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

Akira Imada

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

**Abstract.** This article is not a report of success but rather a challenge to those who claim 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 try one of these approaches, we need a dataset to *train* the system if our strategy is a supervised learning, and also need it to *test* the system afterwards. Usually we employ an artificial dataset for the purpose. In fact we could find a lot of such datasets in public domain. *Spearman's iris-flower* dataset[1] is an example and probably one of the most frequently used ones. Yet another most frequently used is *KDD-cup-99* dataset.[2] We doubt, however, these datasets cannot necessarily reflect data of computer network connections in real world. We don't know what does a coming intrusion look like until it has completed the illegal connection when actually it is too late.

Previously we posed a doubt about using KDD-cup-99 dataset in designing an intrusion detection system [1]. In this paper, on the other hand, we concentrate on the iris flower dataset among others and give it a consideration on how an intrusion can be reflected by a family of iris flowers and how an intrusion can be detected by an intelligent way, if any.

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

2 See http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

## 2 What does iris flower dataset look like?

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

This iris flower dataset is perhaps one of the most often used datasets in research works concerning pattern recognition/classification, machine learning, data mining, etc. As such, there have been a fair amount of studies in which this iris flower dataset is employed as a dataset to train and to test an intrusion detection 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. [2] assumed one family of this iris flower to be abnormal whilst the other two to be normal. 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 their system with fuzzy neural network using Takagi-Sugeno model 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 it would be quite easy for a hacker 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.

### 2.1 A visualization by Sammon mapping

We now take a look at how those iris data are distributed in the whole search space. We tried a *Sammon mapping* to get a bird's eye view of 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 our results of Sammon mapping of iris flower dataset is shown in Fig. 1. Just a brief look at the figure reveals us that there remains an enormously big region of unlearned.

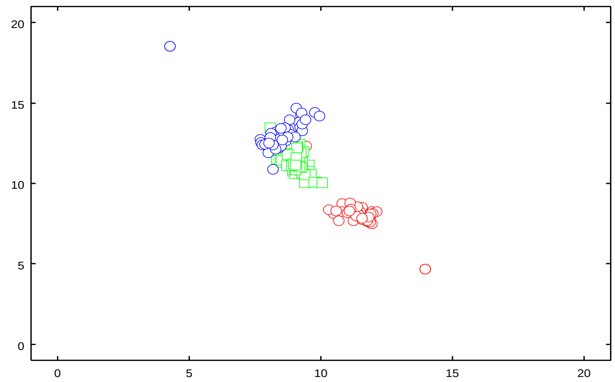


**Fig. 1.** A two dimensional visualization of the iris flower data by Sammon Mapping. The 150 data of three families, 50 each, are plotted.

## 3 Five challenges

In this section we challenge readers of this article with five problems.

**Standard settings.** We now start by formalizing the standard way of using iris dataset to design a network intrusion detection system.

**Standard Experiment (One is attack while the others are normal)**
*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 the remaining data to test the system?*

We have so many candidate techniques to design such a system with. Sabhnani et al. [3] might be a good summary in which we can overview such techniques applied to design intrusion detection system.[3] Those applied in their paper are (i) *Multi-layer Perceptron*, (ii) *Gaussian Classifier*, (iii) *K-mean Clustering*, (iv) *Nearest Cluster Algorithm*, (v) *Radial Basis Function*, (vi) *Leader Algorithm*, (vii) *Hypersphere Algorithm*, (viii) *Fuzzy Art Map*, and (ix) *C4.5 Decision Tree*. Our experiment here was made with a two-layer perceptron, as an example, whose implementation for the iris dataset is shown in Fig. 2.

First of all, in order to see how it works, we apply this two-layer perceptron to the iris dataset under a standard situation where one family assumed to be attack and the other two to be normal. We determined the configuration of synaptic

[3] Target in their paper was not the iris dataset but to the KDD-cup-99 dataset which is more demanding than the iris dataset.

