Sensitive Data Comparison Algorithm Based Spatio-temporal Label Distribution Fusion

Pengfei Yu*, Congcong Shi

State Grid Smart Grid Research Institute Co., Ltd. State Grid Key Laboratory of Information & Network Security, Nanjing China

Abstract

While the IoT cooperates with big data to deeply reconstruct all walks of life, it also poses more severe challenges to data security. Accurate identification of sensitive data is a prerequisite for data security. Compared with traditional machine learning algorithms, deep learning algorithms show great functionality and flexibility in large-scale data processing. However, the existing deep learning-based sensitive data identification methods focus on the mining of a single content feature, ignoring contextual information, and the identification accuracy of sensitive data with insignificant content features is not high.Therefore, this paper proposes a sensitive data comparison algorithm based on spatiotemporal label distribution fusion. The algorithm can simultaneously model the spatial and temporal patterns of the data flow, mine the spatial and temporal labels, and identify the type of data through a comprehensive judgment strategy. It solves the problem of identifying sensitive data with insignificant content characteristics. Finally, the algorithm is independently repeated experiments on multiple data sets and compared with multiple algorithms. The results show that the Best F-score and NAB score of this model are significantly better than other algorithms, which are 0.812 and 69.2, respectively. The algorithm proposed in this paper can more accurately identify sensitive data.

Keywords

label distribution; sensitive data; time stamp; space label

1. Introduction

Markedness ambiguity is a hot research direction in the field of machine learning. In the existing machine learning paradigm, there are mainly two data labeling methods: (1) assigning a label to an example; (2) An example assigns multiple tags. Single-Label Learning (SLL) assumes that all the examples in the training set are labeled in the first way, while Multi-Label Learning (MLL) [1] allows the training examples to be labeled in the second way. Therefore, multi-label learning can deal with the ambiguity that an example belongs to multiple categories. Whether it is single-label learning or multi-label learning, it aims to answer an essential question, that is, "which labels can describe this example?" . However, none of them directly answered the relative importance of each marker to this example.

For many problems in the real world, the importance of different markers is often different. For example, a natural scene image [2] is marked with multiple markers such as "sky", "water", "forest" and "clouds", but these markers describe the image in different degrees; In facial emotion analysis [3], people's facial expressions are often the result of a mixture of many basic emotions (such as happiness, sadness, surprise, anger, disgust and fear), and these basic emotions often express different intensities in a specific expression, thus presenting complicated emotions. There are many similar examples, because once an example is related to multiple markers at the same time, these markers are generally not all equally important to the example, but are more likely to have the priority. For applications similar to the above examples, a natural method is to assign a real number d_x^y to each possible mark y for an example x, indicating the degree to which y describes x. Without losing generality, suppose $d_x^y \in [0,1]$, and

AHPCAI2022@2nd International Conference on Algorithms, High Performance Computing and Artificial Intelligence
EMAIL: * 86503739@qq.com (Pengfei Yu), sshicongcong@geiri.sgcc.com.cn (Congcong Shi)

further suppose that the tag set is a complete set, that is, all the tags in the set can completely describe an example, so $\sum_y d_y^x = 1$. d_x^y that meets the above two conditions is called the description degree of y to x. For an example, the descriptions of all labels constitute a data structure similar to probability distribution, so it is called label distribution, and the process of learning on the data set labeled with label distribution is called Label Distribution Learning (LDL) [4].

Distributed label learning can be described as follows:

Let X=Rq represent the feature space of the example, and $Y=\{y_1, y_2, ..., y_c\}$ represent the marker space. Given a training set $S=\{ (x_1,d_1), (x_2, d_2), ..., (x_n,d_n)\}$, the goal of label distribution learning is to learn a conditional probability quality function p (y|x) from S, where $x \in X$ and $y \in Y$.

Assume that the parametric model of p (y|x) is expressed as p (y|x; θ), where θ is the parameter vector. Given the training set S, the goal of label distribution learning is to find a θ, so that given the example xi, p (y|x; θ) can generate a marker distribution as similar as possible to the real marker distribution di of xi.

