact on classification decision. Therefore, adversarial exsamples will bring huge threats to the KNN model.

Adversarial samples are characterized by finding as few disturbances as possible, and these disturbances are imperceptible to the observer but it is really a disaster for the classifier [4]. Adversarial samples have been shown to be transitive across different similar network architectures, as well as transitive between networks trained on different subsets of data. The KNN model is an important basic component of other discriminant model and used in many areas with high requirements for safety performance, for example, in the field of autonomous vehicles, someone can intentionally alter a normal danger warning sign slightly, this attack is so negligible to detect by human eye, and misclassified to another innoxious sign by the classification model. It is very common in the field of patern recognition but leading to serious consequence.

Currently, there is a little amount of work focused on the adversarial sample to KNN models. The method proposed in [5] mainly analyzes distributional properties and data samples size on robustness, proposing a method of changing the state of the original data set to resist the attack of the adversarial examples. This work theoretically deduce the impact of the data size and the values of k on the robustness of the classifier model, which requires complex calculations of the original data set in advance. Another research direction of KNN classifier is to reduce the dimensionality of the data and compress the data into a smaller feature space [6], so that the data can be more dense and better to represent the original distribution of the data. However, the adversarial samples can be distinguished in the original dimensional space, which will be compressed into "normal" samples in the low dimensional because singularity disappears and brings more serious problem. So just reducing the dimension is not enough, we need more rigorous and abstract data representation.

Deep learning is essentially a feature extraction process [7]. It passes the original input data through a multi-layered learning structure model to obtain a layer-by-layer transformation of the data features , numerous neurons can transform the original data samples into a more abstract and flexible feature space and finally obtains the deep feature expression of the data, and could dig out novel feature representations that are difficult to mine by traditional matrix decomposition methods. Based on the deep feature space, we perform nearest neighbor similarity search in the subspaces of each class, taking training samples into consideration as much as possible. Finally, based on the trade-off consideration and analysis of similarity score of each class to determine the input sample, we can judge whether is a adversarial sample.

The main contributions of this paper are as follows:

- Firstly, continue to study and expound the influence of adversarial exsamples on the parameter-free machine learning algorithm such as KNN algorithm, and propose a new KNN method model to use the hidden layers activations of deep feature for neural network to detect and process the adversarial exsamples.
- Secondly, the distance between two instance samples in the feature space can reflect the similarity degree. The feature space of the KNN model is generally the n-dimensional real vector space R^n. Euclidean distance is commonly used to calculate the distance between data samples in feature space, a more general calculation formula is L_p distance. Besides, tangent distance is a distance measure that is not sensitive to transformations (such as translation, rotation, scale transformation, etc.) and is used to replace the traditional euclidean distance. [8]. Based on the above considerations, a measurement method different from directly calculating the distance between neighbors is adopted. Calculating the feature space of the samples to be classified for each class is used as the similarity calculation method. We realize that the original KNN does not take into account the limitations and complexity of the training data set feature space, so in this paper we performs neighbor search and analysis the subspace of each class and finally consider all the similarity scores of each classes to achieve a robust KNN moedl.

This paper is organized as follows. Section2 reviews the state-of-the-art related work. Section3 gives an overview of the technical challenges and then presents the our method's architecture. Section4 presents the methodology in detail. We give an evaluation of in Sect.5. Finally, we draw an conclusion and indicate some directions for future work.

II. RELATED WORK

A. WORKS ON NEAREST NEIGHBORS

The setting of the KNN algorithm is very simple and efficient, and does not require training parameters. It was first proposed by T. M. Cover and P. E. Hart for the classification problem [1]. KNN algorithm is listed as one of the top ten machine learning algorithms and one of the necessary algorithms for learning machine learning.

There are a lot of optimizations about KNN [1], most of these tasks focus on improving the efficiency of the KNN algorithm and the accuracy of the classification [9]. KNN is a regression method that does not require training parameters. Usually, the value of k can be set according to the size of the data set empirically and k is fixed. [3] proposes an improved KNN algorithm, which is indicated as Dk-NN, by using an interval of parameter k in replace of fixed parameter k, and obtain variation tendency curves under different k values to assist decision-making. The article [2] proposes a weighted KNN method: when performing neighbor space metrics,it sets different weights for different neighbors, and design a pool of classifiers with different weights. The classifier group finally determines the final classification result based on the weighted result of the boost classifiers to improve the classification accuracy.

In addition to considering the selection of the k value and the weighted metric scheme for the KNN algorithm, there are some methods that focus on establishment a spatial index structure to speed up the process of comparison and calculation [10]. In a pattern classification problem, it usually involves the use of high-dimensional feature data, which can easily make the classifier very complicated and time-consuming, another solution is to optimize the data sample feature space and data set. Irrelevant or redundant features will increase the complexity of the classification process and consume more resources and time to complete the classification task, and it is likely to reduce the classification accuracy of the classifier [6]. In [11], the importance of feature selection for discriminant models is analyzed, and some common methods are introduced. Some researchers are committed to streamlining and transforming the data set to improve the retrieval efficiency and convergence of traditional KNN classification [12] [13]. [14] deal with the imbalance of data samples based on the distribution density of the data samples. Earlier research on the robustness of KNN used soft labels instead of crisp labels to classify and solve the problem of overlapping classes between data samples [15].

B. WORKS ON ADVERSARIAL SAMPLES

As the Iran Good fellow et al. first proposed and defined the adversarial sample, by adding a small amount of perturbation to the original data, then the modified data and the original data were used to derive different classification labels [4].

