Application of a Bayesian Semi-supervised Learning Strategy to Network Intrusion Detection

Carlos A. Catania[1], Carlos García Garino[1], Facundo Bromberg[2]

[1] ITIC, Universidad Nacional de Cuyo, Mendoza, Argentina
{ccatania,cgarcia}@itu.uncu.edu.ar
[2] Dept. Sistemas de Información, FRM, UTN, Mendoza, Argentina
fbromberg@frm.utn.edu.ar

Abstract. Supervised learning classifiers have proved to be a viable solution in the network intrusion detection field. In practice, however, it is difficult to obtain the required labeled data for implementing these approaches. An alternative approach that avoids the need of labeled datasets consists of using classifiers following a semi-supervised strategy. These classifiers use in their learning process information from labeled and unlabeled datapoints. One of these semi-supervised approaches, originally applied to text classification, combines a naïve Bayes (NB) classifier with the *expectation maximization* (EM) algorithm. Despite some differences, network intrusion detection shares many of the characteristics of the document classification problem. It is extremely hard to obtain labeled data whereas there are plenty of unlabeled data easily accessible. This work aims to determinate the viability of applying semi-supervised techniques to network intrusion detection, with special focus on the combination of NB classifier and EM. A set of experiments conducted on the 1998 DARPA dataset show using EM with unlabeled data can provide significant benefits in classification performance, reducing the size of required labeled data by 90%.
Keywords: Intrusion Detection Systems - Semi-supervised Learning - Expectation Maximization.

1 Introduction

The use of supervised learning classifiers for network intrusion detection has been applied successfully in previous works [1,2,3]. As it is known, supervised classifiers require a dataset containing labeled traffic instances for the learning process. Unfortunately, in the case of network intrusion detection, obtaining such labeled datasets requires considerable human effort.
One possible solution is the use of classifiers which follows a semi-supervised learning strategy [4]. Algorithms following this strategy are able to learn classification models using information not only from labeled datasets but also from unlabeled ones.
A simple semi-supervised approach, usually applied to document classification problems [5], is introduced by Nigam et al. in [6]. The proposed algorithm is

based on the combination of EM [7] and a *NB* classifier. The algorithm first trains a classifier using the available labeled documents, and probabilistically labels the remaining unlabeled documents. It then trains a new classifier using the labels for all the documents, and iterates to convergence. In many document classification problems, only a reduced set of labeled documents are accessible whereas a big number of unlabeled documents are available. This situation makes document classification problem suitable for using a semi-supervised approach.

Despite some differences, network intrusion detection shares many of the characteristics of the document classification problem. The available labeled dataset are limited, whereas there are plenty of unlabeled data easily accessible. Therefore, it seems a semi-supervised approach like the proposed in [6] could provide benefits to the network intrusion detection problem.

In this work, an study is carried out in order to determinate the viability of applying semi-supervised techniques to network intrusion detection with special focus on the combination of EM and a *NB* classifier. A set of experiments are conducted on the 1998 DARPA dataset [8], a widely used dataset for testing network intrusion detection approaches.

The rest of this paper is organized as follows. Best known approaches for reducing labeling efforts are mentioned in section 2. Section 3 briefly describes the elements involved and special considerations required, when the proposed Bayesian semi-supervised strategy is applied to intrusion detection. The results of the evaluation of the proposed approach on the 1998 DARPA dataset are presented in section 4. Finally, in section 5, concluding remarks and future works are commented.

2 Background and Related works

A number of alternative approaches have been proposed in order to reduce or simply avoid the need of labeled datasets and the consequent human effort required.

A first approach consists of using unsupervised learning techniques. One of the major advantages of this approach is that it is suitable for handling unlabeled training data sets with not only normal traffic but also anomalies (i.e., attacks). Algorithms such as Support Vector Machines (SVM) [9] and clustering [10] were applied to the network intrusion detection field. Unfortunately, as was noticed by Eskin in [10], algorithms following the unsupervised strategies only works under the assumption that the number of normal traffic instances vastly outnumbers the number of anomalies. An assumption which not always holds.