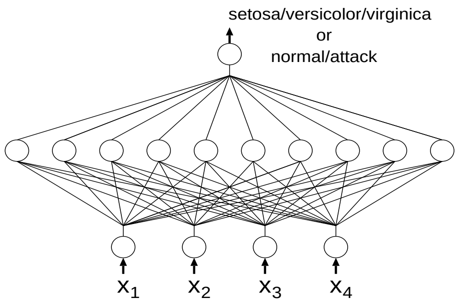


**Fig. 2.** A possible implementation by two layer perceptron to detect one family as attack while the other two as normal.

weight values of the perceptron by a Genetic Algorithm with chromosomes being made up of all the synaptic weights of the perceptron. One of the results of the evolution is shown in Fig. 3 which suggests that the task is quite easy.

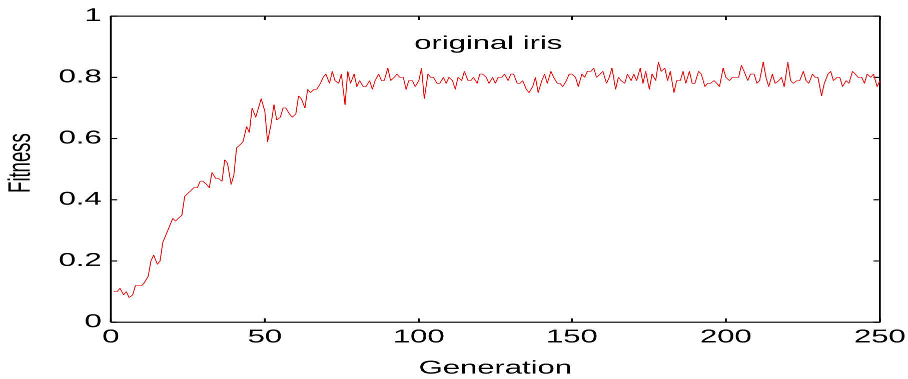


**Fig. 3.** A result of our experiment to evolve neural network attack detector under a standard setting.

### 3.1 What if both normals and attacks are not well clustered?

It is too simplified or too optimistic assuming both normal and attack data are well clustered such as the iris dataset. We now look at what will happen if both normal and attack data are distributed at random as an extreme case, such as those shown in Fig. 4.

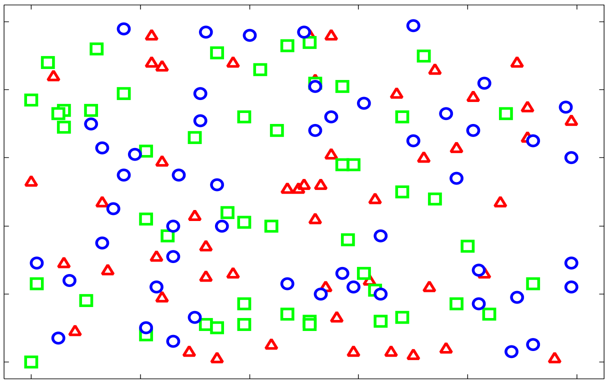

**Fig. 4.** An example of randomly distributed 100 normals and 50 attacks shown in a fictitious 2-dimensional search space.

Now our first challenge is:

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

This might be reminiscent of the experiment by Ayara et al. [4] who created, in the context of network intrusion detection, a set of 8-bit binary random strings as training samples of normal patterns assuming other comparable amount of 8-bit binary patterns as abnormal with their results being very successful in recognizing both normal and abnormal.

Yes, still this is not so difficult because the data used were fixed ones. How about, however, if we create a *mutant from a normal* each time as an attack, and as such, attacks are not *a priori* known fixed data any more?

### 3.2 Can a system recognize a mutant as mutant?

When we use the iris dataset as artificial dataset of normal and abnormal access to a computer network, our assumption is that attacks are apart enough from normal data. However this does not necessarily reflect a reality. Skillful hackers attack a network so that their transaction being as similar as normal, wishing network administrator wouldn't notice. We simulate this situation by mutating normal data and considering them as attacks.

**Mutation.** We now introduce a mutation which is to be applied to the data in order to avoid for the data to be deterministic since 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 $\sigma_i$ is standard deviation of 150 values of $x_i$, and $r$ is a small random number which is created anew each time when mutation occurs. Then our problem might turn into as follows.

**Challenge 2 (Mutant of normal as attack)** *(1) Create 100 normal samples all at random. (2) Then 50 out of those samples are mutated as attack. (3) Train with half of the normal and mutants behind them. (4) Test with the remaining half.*

The situation might be too demanding since we assume all normal samples are distributed at random. In order to reduce the requirement, let us go back to the iris dataset.