Such as it is jammy for the human to recognize a picture of a dog with minor changes without interference, this change is sensitive to the classifier, and the classifier incorrectly classifies it into a cat resultly. The main research work on current adversarial samples can be divided into two kinds:one is how to generate adversarial samples and the other is to defense against adversarial samples. Regarding the defense work against the adversarial sample, in the robust research work of the algorithm model, most of the current research on the adversarial sample is concentrated on the network [5].

At present, the latest research on parameter-free machine learning algorithms such as KNN is still rare [16] [5]. However, the system analyzes different data sets robust and the algorithm's robust. It takes a huge computational cost to get the final robust NN algorithm, and real-time performance is difficult to guarantee. In this paper, different research methods are proposed. Through the deep feature and redesigned nearest subspace-based metrics, an efficient robust KNN algorithm is implemented to detect adversarial samples.

III. BACKGROUND AND PROBLEM STATEMENT

A. KNN BASIC THEORY

The main process of classification of k nearest neighbor is as follows: given a dataset $D_{\text{train}}=\left\{(x, y): x \in R^{N}, y \in Y\right\}$, the feature space of KNN model is generally N-dimensional real vector space R^N, each x in D_{train} is a point in the N-dimensional feature space R^N, describing a specific instace in real life, y is the label corresponding to x, and $Y=1$ or 0 in the task of the binary classification problem. for each sample x_i, x_j in the dataset D_{train}, the L_p distace between samples that represent the similiarity is difined as $x_i, x_j \in D_{train}$, $x_i=\left(x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(n)}\right)^{\tau}, x_j=\left(x_j^{(1)}, x_j^{(2)}, \cdots, x_j^{(n)}\right)^{\tau}$:

$$L_p\left(x_i, x_j\right)=\left(\sum_{l=1}^{n}\left|x_i^{(l)}-x_j^{(l)}\right|^{p}\right)^{\frac{1}{p}} \qquad (1)$$

and find ordered k nearest neighbors with the smallest L_p distance:

$$Neighbors(x, X)=\{(x_1, y_1)\ldots(x_k, y_k)\} \qquad (2)$$

Of course, the farther away among the samples, the smaller the mutual influence is. But the multimodality data shows that even though samples from the same class may have a large distance in the feature space with each other. For example, a disease such as the cold, with multiple possible symptoms (e.g. fever, runny nose, joint pain), the special symptoms are different clusters within the same class. KNN is based on the assumption that the data of the same class must be within a local scope, and the data of different class must be within different range. This assumption may be valid on some datasets, and it has also been confirmed. But there are also some specific scenes that make this assumption of KNN inefficient, that is, one of the class of data exists in different clusters in the feature space. Therefore, we can not be neglected this characteristic of the data when doing the classification task. We try to take all samples in each class into considerration.

B. ATTACK MODEL FOR ADVERSARIAL SAMPLES

The generated adversarial sample model [17] can be defined as: given the normal input data sample x , such as a digital image, and a trained classification discriminant model expressed as $y=f(x)$, after adding noise perturbations η to the normal data sample x' ($x'=x+\eta$) that are hard to detect by human eys, the target classification discriminant model misclassified $f(x+\eta) \neq f(x)$.

The generation of adversarial samples can be divided into targeted attacks and non-targeted attacks according to whether it targets a specific category. Targeted attacks are to generate adversarial samples and make the classification discriminant model classify them into a specified category, while non-targeted attacks refer to the adversarial samples that make the classification model misclassified without caring about the classification results.

A typical target attack adversarial method is under a bounded constraint L-BFGS algorithm [18] to generate adversarial samples for targeted attacks. The author proposes a bound constrained optimization problem [19] to obtain the smallest adversarial perturbation η:

$$\eta=\underset{\eta}{\operatorname{minimize}}\|\eta\|_2+L\left(x_i+\eta, i'\right) \text{ s.t. } x_i+\eta \in[0,1]^m \qquad (3)$$

where x_i represents the original data samples with label i, η is the noise values of added perturbations to the normal data samples and $L(,)$ is the cost function computed between the original instance with class i and the generated target class i' (different from the original image category i). Searching for an approximate solution to this problem, we can find the adversarial perturbations that makes the classifier classify as the target class and is hardly noticeable by human eyes.

Ian Godfell et al. [4] proposed the Fast Gradient Symbol Method (FGSM) to directly and quickly calculate the adversarial perturbation based on the linear characteristics in high-dimensional space. It is a typical non-targeted attack model that calculates all feature dimensions gradient of the input data to obtain the maximum adversarial disturbance. The main idea of the algorithm: calculate the model gradient and then obtain the function related to the weight vector w. The loss will increase when each pixel variable value moves a small distance in the direction of its gradient, and then there will be a maximum change in the classification results. The sign function ensures that the direction of change is consistent with the direction of the gradient.

Specifically, the problem can be modeled as follows Eq. 4, where ∇_x is the gradient direction of parameter for the classification model, y the desired target class related to the input data sample x, and $J(,)$ is the loss of the classification model. Although the change in each dimension is small, as the dimension increases, we can generate adversarial samples in the direction of increased loss, which will cause misclassification different form the desired target class. The cost function can be linearized near the current value of $theta$ to obtain the optimal max-norm constraint perturbation [20]:

$$\eta=\varepsilon * \operatorname{sign}\left(\nabla_x J(x, y)\right) \tag{4}$$

It should be noted that the gradient can be calculated more easily by using the back propagation process of deep learning, so as to obtain the disturbance variable η and ϵ is the hyperparameters controlling the magnitude and fooling power of the added perturbations. The adversarial input is a linear combination by $x'=x+\eta$.