2. Related Technology

2.1 Variational Auto-Encoder

Variational Auto-Encoder (VAE) is a generative model proposed by Kingma and Welling in "Auto-Encoding Variational Bayes" in 2014 [5]. Its network structure is consistent with AE, which consists of encoder and decoder. A known encoder can encode raw data into a low-dimensional vector, and we call this known initial vector a latent vector. The AE algorithm achieves the purpose of reproducing the input to the best of its ability, but it cannot generate any unknown data because it cannot generate reasonable latent variables at will. To solve this problem, the VAE constrains the encoder to produce latent variables that follow a unit Gaussian distribution.

The biggest difference between VAE and AE is that the AE middle layer outputs the specific values of the hidden variables, while the VAE middle layer outputs the specific distribution of the hidden variables. Unlike AE, which produces real-valued vectors, VAE's encoder produces two vectors: one for the mean and one for the standard deviation. This way, the model can take additional samples from this distribution and feed it into the decoder.

It should be noted that the error of the model is not only the reconstruction error at this time, VAE needs to balance the accuracy of the reconstructed data and the fit of the unit Gaussian distribution, so the loss function is the sum of two aspects: on the one hand, and Like AE, the output and the input are used for comparison, that is, the reconstruction error, which is generally measured by Kullback Leibler Divergence (KLD).

The VAE constraint on the Gaussian distribution of the decoder variable, in addition to enabling it to generate random latent variables, also greatly improves the ability of the network to generate pictures. For example, assuming that each real number in the interval [0, 10] corresponds to an object name, the interval can represent an infinite number of object names. For example, 7.01 corresponds to apples, and 7.02 corresponds to bananas. When data 7.01 is received, it is known that it represents Apple. Considering that real-world data contains a certain amount of Gaussian noise, when the received data is 7.01, the original value may be any number between [6.5~7.5], such as 7.02 (banana). Therefore, the greater the variance of a given data, the less usable information this vector of averages will carry. Similarly, in VAE, the more efficient the encoding, the closer the standard deviation vector is to the unit standard deviation of the standard Gaussian distribution. This constraint forces the encoder to be more efficient and able to generate informative latent variables. This in turn improves the performance of generating images.

2.2 Hierarchical Temporal Memory

Hierarchical Temporal Memory (HTM), also known as cortical learning, is a new generation of artificial intelligence algorithms published by Numenta, and has now launched the corresponding Python platform and visual recognition software toolbox. HTM originated from the memory-prediction framework proposed by Jeff Hawkins in his book "On Intelligence". The framework has a bionic

hierarchical structure, which can be modeled by memory patterns and sequences, and information between levels is transmitted up and down. HTM is designed to simulate how the neocortex works, turning complex problems into pattern matching and prediction. True to its name, this algorithm differs from ordinary neural network algorithms in many ways. HTM emphasizes the layering of "neurons". Hierarchy, Invariant Representations of Spatial Patterns and Temporal Patterns of information, and Sequence Memory are the three core points of HTM.

The fundamental difference between HTM and neural network algorithms is like the difference between general circuits and gate circuits. Connecting the simulated "neurons" according to the structure of the neocortex will produce a completely different effect from the general neural network. The general neural network pays attention to feedforward, while the HTM algorithm pays more attention to the two-way communication of information, which is also the reason why neuroanatomy found that the number of feedback synapses is no less than that of feedforward. And feedback doesn't get most people's attention.

In addition, most of the traditional artificial intelligence algorithms are designed for specific task objectives, while the HTM algorithm focuses on transforming the problem into a pattern matching and prediction problem before solving it, making the "unified theory" of artificial intelligence possible. HTM algorithms are based on a lot of anatomy and neuroscience. The HTM algorithm believes that the new cerebral cortex is an indispensable and necessary condition for human intelligence, and it is responsible for high-level brain activities. Our brains work by matching the various patterns we receive with those in memory, predicting and reacting to the information we will receive in the next moment, and so on. This is the manifestation of its timeliness (Temporal).

3. Sensitive Data Comparison Model

3.1 Model Overview

Fig. 1 is the block diagram of sensitive data identification system based on spatio-temporal tag distribution fusion. The system is divided into two parts: hierarchical real-time memory time tag extraction stage (top) and variational self-encoder space tag extraction stage (bottom). The former is mainly responsible for mining time tags in time series, while the latter is mainly responsible for mining space tags in time series.

