When a Family of Iris Flower is Normal, Then are Others Abnormal?

Akira Imada

Brest State Technical University
Moskowskaja 267, 224017 Brest, Republic of Belarus
akira@bsty.by

Abstract. This article is not a report of a success but rather a challenge to those who claimed to have successfully designed a network intrusion detection system by means of a machine learning technique using artificial dataset to train and to test the system.

1 Introduction

Those highly qualified hackers who provide security services to companies during the daytime and then go home at night to conduct totally illegal hacking are the ones who are the most dangerous. – by Enis Senerdem from Turkish Daily News on 29 March 2006.

Designing a network intrusion detection system is one of the hottest issues in our computer network society. Designing such a system with so-called a soft-computing seems to become a sort of fashion these days. When we want to try one of these approaches, we usually need a dataset to *train* the system (if we employ supervised learning), and to *test* the system afterwards. Sometimes an artificial dataset is employed for the purpose. In fact we could find a lot of such datasets in public domain. *Spearman's iris-flower* dataset is one of those examples and probably one of the most frequently used ones. We doubt, however, these dataset cannot necessarily reflect data of computer network connections in real world. We usually don't know what does a coming intrusion look like until it has completed the illegal connection when actually it is too late. In this paper, we give it a consideration on how an intrusion can be detected by an intelligent way, if any.

2 Iris flower dataset

Iris flower dataset[1] is made up of 150 samples consists of three species of iris flower, that is, *setosa, versicolor* and *virginica*. Each of these three families includes 50 samples. Each sample is a four-dimensional vector representing four

[1] This can be obtained from University of California Urvine Machine Learning Repository. ftp://ics.uci.edu: pub/machine-learning-databases.

attributes of the iris flower, that is, *sepal-length, sepal-width, petal-length*, and *petal-width*.

As is often mentioned, this iris flower dataset is perhaps one of the most often used datasets in pattern recognition/classification, machine learning, data mining, etc. As a matter of fact, there have been a fair amount of studies in which this iris flower dataset is employed as a dataset to train and to test the system.

Quite naturally, all of these papers report their success in designing a system. Let us take an example from among many others. Castellano et al. [1] assumed one family of this iris flower to be normal whilst the other two to be abnormal. The whole dataset was divided into 10 parts each of which has 15 samples and are uniformly drawn from the three classes. The system is trained by the remaining 135 samples. The originally picked up 15 samples are used to test the results. After this 10-fold cross validation, the authors concluded that the system shows the invasion detection rate is 96% while the false alarm rate is 0.6%.

In reality, however, it is not so simple. Just imagine that a hacker always tries to explore an unlearned region to invade the network. Or worse, we should know hacker is a person who is very good at locating attack just behind a normal.

3 Challenges

In this section we challenge the reader with five problems.

3.1 Standard settings

First of all, let us formalize the standard way of using iris dataset for designing a network intrusion detection system.

Problem 1 (One is normal while the others abnormal) *Assuming one out of three families of iris flower to represent illegal transactions while the remaining two families represent legal ones. Is it possible then to simulate a system for network intrusion detection by using part of this dataset to train and remaining data to test the system?*

Let us now take a look at how those iris data are distributed in the whole search space. We tried a Sammon mapping to see those data in a fictitious 2-dimensional space.

Sammon mapping maps a set of points in a high-dimensional space to the 2-dimensional space with the distance relation being preserved as much as possible, or equivalently, the distances in the n-dimensional space are approximated by distances in the 2-dimensional distance with a minimal error.

One of the results of Sammon mapping of iris flower dataset is shown in Figure 1. Just a brief look at the figure reveals us that there remains an enormously big region of unlearned.

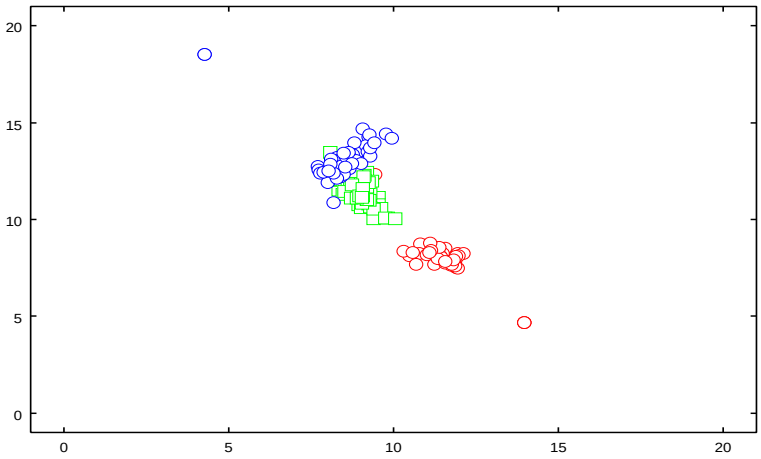

Fig. 1. A 2-D visualization of iris flower data by Sammon Mapping

3.2 Mutation

We now introduce a mutation which is to be applied to the data in order to avoid for the data to be deterministic. Real data are never expected to be deterministic. Each of the iris dataset is given as the form of

$$(x_1, x_2, x_3, x_4).$$

Then with a small probability called mutation-rate we modify records by

$$x_i^{\text{new}} = x_i^{\text{old}} + r\sigma_i \qquad (i = 1, 2, 3, 4)$$

where σ_i is standard deviation of 150 values of x_i, and r is a small random number anewed each time when mutation occurs. Then our problem might turn into as follows.