C. DEEP FEATURE SELECTION

Feature selection is generally given an original feature set, from which a subset of features is selected to represent the original features, so that based on the improved representation the best classification results are achieved on classification models. Irrelevant or redundant features will increase the complexity of the classification process and consume more resources and time to complete the classification task, and it is likely to reduce the classification accuracy of the classifier. Therefore, for general pattern classification problems, feature selection is very necessary. The compression and technique of dimension reduction of features can be considered as the representation of data sample, mapping data into a lower dimension without destroying the internal relevance information of the data samples. A typical application is processing time series data [21]. The features under the new representation have better distinguishing characteristics and improve the classification performance. There are some other feature processing methods: PCA dimensionality reduction, word embeding [22], matrix decomposition [23] and so on.

Deep learning is essentially a model of feature extraction as a subset of deep learning representation learning methods. It can automatically extract different levels of feature sets from the original data through the greedy algorithm's layer-by-layer training strategy, and these feature sets represent different levels of abstraction in the original data with special concept and meaning due to the development of back propagation technology. Deep learning is a feature learning model that can automatically extract excellent features from the original data and has achieved great success in the fields of images, speech, text and so on. Through deep learning, you can get features at different depth levels from the original input data. And through such a deep learning process, the original input The data information hidden in the data can be extracted and abstracted layer by layer. The deeper the number of layers, the deeper the data concept represented by the extracted features is, which cannot be expressed and obtained by the shallow structure.

The trained neural network is used to extract the deep features containing the semantic information from the original data samples to obtain a reliable training sample feature set. These representations may be far removed from human understanding [24] but the results of the experiment show that the features extracted by the multi-layer neural network finally have a good effect on the classification [25].

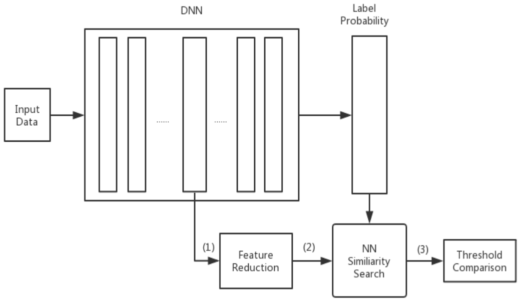


FIGURE 1: Framework of Adversarial Samples Detections

IV. IMPROVED ROBUST CLASSIFIER METHODOLOGY

This section presents the detailed design of the three key modules as shown in Fig 1: (1)deep feature based feature represention and reduction; (2) nearest space classifier - all samples considered nearest neighbors similarity search, and (3) detect adversarial samples.

A. DEEP FEATURE BASED FEATURE REPRESENTION AND REDUCTION

Currently, deep learning is powerful and effective, and it has achieved remarkable results. The reason why deep learning has achieved such great success is partly by building a deep enough network and enough neurons, converting original feature to a more abstract space, geting a better representation of the data, and completing the task of linear partition that was difficult to handle in the previous feature space. Different abstract representations of the data are obtained at each layer of the neural network, while retaining the relevance relationship between the original data as much as possible. The output features of the intermediate activation layer are extracted from the pre-trained deep neural network to build a deep feature data set. These layer-by-layer non-linear transformation features often have excellent representation capabilities.

For different data sets, we will first use DNN to train a reliable classifier (classification accuracy is above 95 percentenge), and set different numbers of neurons. We then get different neuron weight parameter matrices, that is, get different dimensions deep feature of data samples. Using the obtained deep feature as represrntion of original data samples, we input it into KNN classifier to calculate the similarity on the feature subspace. Under the specified constrained attack budget, perturbation is added to each feature pixel along the direction of maximum loss, which eventually leads to agglomeration effects and misclassified especially in high-dimensional features. Therefore, the features dimensionality reduction method and the parameterless discrimination model can play a certain role in this attack [20]. By comparing the similarity scores between the deep features in the same classes, we can make a valid and reliable judgement. We also can operate feature dimensionality reduction method on obtained deep feature as done on raw data, such as Local Fisher Discriminant Analysis [26].

Based on the above feature processing, we expect the new feature representation to have better trade-off between discrimination and robustness of row samples, and it is more friendly to parameterless models based on spatial distance metrics as showed in Eq. 5:

$$Distance\left(\boldsymbol{V}, \boldsymbol{x}, \boldsymbol{X}_{i,j}\right) = \left\|\boldsymbol{V}\boldsymbol{x} - \boldsymbol{V}\boldsymbol{X}_{i,j}\right\|_2 \tag{5}$$

The V is the transformation matrix based on Local Fisher Discriminant Analysis or the Generalized Eigenvector Method, x is the input test sample and $X_{i,j}$ are the data samples of class i in the training set. The projection transformation matrix V is designed to make the data samples in training set closer of the same class, while samples from differet class are separated in the projected subspace.

B. NEAREST SPACE CLASSIFIER - ALL SAMPLES CONSIDERED K NEAREST NEIGHBORS SIMILARITY SEARCH

The KNN algorithm finds the k nearest samples in the entire training data set as the neighbors of the sample to be classified, so the method of searching the neighbors is simple and efficient. But it is also vulnerable. For example, the k-neighborhood range of the sample to be classified is full of different labeled samples. Since the multimodality and randomness of the data samples are, different classes of data will also appear together in the same field. This scenario shows that the data has universal multi-modal properties, and some characteristics of the data are usually scattered in multiple samples, and these samples deviate from each other and do not exhibit the characteristics of cluster. Therefore, in classification discrimination based on neighborhood of test sample, considering more data samples can often obtain more general information for each category, and it is robust to classification tasks.