A second approach uses semi-supervised learning techniques. Following the promising idea of learning from labeled and unlabeled traffic instances, some authors [11,12,13] have focus their work on this learning alternative applied to network intrusion detection.

Among the best known semi-supervised learning techniques [4], the combination of *NB* and the EM provides a good trade off between simplicity and performance.

Thus, it seems important to evaluate its performance on the network intrusion detection field.

3 A Bayesian Semi-supervised Strategy for Network Traffic Classification

Nigan et al. [6] proposed a semi-supervised learning algorithm based on the combination of EM and a *NB* classifier. The following subsections describes main characteristics of the *NB* and EM semi-supervised algorithm when applied to the network traffic classification.

3.1 Naïve Bayes

Network traffic classification implies assigning a traffic instance to one or more predefined classes $C = \{c_1..., c_k\}$. In the simplest case only two classes are considered, attacks and normal traffic.
Let L be a dataset containing labeled traffic instances $\{l_i..., l_{|L|}\}$, classification can be done by just estimating the probability of each class c_k given l_i and a set of distribution parameters denoted ϕ. Then, classification is done according to the $c_k \in dom(C)$ with maximum probability.
An estimation of the class c_k probability, $P(c_k|l_i;\hat{\phi})$, can be obtained by means of Bayes theorem, giving

$$P(c_k|l_i;\hat{\phi}) = \frac{P(c_k|\hat{\phi})P(l_i|c_k;\hat{\phi})}{P(l_i|\hat{\phi})} \tag{1}$$

Note that the numerator $P(l_i|\hat{\phi})$ is the same for all $c_k \in dom(C)$ and can be removed from the equation. Whereas the class c_k Prior probability estimator $P(c_k|\hat{\phi})$ is defined as the count of instances l_i belonging to class c_k in the whole L dataset. Equation (2) defines class Prior for class c_k.

$$P(c_k|\hat{\phi}) = \frac{\sum_{i=1}^{L} l_i \in c_k}{|L|} \tag{2}$$

According to naïve Bayes independence assumption, l_i attributes are mutually independent given the class, so that

$$P(l_i|c_k;\hat{\phi}) = \prod_{j=1}^{N} P(l_i^j|c_k;\hat{\phi}) \tag{3}$$

Due to l_i traffic instances contain continuous attributes, a common approach is to assume that the distribution followed by attribute l_i^j given C is a Gaussian [14], that is $P(l_i|c_k;\hat{\phi}) = N(l_i^j|\mu_{jk}, \sigma_{jk}^2)$. Estimates for μ_{ik} and σ^2 are defined as follows

$$\hat{\mu}_{jk} = \frac{\sum_{i=1}^{L} l_{ik}^{j}}{|l_{ik}|} \tag{4}$$

$$\hat{\sigma}_{jk}^{2} = \frac{\sum_{i=1}^{L} (l_{ik}^{j} - \mu_{jk})^2}{|l_{ik}|} \tag{5}$$

When *NB* is trained with an small set of labeled traffic instances, classification will suffer performance loss due to a high variance in parameter estimates ϕ. However, when this small set is combined, by means of EM, with a large set of unlabeled traffic instances, it is possible to improve parameter estimates. Details of this process are shown in the following section.

3.2 Expectation Maximization

EM is an iterative algorithm for maximum likelihood or maximum a posteriori estimation in problems with incomplete data [7]. In this case, unlabeled data are considered incomplete due to missing class labels.
The basic algorithm written in pseudo code is shown in Figure 1. EM consists of two steps, an Expectation step or *E-step* plus a Maximization step or the *M-step*. The process is initialized with the *M-step*, where *NB* classifier parameters ϕ are estimated using only labeled traffic instances from dataset L. Then, the cycle begins with an *E-step* that uses recently learned *NB* classifier to probabilistically label the unlabeled traffic instances in dataset U. Then, paremeters ϕ are estimated once again in a new *M-step* but using the union of datasets L and U. Algorithm iterates until estimates of the parameters ϕ does not change.

```
1: Learn parameter estimates $\hat{\phi}$ for a NB classifier $f$ using traffic instances
   from the labeled set $L$
2: repeat
3:     for each traffic instance $l_i$ in $U$ do
4:         Using the current classifier $f$ classify each $l_i$
5:     end for
6:     Learn $\phi$ for a new naive Bayes classifier $f$ using the union of $L$ and $U$
7: until the parameter estimates $\phi$ converge
8: Return the classifier $f$ from the last iteration
```