### 3.3 What if outlier hides behind normals?

It would be more realistic that normal data are well clustered in the whole universe of possible data, and the attack data are mutants of those normals as the previous subsection. A fictitious 2-dimensional plots are shown in Fig. 5.

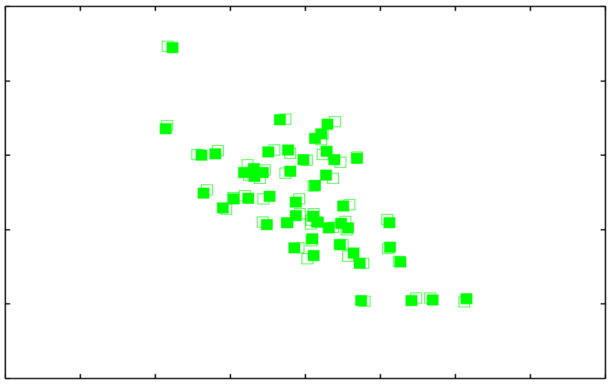

**Fig. 5.** An example of 50 attacks who try to hide behind the normals of the iris family which is one of the three shown in Fig. 1.

Then the challenge here is:

**Challenge 3 (Attacks who hide behind normals)** *(1) Assume one family of iris flower as normal (2) Then all of the family member are mutated as attack. (3) Train with half of normal and mutated attacks. (4) Test with the remaining half of normals and newly created mutants of attack.*

Is the system which was trained by a given deterministic attack samples still immune to attacks by new mutants from normals? It seemed hard to believe.

### 3.4 Previous attack data work as training sample?

Then how about an idea of not using attack data in training the system? Can we train a system only with normal samples? How the system can learn only from normal data to detect attacks? This is further difficult issue but actually is our must, since we usually have enormous amount of normal data while we have no information about coming attacks until it's too late.

In their articles, Gomez et al. [5] claimed, *"A new technique for generating a set of fuzzy rules can characterize the abnormal space using only normal samples."*[4] Gomez et al. designed the detector they call an *evolving-fuzzy-rule-detector* with what they call an *immuno-fuzzy approach.*

Let's be a little more specific and apply the method to the iris flower dataset.[5] First of all, we define five linguistic terms:

{*Low, Medium-low, Medium, Medium-high, High*}

where *Medium* and its both neighbors are specified using triangle membership functions while *Low* and *High* are using trapezoidal membership functions. Then an example of a set of rules is

---

*IF*
$x_1$ *is Law or Medium-law;*
$x_2$ *is High;*
$x_3$ *is Medium or Medium-high or High; and*
$x_4$ *is Medium-low or Medium.*

*THEN*
$\mathbf{x} = (x_1, x_2, x_3, x_4)$ *is Attack.*

---

Assuming we have $k$ such rules, Gomez et al. defined a *degree of membership of* $\mathbf{x}$ *to be an attack* as

$$\mu_{\text{attack}}(\mathbf{x}) = \max_{i=1,\ldots,k} \{\text{Eval}_{R_i}(\mathbf{x})\}$$

[4] This is not the original expression in their paper, but paraphrased by the author of this article.
[5] Gomez et al. used the KDD-cup-99 dataset instead of the iris flower dataset.

where $\mathrm{Eval}_{R_i}(\mathbf{x})$ is the fuzzy true value evaluated by the *IF*-part of the rule $R_i$, and thus $\mu_{\mathrm{attack}}(\mathbf{x})$ closer to zero means that $\mathbf{x}$ is more likely to be a normal and closer to one is more likely to be an attack.

Then our chromosome comprises four genes corresponding to $x_1, \cdots, x_4$, and each of those genes is 5-digit binary string corresponding to five fuzzy linguistic terms. For example, ((11001)(01001)(10000)(11100)) implies:

| |
|---|
| $x_1$ *is Low or Medium-Low or High* |
| $x_2$ *is Medium-Low or High* |
| $x_3$ *is Low* |
| $x_4$ *is Low or Medium-Low or Medium* |

where fitness is essentially the evaluation of *how few a rule covers a set of given N normal training samples.*[6]

Thus the authors claimed, *"It detects attacks with the detection rate 98.30 % and false alarm rate 2.0 %."* It would be terrific if the report was really successful, but we think it fishy more or less.