This paper proposes two feasible methods to demonstrate the ideas mentioned above. First, when searching k nearest neighbors, neighbors are not found in the entire training data set but in the subspace of each class, that is, we find k neighbors in each subclass. Finally $k * L$ nearest neighbors of a instance y to be classified are based on the smallest cumulative sum distance $l_i(\boldsymbol{y})$ to each subspace among all classes $l_i(\boldsymbol{y}) = \min_{i\in\{1,\ldots,L\}} \sum_1^k \left|x_y^{(l)} - X_i^{(l)}\right|^p$, and we assign y to the class i, L is the total amounts of all classes, we defined the method as SK-NN. The other is based on the idea of the Nearest Space algorithm [27], all data samples $\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n_i}]$ of class i can expanded a feature subspace as the basis, we no longer calculate the distance as the similarity between the data samples, but calculate the distance between the test sample and the feature subspace spanned by the training data set of each class and the lable of y is that has smallest distance one. We call the method as AK-NN.

Different from traditional using the fixed k neighbors as the final criterion, this paper calculates the test sample y among all the samples of each class in the training data set as Eq. 6 to obtain the distance similarity. That is to say, using all the training data in each class to fit y, among all the $l_i(\boldsymbol{y})$ which have the smallest residual with the original y, i is the final class label, $\phi(\boldsymbol{y})$ indicates deep feature extracted from neural networks.

$$d_i(\boldsymbol{y}, X_i) = \min_{\boldsymbol{\alpha}_i \in \mathbb{R}_i^n, i\in\{1,\ldots,K\}} \left\|\phi(\boldsymbol{y}) - \phi(\boldsymbol{X}_i)\boldsymbol{\alpha}_i\right\|_2 \tag{6}$$

To avoid overfitting and smooth model obtained from the training process of back propagation, add the $L2$ regularization term:

$$d_i(\boldsymbol{y}, X_i) = \min_{\boldsymbol{\alpha}_i \in \mathbb{R}_i^n} \left\|\phi(\boldsymbol{y}) - \phi(\boldsymbol{X}_i)\boldsymbol{\alpha}_i\right\|_2 + \lambda \left\|\boldsymbol{\alpha}_i\right\|_2^2 \tag{7}$$

Based on the above design, there is another advantage, we can search the nearest neighbors in each subclass in parallel on a multi-core machine, so that the time-consuming computing similarity distances of each class can be reduced to some extent. The final classification criterion is based on the score of nearest neighbors of each class. The class of the neighbor with the highest score(inverse of similiarity distance value) is the class of the sample to be classified. By this way, it is possible to take the multi-model nature of the data and the distribution differences of the data samples into consideration, and detect the adversarial sample finalllty.

Designing similarity score calculation method is listed as follow: The sample to be classified x is compared with all other samples X_i of each class i ($i \in (1, L)$ classes) in the deep feature space and KNN classifier assigns lable y according to the ordered k neighbors $Neighbors(x, X_i) = \{(x_1, y_1)\ldots(x_k, y_k)\}$ to test sample x, neighbors (x_k, y_k) are calculated and searched according to the predefined two methods SK-NN and AK-NN as Eq.7. Obtained all the distances between test y and nearest neighbors, we can get a score distribution, then the probability that the test sample x belongs to category i is $\Pr(x = i)$ as Eq. 8.

$$\Pr(x = i) = 1 - \frac{\sum_{i=1}^{K} distance(x, x_i) D(y_i = i)}{Neighbors(x, X_i)} \tag{8}$$

When the residual of $distance(x, x_i)$ is smaller, the probability of sample x labeled as $\Pr(x = i)$ is more confident and we regard $distance$ as a weight parameter. Here we use the distance value to represent it, but it can also use other forms to represent it, $D(y_i = i)$ is an indicator function, when the condition is true, output 1, otherwise output 0. When it is greater than a certain threshold, we think it is trustworthy. Of course, under different L_p distance calculation on the obtained deep feature with the LFDA and GEM method we can get different $\Pr(x = i)$. This is the basis we use to determine whether it is an adversarial sample.

C. ADVERSARIAL SAMPLES DECTION

First, we are based on a trained deep neural network which structure is designed as eight layers composed of convolution layers with max pooling and fully connected layers refered to [28]. We choose the fifth layer of the network structure as the source of deep features, and the last layer as the output layer, and output the probability distribution vector of the category of the test sample, so that we can get a corresponding deep feature dataset of the training dataset. Based on the data representation obtained from the deep feature, we input it into a fast-computing non-parameter neighbor discriminant model. We can get the nearest neighbor similarity score based on this distance similarity model, which is compared with the result vector of the DNN. If the confidence score is substantially positive correlation with the result vector, we can consider the test data to be clean. If the similarity score of the discriminant model is below a certain threshold, we consider the test data sample to be attacked and it is an adversarial sample. The classification results are unreliable and discarded.

In order to ensure the robustness and quality of the classification results, and considering the perturbations accumulation of various feature dimensions in a high-dimensional feature space, we designed a compression and reduction stage of deep feature. In the feature compression phase, according to the categories to which the deep features belong, and the visual similarity between the categories, we use the GEM method and the LFDA method to design the compression transformation matrix V as shown in Eq. 9 and optimize the data representation.

$$\boldsymbol{V} = \arg\max_{\boldsymbol{V}} \frac{\boldsymbol{V}^T \boldsymbol{M}_i \boldsymbol{V}}{\boldsymbol{V}^T \left(\boldsymbol{M}_j + \frac{\gamma}{d}\,\mathrm{trace}\left(\boldsymbol{C}_j\right)\right) \boldsymbol{V}} \tag{9}$$

where V is the obtained compression transformation matrix, M_i represents the difference between the data we want to enlarge, including singularity, covariance or the spatial distance between categories, and M_j is the characteristic feature of the data we want to aggregate, which can be spatial distance within the class, $\frac{\gamma}{d}$ is a scaling parameter and d is the feature dimension of the data sample.