Problem 2 (Mutants of normal and attack) *(1) Train the system, in the same way as Problem 1, using three families of iris flower as normal and abnormal. (2) All points of all the three families of iris flower are given mutation. Call them mutants-of-normal and mutants-of-attack. (3) Then the system trained in step 1 can recognize these mutants properly as normal or attack?*

Does the system which is trained by a given deterministic set of normal and attack samples still immune to those mutants? It would be really hard to believe.

3.3 Detection of randomly-located but already-known attacks

Let us now be more demanding. We explore the same universe of iris flower data set whilst we take no notice of the iris families for a while. That is, we assume the universe is the 4-dimensional Euclidean space all of whose coordinates lie between 0 and 1.

Problem 3 (Randomly located normals and attacks) *(1) Create 50 normal samples and 100 abnormal samples all at random. (2) Then train the system using all of these samples of normal and attack. (3) Test the system again with these samples of normal and attack.*

This is reminiscent of the experiment by Ayara et al. [2] who created 8-bit binary random strings, as a set of training samples of normal patterns assuming other comparable amount of 8-bit binary patterns as abnormal with their pattern-recognition results being very successful.

Yes, still this is not so difficult because the data used were fixed ones. How about, however, if we are interested in mutants such as in the Problem-2 that are not *a priori* known fixed data any more?

3.4 Don't we expect the result before making an experiment?

We have to be careful, because we sometimes tend to *unconsciously* pick up only a set of data that will be suitable to draw our *a priori* expected conclusion, if not *intentionally* at all.

In the way that just a *harmless dummy pill* or even *powder-from-sugar* sometimes has a same effect as, or more efficient than, a medicine under developing enough to cure a disease for a group of innocent volunteers. Why don't we try the following question.

Problem 4 (A placebo experiment) *(1) Create a simple device which randomly returns either one of normal or attack regardless of the input. (2) Prepare a test dataset including enough amount of records uniformly from normal and attack. (3) Compare the performances of the detector you designed with the random-reply-machine created in step 1, feeding the same dataset prepared in step 2.*

3.5 Abnormal & dummy to a system trained with normal alone

Though we have not remarked so far, there remains further difficult issue, that is, "How the system can learn only from normal data to detect attack?" We usually have enormous amount of normal data but we have no information about coming attacks untill it's too late.

Gomez et al. [3] claimed, *"A new technique for generating a set of fuzzy rules can characterize the abnormal space using only normal samples."* [2] It would be terrific if the report was really successful, but we are fishy more or less.

This issue is something like we require a wine-taster to recognize bootleg champagne by only providing him/her a plenty of real champagne to learn.[3]

Though this *training-only-with-normal* is our ultimate goal, but not so simple to be realized. To study how this is difficult, why not try the following?

Problem 5 (Can a sommelier be trained without bootlegs?) *(1) Assume one family of iris as normal while the other two abnormal. (2) Furthermore, randomly create an attack dataset. Call them dummy attacks. (3) Train your intrusion detection system only with the normal set. (4) Then, try two tests, one with only abnormal, and the other with only dummy, avoiding any a priori prediction.*

4 Experiment

Each of the above 5 different challenges is now being tried by a *decision tree algorithm* — C4.5. However, our goal will not be to show successes but rather be opposit.

Also we will try to specifies the samples so that successful detection rate become as low as possible, and false alarm rate becomes as high as possible. Not performed yet though, we also plan to create abnormal samples by using a co-evolution of pledetor-and-prey type, for the purpose.

5 Concluding Remarks

We feel sorry that we have described the above not so optimistically. However, we should be careful to draw a conclusion from our experiment or simulation. We sometimes tend to overestimate our results so that we like it. Needles to say, however, this article is not to deny the possibility, but instead we hope to be a challenge for real new innovative approaches to be emerged.

[2] Not the orignal expression in their paper, but paraphrased by the author of this article.

[3] Or, in an opposite way. I usually enjoy Georgian sparkling wine like once a week, but still a real champagne would be able to pretend to be a Georgian one to me."

References

1. G. Castellano and A. M. Fanelli(2000) "Fuzzy Inference and Rule Extraction using a Neural Network." Neural Network World Journal Vol. 3, pp. 361–371.
2. M. Ayara, J. Timmis, R. D. Lemos, L. N. D. Castro, and R. Duncan (2002) "Negative Selection: How to Generate Detectors." Proceedings of 1st International Conference on Artificial Immune Systems pp. 89–98.
3. J. Gomez, F. Gonzalez, and D. Dasgupta (2003) *"An Immuno-Fuzzy Approach to Anomaly Detection."* Proceedings of IEEE International Conference on Fuzzy Systems, Vol. 2, pp. 1219–1224.