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

**Challenge 4 (Can a sommelier be trained without bootlegs?)**
*(1) Assume one family of iris as normal while the other two abnormal. (2) In addition, create an attack dataset at random and 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.*

This issue is something like we require a wine-taster to recognize a bootleg champagne or a not-champagne-but-sparkling-wine by only providing him/her a plenty of real champagne to learn with.[7] Can a sommelier be trained without bootlegs?

### 3.5 Is an intelligent decision better than random one?

Moreover, we have to be careful, because we sometimes have a tendency 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.

[6] A volume of the subspace represented by the rule is also taken into account.

[7] 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.

*Drugs to ... are no more effective than placebos for most patients ... found no significant difference than the placebos or at increased risk of death compared to placebo.*

This is from an article of a newspaper.[8] A *harmless dummy pill* or even *powder-from-sugar* sometimes has an 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?

**Challenge 5 (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.*

See Fig. 6.

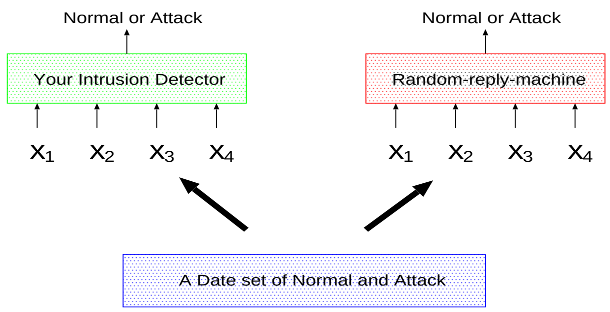


**Fig. 6.** Intelligently designed attack detector versus random reply machine.

## 4 A result — Still seems to be an easy task?

We tried to evolve a configuration of weights for the two-layer perceptron mentioned above with fitness evaluation being how the perceptron detects the samples of normal and attack data correctly. When we used (i) a set of random data as normal and attack such as those in Fig. 4, and (ii) one family of iris as normal and their mutants as attack, such as the ones in Fig. 5, the evolution never reached to a satisfactory level while, as already mentioned, it evolved fairly easy when we gave two families of iris as normal and the other family as attack such as the ones in Fig. 1. The fitness versus generation of these three cases are shown in Fig. 7.

[8] By Benedict Carey in the New York Times on 12 October 2006.

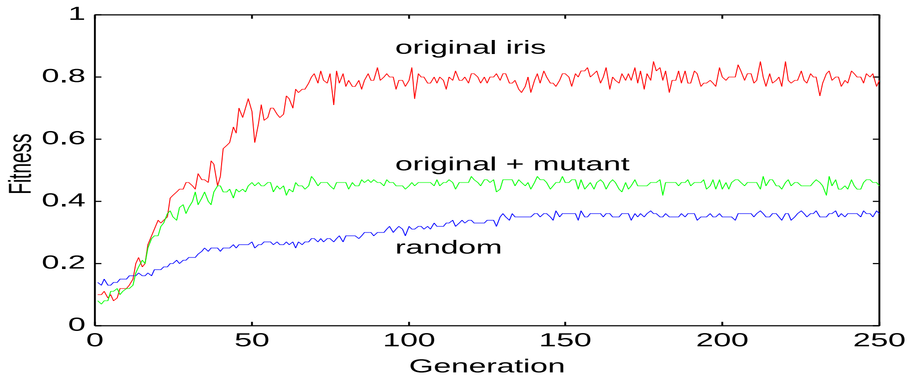


**Fig. 7.** The results of three of our experiments: under a standard setting (original iris), with attack hidden behind normal (original + mutant), and with random dataset (random).

## 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 it to be a challenge for real new innovative approaches to be emerged.

## References


1. A. Imada (2006) *"How Many Parachutists will be Needed to Find a Needle in a Pastoral? — Who is a lucky one?"* Proceedings of the International Conference on Neural Networks and Artificial Intelligence, pp. 53–60.
2. 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.
3. M. Sabhnani, and G. Serpen (2003) *"Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context."* Proceedings of the International Conference on Machine Learning: Models, Technologies and Applications, pp. 209–215.
4. 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.
5. 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.


This article was processed using the LaTeX macro package with LLNCS style