Next, we need to calculate the similarity for the features we have processed, and find neighbors to assist decision-making. As we all know, the KNN model based on the L_p distance is very sensitive to the transformation (common inversions, translations, rotations and etc.) between data samples, so we employ the improved tangent distance based on [8] to measure the distance between data samples under the transformation space. In the method framework proposed in the previous chapter, we use two schemes to determine the domain of the neighborhood, and accumulate the nearest neighbor distance in the category subspace as the similarity score $S(x, X_i)$. The probability $Pr(x = i)$ (distribution of similarity scores in different classes of feature subspaces) is used as a sign of the confidence of the classification result, and the output layer of the deep network is compared to determine whether it is an adversarial sample.

TABLE 1: Classification Accuracy under Adv. Perturbations

	p_0	p_1	p_2	p_3	p_4
CIFAR-10	0.96	0.88	0.80	0.73	0.62
MNIST	0.95	0.86	0.82	0.70	0.59
Caltech	0.95	0.96	0.78	0.72	0.60

The threshold associated with the $s(x, X_i)$ is a hyper-parameter that requires multiple experimental comparisons and trade-offs. If we choose a too large threshold, then some normal samples with low confidence scores will be classified as adversarial samples, that is False Negative(FN) in the binary classification problem. If we choose a too small threshold, the adversarial samples will escape the detection of the classifier model and be classified into normal samples, that is False Positive(FP), which will affect the practicality of the model. The choice of the probability threshold is a trade-off to balance the problem above, the choice of our threshold needs to take into account the ratio of True Positive(TP) and False Positive for the final result, TP is normal sample that is classified as a normal sample correctly, we need to ensure that as many normal samples as possible are correctly classified.

V. PERFORMANCE EVALUTION

In this chapter, we will conduct experiments to evaluate the effectiveness and robustness of our models. The proposed method has mainly been evalauted as detecting adversarial samples when a normal sample is attacked by Fast Gradient Sign and L-BFGS and to give a confidence score of the result of the classification. When the output of confidence score above the trade-off threshold it means the sample to be classified is normal; otherwise it indicades the sample is attacked and is an adversarial sample.

We use three datasets: MNIST, CIFAR-10 and Caltech to verify the proposed model. Pictures of objects belonging to 101 categories of the Caltech. It is about 40 to 800 images for per category. Most categories have about 50 images. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class.

Table 1 shows the results of classification accuracy under different proportions of adversarial samples in the training data set, p_0 is with no adversarial samples while p_1 is with 5 percentage adversarial samples, p_2 is with 10 percentage adversarial samples, p_3 is with 15 percentage adversarial samples, p_4 is with 20 percentage adversarial samples. For the three data sets, the appearance of the adversarial samples did have an impact on the classification task. When the number of adversarial samples gradually increased, the accuracy of the classification decreased significantly. In Table 2, we record the effect performance of detecting adversarial samples with different discriminant model schemes. Among them, the traditional KNN discriminant model is quoted from the work [20]. We can see that the precision rate is improved compared with the deep network without the discriminative model, but the recall rate is relatively low,

TABLE 2: Classificaton Prediction Performance Comparison.

Method	FGS attack		L-BFGS attack	
	Precision	Recall	Precision	Recall
KNN	0.7324	0.618	0.7371	0.6142
SK-NN	0.7621	0.6685	0.7163	0.6762
AK-NN	0.7831	0.7222	0.7843	0.7527
AK-NN with GEM	0.8142	0.7599	0.8695	0.7984
AK-NN with LFDA	0.8335	0.7933	0.8386	0.8148

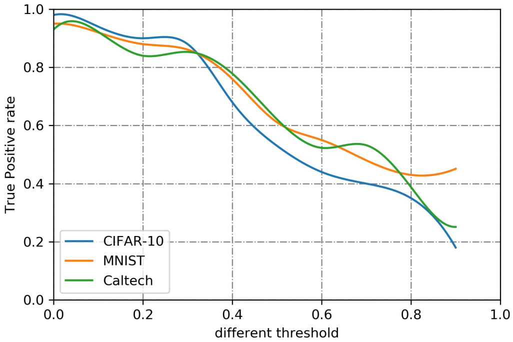


FIGURE 2: The Ratio of TP under Different Thresholds

which indicates that the model has the problems of False Negatives, that is, it will treat clean inputs as adversarial samples and discarded totally, which needs to be improved. The next four lines are the SK-NN and Ak-NN methods proposed in this paper. Based on the redesigned distance measurement and data compression methods, the precision and recall rate have been improved significantly, and the problem of False Negatives has been solved better. In the subsequent experiments, we will continue to demonstrate our improvement of this achievement. Among them, Ak-NN with LFDA data dimensionality reduction method obtains the best performance.