As mentioned above, the sequential memory algorithm of hierarchical real-time memory further expands the original input information, and allows the algorithm to make a more accurate prediction and identification of the next sensitive data in the context of understanding the current input. In this paper, the type identification of sensitive data is transformed into a binary classification problem, and the distribution of labels is represented by the distribution probability L_t, and a threshold is set. When $L_t \geq 1-\gamma$, that is, $1- L_t \leq \gamma$, HTM considers that the label distribution characteristics of this sensitive data belong to Type II; otherwise, the label distribution characteristic of the sensitive data is Type I. VAE can accurately grasp the general pattern of current data and reconstruct its label distribution with high probability, but can't reconstruct outliers (abnormal label distribution) well. Label distribution is characterized by reconstruction probability Pr, and the threshold is also set. When $P_r \leqslant \eta$, VAE thinks that the label distribution characteristics of this data belong to the Type II of spatial label distribution, and vice versa is Type I on the spatial label distribution.

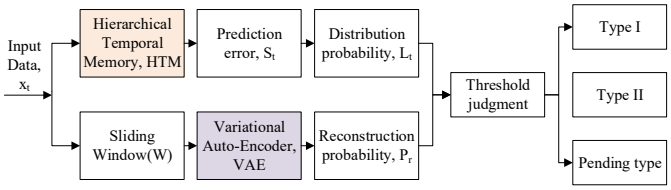

Figure 1. System block diagram of sensitive data identification scheme based on spatiotemporal label distribution fusion

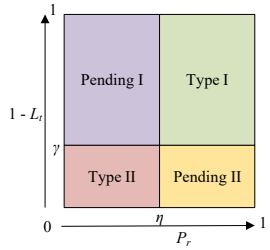

Figure 2. Sensitive data comparison scheme determination strategy based on spatiotemporal label distribution fusion

3.2 Label Distribution Decision Strategy

The decision strategy is shown in *Fig. 2*. The abscissa is reconstruction probability P_r, the ordinate is $1\text{-}L_t$, and the horizontal and ordinate ranges from 0 to 1. When the evaluation results of the two algorithms are consistent, the evaluation result of either party is the identification result of the sensitive data: the sensitive data both judged as Type II are classified as Type II sensitive data (red area "Type II"), and the sensitive data both judged as Type I are classified as Type I sensitive data (green area "Type I"). When the two algorithms determine the contradiction, the sensitive data will enter the pending state, and the system will further determine it: (1) If HTM determines that its time label distribution is the first type, but VAE determines that its space label distribution is the second type, the point will enter the pending area in the upper left corner (purple area "Pending I"). Reduce the value of sliding window W and observe P_r. If P_r continues to increase until the point enters the first-class area, the system determines that the sensitive data is the first-class sensitive data. On the contrary, if P_r is always less than η, it is determined that the sensitive data is the second sensitive data. (2) If HTM determines that its time label distribution is the second category, but VAE determines that its space label distribution is the first category, then the point enters the undetermined area in the lower right corner (yellow area " Pending II"). Increase the value of sliding window W and observe $1\text{-}L_t$. If $1\text{-}L_t$ continues to increase until the point enters the first-class area, the system will determine the sensitive data as the first-class sensitive data. On the contrary, if $1\text{-}L_t$ is always less than γ, it is determined that the sensitive data is the second sensitive data.

The significance of the above operation is: for the point to be Pending I, the system knows the distribution of the time label but not its spatial label. After reducing the window value, if VAE admits that it belongs to the first class in this small range, it will be judged as the first class sensitive data, thus avoiding the false alarm caused by VAE's inability to respond to the concept drift phenomenon in time. For the point to be Pending Ⅱ, the system knows its spatial label distribution but not its time label distribution. After increasing the window value, if HTM learns this time pattern in a wider range, the prediction error of this point will be reduced, and the corresponding label distribution in the threshold range will change this point into the first sensitive data, thus avoiding the false alarm caused by HTM's inability to learn the complete distribution pattern due to its small window value. At the same time, the multi-terminal control of time-space distribution label fusion makes the system solve the problem of "whether the value of the sample point is right" and "whether the sample point should come at this time", which greatly reduces the false alarm rate and false alarm rate, and makes the model more detailed and three-dimensional.