Fig. 1: naïve Bayes combined with EM Algorithm

4 Experiments

This section aims to evaluate the performance of a simples *NB* classifier when it is combined with EM algorithm (denoted *NBEM*). A set of experiments are conducted in order to evaluate the minimum amount of labeled traffic required by *NBEM* for achieving acceptable performance.

4.1 Dataset description

The experiments were conducted over five weeks of the 1998 DARPA data set, a dataset widely used for intrusion detection evaluation. DARPA dataset contains around 1.5 millions traffic instances with almost 50% of them labeled as attacks.

Selected attributes for describing the input data consist of a number of fields available from a network traffic instance, as well as other high level attributes obtained after some network packet preprocessing.

Table 1 shows a total of fifteen fields related to a network traffic instance. Attributes such as *protocol*, *tcp.srcport*, *tcp.dstport*, *ip.src* and *ip.dst* are easily obtained from network connection. Remaining ones are higher level attributes which provide information related to connection time and data transferred.

Table 1: Basic attributes of individual traffic connections.

Feature Name	Description	Quantity
connection.time	Time of the connection in hours, minutes and seconds	3
protocol	Type of protocol, e.g ssh,http,ftp	1
tcp.srcport	TCP source port	1
tcp.dstport	TCP destination port	1
ip.src	IP source address	4
ip.dst	IP destination address	4
tcp.len	Number of bytes transfered	1
num.pkts.src.dst	Number of packets from src IP to dst IP	1
num.pkts.dst.src	Number of packets from dst IP to src IP	1
num.ack.src.dst	Number of packet with ACK flag active from src to dst	1
num.ack.dst.src	Number of packet with ACK flag active from src to dst	1
num.syn.src.dst	Number of packet with SYN flag active from src to dst	1
num.syn.dst.src	Number of packet with SYN flag active from dst to src	1
num.bytes.src.dst	Number of bytes from src to dst	1
num.bytes.dst.src	Number of bytes from dst to src	1

A second set of attributes are shown in Table 2. These attributes provide information about the number of connections using a five-second time windows as well as information related to the last 100 connections.

Many of these fields have been used in previous works [15,16] and have provided a good trade off between overall performance and the computational effort needed for training process. Selected fields are represented according to *Quantity* value shown in tables, resulting a total of 32 attributes used for training proposed classifiers.

Table 2: Attributes involving many connections

Feature Name	Description	Quantity
Information about connections in the last five seconds		
count.time.src.	Number of connections from the same address as the current connection source address	1
count.time.dst	Number of connections to the same IP address as the current connection destination IP address	1
count.time.srv.src	Number of connections from the same service as the current connection	1
count.time.srv.dst	Number of connections to the same service as the current connection	1
Information about the last 100 connections		
count.src	Number of connections from the same address as the current connection source address	1
count.dst	Number of connections from to the same address as the current connection destination address	1
count.srv.src	Number of connections from the same service as the current connection	1
count.srv.dst	Number of connections to the same service as the current connection	1

4.2 Dataset sampling

A randomly selected 1% subset of the DARPA data is used for the training process, whereas another 0.5% subset is used for testing purposes, following standard ratios used in classification problems.

The number of traffic instances containing attacks is extremely variable. Therefore, no assumption is made about attack class distribution. Experiments are conducted against datasets containing attack distributions of 10%, 20%, 30%, 50%, 60%, 70%, 80%, 90%. The 0.01 fraction of the whole DARPA dataset (i.e., 1% of it) with proportion of attacks of p% is sampled from the whole dataset in two steps, one that samples attacks from the set of all attacks, and another for sampling the normal data from the set of all normal traffic instances. To maintain the p% ratio of attacks in the resulting 1% dataset, a fraction $p \times 10^{-4}$ of attacks are randomly and uniformly sampled from the set of all attacks. Similarly, a fraction of $(1 - p) \times 10^{-4}$ is randomly and uniformly sampled from the set of all normal traffic instances.