We performed experiments on three data sets and measured the proportion of TP in the classification results under different threshold conditions. In Fig.2 as the threshold increases, we can see that the trend of decrement is not as fast as each other, which means that different data sets, also has original robustness naturally, as the discrimination threshold increases, the data set that can retain more True Positive samples is a more robust data set for research. We find that when the threshold is set to 0.6, we can still retain the next 50 percent of the normal sample in MINIST and Caltech, but only 40 percent of the normal samples are retained in the CIFAR-10 dataset, and the rest are incorrectly classified as adversarial samples because they are not with confidence scores enough assigned by the AK-NN. Therefore, these data sets which are inherently robust are studied. The deep features extracted from the samples in these data sets are not sensitive to the malicious disturbance, which helps to further improve the defensive performance of the model against the adversarial sample.

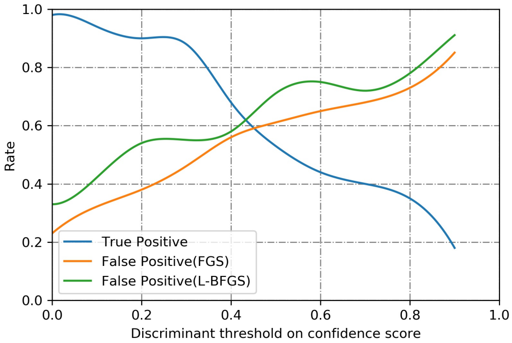
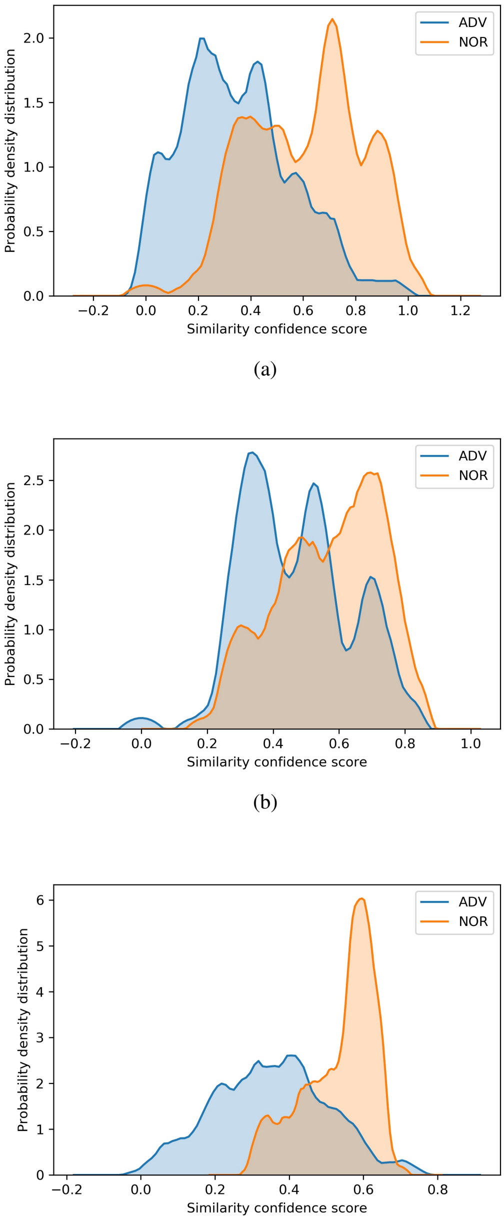


FIGURE 3: TP and FP Rates with Different Threhold on Confidence Score

Then we devote mind to the similarity score calculated based on the best AK-NN with LFDA. In Fig.3, we show the rate distribution for True positive and false positive rates with different discrimination threhold on confidence score. The results show that it is not necessary to use a very low threshold such as 0.2, and the adversarial samples produced by L-BFGS and FGS can be correctly filtered out more than 50 percent while retaining more than 90 percent of clean data samples classified properly. In the previous work, can only use low thresholds below 0.2, and so the similarity scores are not effective when classifying images. The method proposed in this paper has been improved.

From the experimental results under three data sets in Figure 4, we can see that the confidence score given by the discriminant model has a clear boundary, which illustrates the effect of our work. Our method can better distinguish between the adversarial sample and clean sample. Most of the similarity scores of the adversarial samples are concentrated at around smaller value and the normal samples have similarity scores greater than the value of adversarial samples. In Figure 4(b), we find that a high proportion of the adversarial samples also have higher confidence scores, especially in the case that the classified samples have more categories more than 100. It is necessary to calculate the similarity with all the feature subspaces, and sometimes the generalization of the adversarial sample detection is weakened. But by comparing these misclassified samples and their nearest neighbor samples, we find that those misclassified samples with very high confidence scores usually reflect the similarity in visual patterns between classes, and these similarities are irrelevant the input adversarial samples which is likely producing a batch of 'real' samples by the generating models. Overall all the results show, that based on the method proposed in the paper, the adversarial sample can be effectively detected from the normal sample, so some simple metrics on those densities could be computed on-line to detect against a malicious attack, rejected the sample, and improved robustness of the classification model.



(a)

(b)

(c)

FIGURE 4: Scores Distribution for the Adversarial and Clean Samples with the AK-NN under Different Data Sets.

VI. DISCUSSION AND CONCLUSION

We injected specially generated adversarial samples into the original normal data set, and the classification accuracy of the KNN classification model showed a significant drop. This shows that KNN a non-parameter, data driven machine learning algorithm is sensitive and vulnerable to adversarial samples. Secondly, the different representations of the data, the feature space of the original data is changed by means of dimensionality reduction, mapping, transformation, etc.. The representations of the original data under different feature spaces have different degrees of robustness. Choosing an more abstract, different dimension representation instead of the original data representation can protect against the adversarial attack to a certain extent, or make the adversarial samples more difficult to generate. Finally, for the data with multimodality, cluster distribution, and propagation properties, when classifying and predicting the label, the similarity scores of KNN are calculated according to the distribution of different categories. Based on the characteristic subspace of each category, as many samples as possible to ensure the integrity and comprehensiveness of the data samples. The calculated similarity score can be used as a kind of credibility judgment for the final result, which can effectively detect the adversarial samples. The classification result is more credible, the proportion of misclassification decreases.