4. Experiments and Results

For the test of sensitive data comparison, this paper selects 12 typical IoT data sets (three data sets for each type) from an open source real-world data set [5, 6], including CPU utilization, intelligent industrial system temperature sensor, Electro Cardio Gram (ECG) and HTTP service response delay.

Fig. 3 shows the comprehensive detection results of sensitive data comparison scheme based on spatio-temporal distribution tag fusion. *Fig. 3* (a), 3 (b), 3 (c) and 3 (d) correspond to data sets A, B, C and D, respectively. A represents the data set of CPU utilization, B represents the data set of temperature sensor,

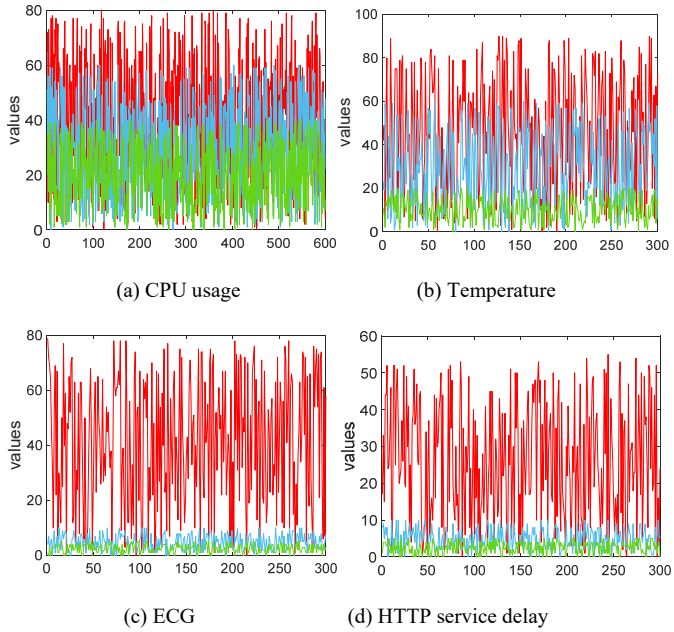

Figure 3. Comprehensive detection results obtained by the sensitive data comparison scheme

TABLE 1. MODEL SCORES OBTAINED BY MULTIPLE ALGORITHMS USING NAB SCORING MECHANISM

Algorithm	Standard Score	Reward Low FP	Reward Low FN
perfect	100	100	100
ST	69.5	62.1	72.4
HTM only	31.2	30.8	34.9
VAE only	43.1	39.9	46.7
LSTM-AE	50.8	46.4	55.3
LSTM-VAE	30.3	25.6	33.8
random	8.5	3.1	11.5
null	0	0	0

C represents the data set of ECG, and D represents the corresponding delay of HTTP service. The proportion of sensitive data markers in the original data set is 10%. In the figure, the abscissa is the time axis, the blue line is the original data, the red line is the spatial label result obtained from the encoder based on variation, and the green line is the time label result obtained from hierarchical real-time memory. It can be seen from the figure that VAE can stably and successfully detect the corresponding spatial tags for obvious abnormal changes, such as sudden peaks and valleys. Fortunately, HTM can sensitively detect subtle changes and successfully detect the time stamp.

Table 1 shows the model scores obtained by various algorithms using NAB scoring mechanism. The result of "null" detector is 0, the result of "perfect" detector is 100, and the result of "random" detector is the average of a series of random seeds. In addition, the algorithms involved in the comparison are as follows: Spatio-temporal sensitive data comparison schemes ST(Spatio-Temporal), HTM only, VAE only, LSTM-AE [7-9] which uses long short term memory, and LSTM-VAE [10-12] which uses long short term memory. Standard Score is the score obtained by NAB standardized calculation (AFP = AFN) after all data sets are tested, Reward Low FP is the score obtained by testing and calculating FP preference NAB (AFP > AFN) on D1~D3 data sets, Reward Low FN is the score obtained by testing and calculating FN preference NAB (AFP < AFN) on C1~C3 data sets.

Generally speaking, ST, the spatio-temporal fusion detection scheme in this paper, has the highest score, and the self-encoder with LSTM is ranked second, followed by the spatial detection algorithm that only uses VAE. HTM only and LSTM-VAE have the lowest scores. The results show that the spatio-temporal fusion measurement model proposed in this paper has excellent detection performance in anomaly detection tasks, and it also proves the effectiveness of using LSTM as encoder and decoder to fit time series.