Let T be a randomly and uniformly selected subset sampled for training as described in previous paragraph, a percentage of T is considered as labeled (denoted L) whereas the remaining instances are considered as unlabeled (denoted U).

As usual, for the supervised strategy only, L is used for learning the naïve Bayes classifier. On the other hand, for the semi-supervised strategy, an union of L and U datasets is used in EM algorithm as shown in Figure 1. As in real life situation L is a dataset labeled by experts, random sample of L is forced to maintain equally distributed classes while U sample keeps proper dataset attack

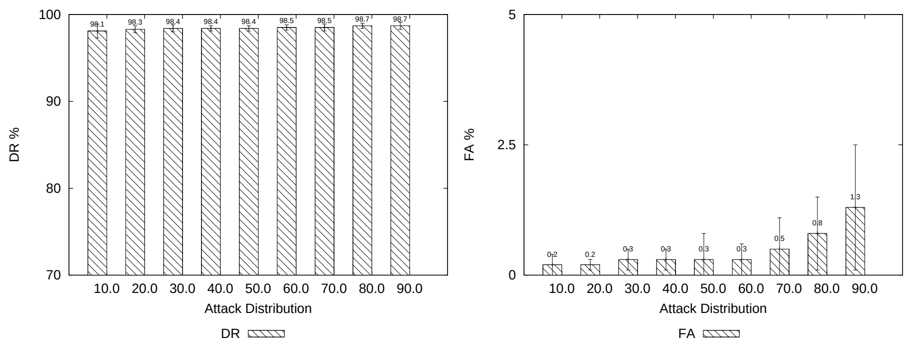
distribution. The classification performance of both classifiers, *NB* on L and *NBEM* on L and U, are evaluated on a test dataset.
For statistical significance a total of 40 repetitions of the experiments are conducted using different randomly and uniformly selected subsets for each attack distribution.

4.3 Performance Metrics for Network Intrusion Evaluation

Standard performance metrics for network intrusion evaluation are used for comparing the different approaches discussed. These metrics correspond to Attack Detection rate (DR) and False Alarm rate (FA).
DR is computed as the ratio between the number of correctly detected attacks and the total number of attacks. While FA rate is computed as the ratio between the number of normal connections that are incorrectly classified as attacks and the total number of normal connections.

4.4 Evaluation of Naïve Bayes classifier

Before evaluating the *NBEM* approach it is important to evaluate performance of a simple *NB* classifier.
Figure 2 shows performance of a *NB* classifier trained with the whole sample labeled dataset T.



(a) DR values (b) FA values

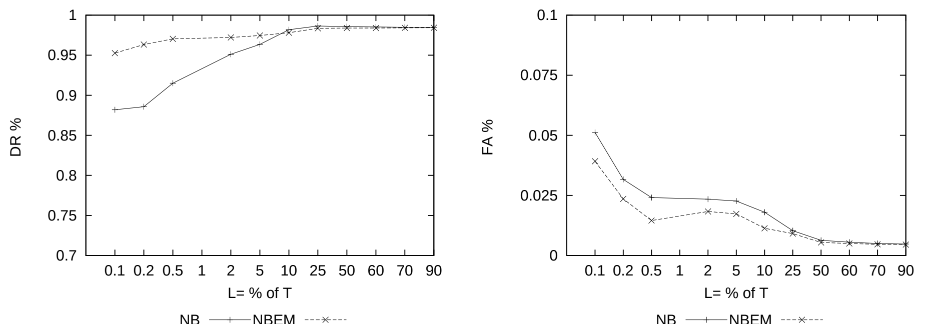
Fig. 2: DR and FR for *NB* trained with T

This training process corresponds to the traditional supervised strategy. A simple *NB* classifier assuming a Gaussian distribution for continuous features was using as discussed in section 3.
Despite the Gaussian distribution assumption can be considered rather strong, results shows *NB* seems to be very accurate, showing a very high average DR, (above 98%) together with a low average FA (beyond 2.0%) over all the attack distribution datasets.