We design a novel method for detecting adversarial samples. Compared with previous achievements, we focus on the non-parameterization method KNN which is more widely used and implement in principle. Our experiment shows when there are a certain number of adversarial samples, the classification accuracy of our KNN model will be significantly reduced. Then, we use the deep feature as the data representation. By using this abstract data representation method, it is more powerful than the original data representation combined with the feature subspace-based neighbor similarity calculation method. It can effectively guarantee the classification result, and make the final classification result reliable and effective. Not only can we detect the sample being attacked, but also we ensure the classification performance of normal samples. We do not exaggerate the proportion of the adversarial samples to reduce the practical performance of the KNN classification.The relational (non-i.i.d.) nature of the data might improve robustness since the KNN similiarity scores are computed for all samples jointly, rather than for individual samples in isolation. When performing classification prediction, KNN needs to refer to the distribution of the labels of the samples in the neighborhood, if the neighborhood of the voting decision is small, the attack effect can be achieved by only modifying a small number of samples. But if the neighborhood is large enough, the cost of the attack will be large, because the number of samples in the neighborhood to be modified will increase as the neighborhood increases and evevtly generate adversarial samples more arduously. This will result in a poor adversarial attack performance, and the impact on the KNN model will be reduced.

Moreover, we observed the experimental results and found the following phenomenon: some adversarial samples are easily detected by our discriminant model, that is, they have a lower confidence score $s(x, X_i)$; but some adversarial samples are difficult to detect, the confidence scores of the adversarial samples are confusing, either the distributions are roughly uniform without directivity or having a higher probability consistent with the original category by DNN. The second situation deserves our attention, we observe and compare the adversarial samples and the normal samples in the training data set, and we find that these adversarial samples tend to have similar patterns, including similarities in the categories themselves and similarities in certain visual patterns, that is also a widespread phenomenon in nature, such as between Japanese Spitz and white foxes, which belong to different ethnic groups but in appearance there is a lot of similarity on it.

How to use the adversarial sample for the proper purpose: Generally speaking, the adversarial sample is a kind of malicious application to artificial intelligence. It is often associated with deception, attack and camouflage, which will pose a great threat to the artificial intelligence system. However, more attention should be paid to the correct use of adversarial sample technology to make it play a positive role. In the aspect of privacy protection, using artificial intelligence system to collect and analyze the image data uploaded to the network is a great threat to the user's privacy. At this time, we can camouflage or hide the image data through the adversarial sample technology, and cheat the image recognition system based on neural network, so as to protect the privacy of users. Therefore, it is worth studying how to apply the adversarial samples to the proper purpose.

In general, our method has obtained satisfactory results in terms of classification accuracy, scalability, and practicality, and we will continue to expand our research in the next work.

ACKNOWLEDGMENT

REFERENCES

[1] Thomas M Cover, Peter E Hart, et al. "Nearest Neighbor Pattern Classification". *IEEE transactions on information theory*, 13(1):21–27, 1967.

[2] Manuele Bicego and Marco Loog. "Weighted K-nearest Neighbor Revisited". In *Proceedings of the 23rd International Conference on Pattern Recognition*, pages 1642–1647. IEEE, 2016.

[3] Xiao-Feng Zhong, Shi-Ze Guo, Liang Gao, Hong Shan, and Jing-Hua Zheng. "An Improved k-NN Classification with Dynamic k". In *Proceedings of the 9th International Conference on Machine Learning and Computing*, pages 211–216. ACM, 2017.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In *Proceedings of 3rd International Conference on Learning Representations*, pages 1–11, 2015.

[5] Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. "Analyzing the Robustness of Nearest Neighbors to Adversarial Examples". In *Proceedings of the 35th International Conference on Machine Learning*, pages 5133–5142. PMLR, 2018.

[6] Shih-Wei Lin and Shih-Chieh Chen. "Parameter Tuning, Feature Selection and Weight Assignment of Features for case-based Reasoning by Artificial Immune System". *Applied Soft Computing*, 11(8):5042–5052, 2011.

[7] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. "Deep Image Retrieval: Learning Global Representations for Image Search". In *Proceedings of European conference on computer vision*, pages 241–257. Springer, 2016.

[8] Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. "Transformation invariance in pattern recognitionâĂŤtangent distance and tangent propagation". In *Proceedings of Neural networks: tricks of the trade*, pages 239–274. Springer, 1998.

[9] Kamalika Chaudhuri and Sanjoy Dasgupta. "Rates of Convergence for Nearest Neighbor Classification". In *Proceedings of Advances in Neural Information Processing Systems*, pages 3437–3445, 2014.

[10] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Ruili Wang. "Efficient Knn Classification with Different Numbers of Nearest Neighbors". *IEEE transactions on neural networks and learning systems*, 29(5):1774–1785, 2017.

[11] Isabelle Guyon and André Elisseeff. "An Introduction to Variable and Feature Selection". *Journal of machine learning research*, 3(3):1157–1182, 2003.

[12] Karol Draszawka, Julian Szymański, and Francesco Guerra. "Improving css-KNN Classification Performance by Shifts in Training Data". In *Proceedings of International Conference on Semantic Keyword-Based Search on Structured Data Sources*, pages 51–63. Springer, 2015.