5. Conclusions

To solve the problem that the existing machine learning algorithms have low recognition accuracy for sensitive data with insignificant features, this paper studies the sensitive data label generation technology and sensitive data label comparison technology, then proposes a sensitive data comparison model based on spatio-temporal label distribution fusion, and formulates a label distribution comprehensive judgment strategy for this model, which divides the sample data into the first sensitive data, the second sensitive data and the pending data. For the sample points in the undetermined area, dynamically debug by changing the window size, and make two rounds of judgment to get the final detection result. Experimental results show that the Best F-score and NAB score of this model are obviously due to other algorithms, which are 0.812 and 69.2 respectively.

However, in the experiment of this paper, the classified experimental data is only part of the training set of the original data set. Next, we can try to test the classification of the complete data set, which is of great significance to investigate the generalization ability and multi-classification ability of the model (up to 39 abnormal and 1 normal, totaling 40 kinds).

6. Acknowledgment

This paper is supported by the science and technology project of State Grid Corporation of China: " Research and Application of Scenario-Driven Data Dynamic Authorization and Compliance Control Key Technology " (Grand No. 5700-202058481A-0-0-00).

7. References

[1] L. Yang, Y. Song, S. Gao, A. Hu and B. Xiao, "Griffin: Real-Time Network Intrusion Detection System via Ensemble of Autoencoder in SDN," in IEEE Transactions on Network and Service Management, vol. 19, no. 3, pp. 2269-2281, Sept. 2022, doi: 10.1109/TNSM.2022.3175710.

[2] Q. Chen, Y. Song, B. Jennings, F. Zhang, B. Xiao and S. Gao, "IoT-ID: Robust IoT Device Identification Based on Feature Drift Adaptation," 2021 IEEE Global Communications Conference (GLOBECOM), 2021, pp. 1-6, doi: 10.1109/GLOBECOM46510.2021.9685693.

[3] Y. Song, B. Chen, T. Wu, T. Zheng, H. Chen and J. Wang, "Enhancing Packet-Level Wi-Fi Device Authentication Protocol Leveraging Channel State Information," Wireless Communications and Mobile Computing, vol. 2021.

[4] Y. Song, Y. Geng, J. Wang, S. Gao and W. Shi, "Permission Sensitivity-Based Malicious Application Detection for Android," Security and Communication Networks, vol. 2021.

[5] B. Chen, Y. Song, T. Wu, H. Chen, J. Wang and T. Li, “Enhancing Wi-Fi Device Authentication Protocol Leveraging Channel State Information,” 2021 International Conference on Mobile Multimedia Communications. Springer, Cham, 2021: 33-46.

[6] B. Chen, Y. Song, Z. Zhu, S. Gao, J. Wang and A. Hu, "Authenticating Mobile Wireless Device Through Per-packet Channel State Information," 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2021, pp. 78-84, doi: 10.1109/DSN-W52860.2021.00024.

[7] T. Wu, Y. Song, F. Zhang, S. Gao and B. Chen, "My Site Knows Where You Are: A Novel Browser Fingerprint to Track User Position," ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500556.

[8] X. Ma, Y. Song, Z. Wang, S. Gao, B. Xiao and A. Hu, "You Can Hear But You Cannot Record: Privacy Protection by Jamming Audio Recording," ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500456.

[9] R. Song, Y. Song, S. Gao, B. Xiao and A. Hu, "I Know What You Type: Leaking User Privacy via Novel Frequency-Based Side-Channel Attacks," 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1-6, doi: 10.1109/GLOCOM.2018.8647385.

[10] R. Song, Y. Song, Q. Dong, A. Hu and S. Gao, "WebLogger: Stealing your personal PINs via mobile web application," 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), 2017, pp. 1-6, doi: 10.1109/WCSP.2017.8171036.

[11] C. Shi, R. Song, X. Qi, Y. Song, B. Xiao and S. Lu, "ClickGuard: Exposing Hidden Click Fraud via Mobile Sensor Side-channel Analysis," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp.
[12] Y. Song, S. Gao, A. Hu and B. Xiao, "Novel attacks in OSPF networks to poison routing table," 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1-6.