Behaviour presented by *NB* on DARPA 1998 dataset may be favored due to some unrealistic procedures observed during dataset generation. For a more detailed explanation, the reader is referred to [17].

4.5 Evaluation of Naïve Bayes with EM

Experiments in this section aim to evaluate the performance of the *NB* and EM combined approach.
Figure 3 shows average DR and FA values for *NB* and *NBEM* along datasets with different attacks distributions, following training processes as mentioned in subsection 4.2.
As can be observed major benefits provided by *EM* are observed when labeled dataset L takes values up to 10% of the training set T. Beyond this point, no significant appreciable differences between *NB* and *NBEM* are observed. Therefore, *EM* seems to be unnecessary and could be avoided.
As can be seen in Figure 3 (a), in the case of DR, *NBEM* shows values from 95% to 99% while in the case of *NB* values shown are from 88% to 98%.



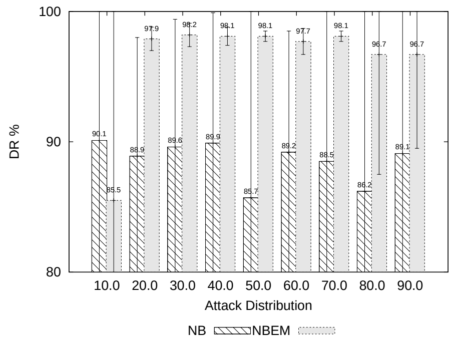
(a) Average DR values for *NB* and *NBEM* (b) Average FA for *NB* and *NBEM*

Fig. 3: Average DR and FA values for *NB* and *NBEM* along datasets with different attacks distributions for selected L sizes.

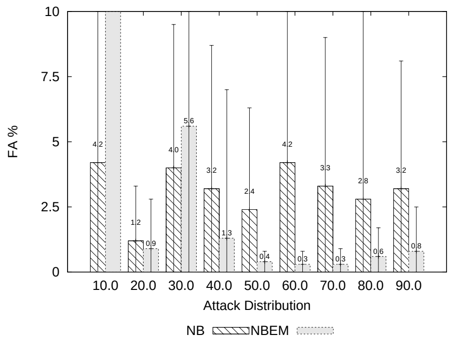
Although not as remarkable as in the case of DR, the *NBEM* approach also provides benefits in the case of FA. Figure 3 (b) shows FA values from 0.4% to 3% for *NBEM* while for NB values are from 0.5% to 5%.
Following figures show in details different size for L which provides a good trade off between the labeled dataset size and the benefits obtained when *NBEM* is used.
Figure 4 shows DR and FA performance for both algorithms. A L labeled dataset with 0.2% of T is used for training *NB* whereas *NBEM* is trained with L and $U = T - L$. As can be seen, DR values for *NB* vary from 85% to 90% under different attacks distributions, which is considerable worse than performance shown in Figure **??** when *NB* is trained using the complete labeled dataset T. In the case of FA, an increment can be also observed, however values do not exceed

5%. On the other hand, for *NBEM*, DR ranges between 93% to 97% for all of the attack distributions. In the case of FA, significant performance improvements are observed for dataset distributions beyond 30% of attacks.

Also notice that *NB* results show a significant variance which in many cases is considerable reduced when the *NBEM* approach is used.



(a) DR for *NB* and *NBEM*



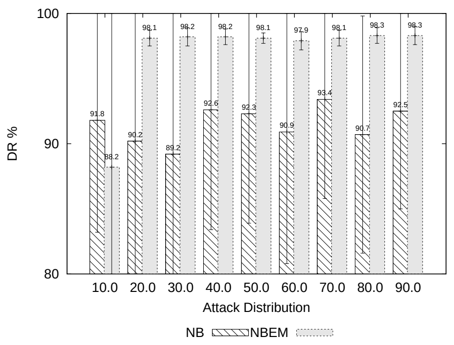
(b) FA for *NB* and *NBEM*

Fig. 4: *NB* trained with $L = 0.2\%$ of T and *NBEM* trained with $L = 0.2\%$ of T and $U = T - L$