[13] Alberto Palacios Pawlovsky and Mai Nagahashi. "A Method to Select a Good Setting for the KNN Algorithm When Using It for Breast Cancer Prognosis". In *Proceedings of IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 189–192. IEEE, 2014.

[14] Kansheng Shi, Lemin Li, Haitao Liu, Jie He, Naitong Zhang, and Wentao Song. "An Improved KNN Text Classification Algorithm Based on Density". In *Proceedings of 2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, pages 113–117. IEEE, 2011.

[15] Neamat El Gayar, Friedhelm Schwenker, and Günther Palm. "A study of the robustness of KNN classifiers trained using soft labels". In *Proceedings of IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 67–80. Springer, 2006.

[16] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. "Adversarial Attacks on Neural Networks for Graph Data". In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2847–2856. ACM, 2018.

[17] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. "Intriguing properties of neural networks". *arXiv preprint arXiv:1312.6199*, 2013.

[18] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. "A limited memory algorithm for bound constrained optimization". *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[19] Pedro Tabacof and Eduardo Valle. "Exploring the Space of Adversarial Images". In *Proceedings of 2016 International Joint Conference on Neural Networks*, pages 426–433. IEEE, 2016.

[20] Fabio Carrara, Fabrizio Falchi, Roberto Caldelli, Giuseppe Amato, Roberta Fumarola, and Rudy Becarelli. "Detecting Adversarial Example Attacks to Deep Neural Networks". In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, pages 38:1–38:7. ACM, 2017.

[21] Zhenguo Zhang, Yanlong Wen, Ying Zhang, and Xiaojie Yuan. "Nearest Subspace with Discriminative Regularization for Time Series Classification". In *Proceedings of International Conference on Database Systems for Advanced Applications*, pages 583–599. Springer, 2018.

[22] Hongchang Gao and Heng Huang. "Self-paced Network Embedding". In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1406–1415. ACM, 2018.

[23] Nikos Karampatziakis and Paul Mineiro. "Discriminative Features via Generalized Eigenvectors". In *Proceedings of the 31th International Conference on Machine Learning*, pages 494–502, 2014.

[24] Sadiq Sani, Nirmalie Wiratunga, and Stewart Massie. "Learning Deep Features for KNN-based Human Activity Recognition". In *Proceedings of the ICCBR Workshops: Case-Based Reasoning and Deep Learning Workshop*, pages 95–103. ICCBR (Organisers), 2017.

[25] Vijay Chandrasekhar, Jie Lin, Olivier Morere, Hanlin Goh, and Antoine Veillard. "A Practical Guide to CNNs and Fisher Vectors for Image Instance Retrieval". *Signal Processing*, 128(11):426–439, 2016.

[26] Masashi Sugiyama. "Dimensionality Reduction of Multimodal Labeled Data by Local Fisher Discriminant Analysis". *Journal of machine learning research*, 8(5):1027–1061, 2007.

[27] Yiguang Liu, Shuzhi Sam Ge, Chunguang Li, and Zhisheng You. "k-NS: A Classifier by the Distance to the Nearest Subspace". *IEEE Transactions on Neural Networks*, 22(8):1256–1268, 2011.

[28] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. "Overfeat: Integrated recognition, localization and detection using convolutional networks". *arXiv preprint arXiv:1312.6229*, 2013.

YI REN received the B.E. degree in computer science and technology from Sichuan University, Chengdu, China, in 2017. He is currently pursuing the M.E. degree in computer science with the Huazhong University of Science and Technology. His current research interests include big data and machine learning.

XIA XIE is an associate professor at Huazhong University of Science and Technology (HUST) in China. She received her Ph.D. in computer architecture from HUST in 2006. Her research interests include data mining and big data. Contact her at shelicy@hust.edu.cn.

HAI JIN is a Cheung Kung Scholars Chair Professor of computer science and engineering at Huazhong University of Science and Technology (HUST) in China. He is now Dean of the School of Computer Science and Technology at HUST. Jin received his PhD in computer engineering from HUST in 1994. In 1996, he was awarded a German Academic Exchange Service fellowship to visit the Technical University of Chemnitz in Germany. Jin worked at The University of Hong Kong between 1998 and 2000, and as a visiting scholar at the University of Southern California between 1999 and 2000. He was awarded Excellent Youth Award from the National Science Foundation of China in 2001. Jin is the chief scientist of ChinaGrid, the largest grid computing project in China, and the chief scientists of National 973 Basic Research Program Project of Virtualization Technology of Computing System, and Cloud Security.

Jin is an IEEE Fellow and a member of the ACM. He has co-authored 15 books and published over 500 research papers. His research interests include computer architecture, virtualization technology, cluster computing and cloud computing, peer-to-peer computing, network storage, and network security.

Jin is the steering committee chair of International Conference on Green, Pervasive and Cloud Computing (GPC), Asia-Pacific Services Computing Conference (APSCC), International Conference on Frontier of Computer Science and Technology (FCST), and Annual ChinaGrid Conference. Jin is a member of the steering committee of the IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid), the IFIP International Conference on Network and Parallel Computing (NPC), and the International Conference on Grid and Cooperative Computing (GCC), International Conference on Autonomic and Trusted Computing (ATC), International Conference on Ubiquitous Intelligence and Computing (UIC).

HANHUA CHEN is a professor at Huazhong University of Science and Technology (HUST) in China. He received the Ph.D. in computer architecture from HUST in 2010. His research interests mainly include distributed systems and Big Data processing.