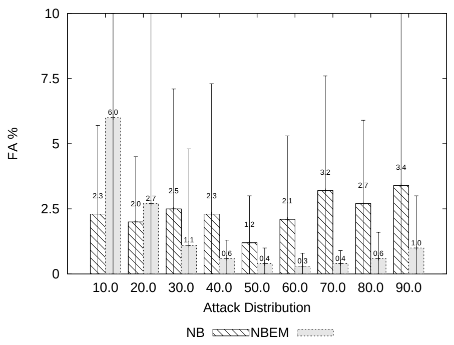
Figure 5 shows results when a dataset L containing 0.5% of T is used for training. DR values for *NB* are around 92% whereas FA values remain below 4%. On the other hand, for this size of L, the semi-supervised approach holds above 98% for DR, in all of the attacks distributions, with the exception of a 10% attack distribution where DR decreases to 88%. In the case of FA, *NBEM* benefits appear for dataset with more than 20% attacks. For those cases, FA values drop down to 1%. Here, *NBEM* also helps in reducing the high variance presented by *NB*.

When $L = 2\%$ of T, *NB* begins to exhibit appreciable performance improvements. As can be observed in Figure 6, DR values vary from 94% to 96% for every attack distribution, whereas FA drop down to a 3.5%. *NB* trained with $L = 2\%$ shows a performance closer to the values when the complete training dataset is used. However, even in this case, the use of the *EM* approach shows some benefits. DR values remain above 98% for every attack distribution. And in the case of FA, they keep being lower than the ones exhibited by *NB*. Except for dataset containing 10% and 20% of attacks.

Finally, Figure 7 shows results for $L = 10\%$. In this case, *NB* exhibits DR values around 98% for every attack distribution while FA slightly varies from 1.5% to 2.2%. Due to near-optimal performance shown by *NB*, benefits provided by *EM* are less appreciable. In the case of DR, *EM* shows a performance very similar to *NB*. Although, for datasets under some attack distributions, *EM* slightly degrades DR value. On the other hand, although small in proportion, FA values

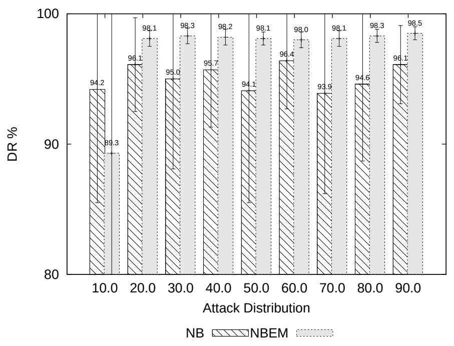


(a) DR for *NB* and *NBEM*

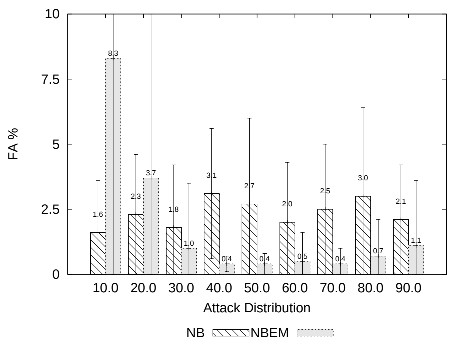


(b) FA for *NB* and *NBEM*

Fig. 5: *NB* trained with $L = 0.5\%$ of T and *NBEM* trained with $L = 0.5\%$ of T and $U = T - L$



(a) DR for *NB* and *NBEM*



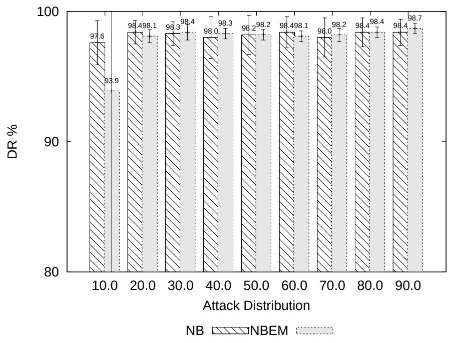
(b) FA for *NB* and *NBEM*

Fig. 6: *NB* trained with $L = 2\%$ of T and *NBEM* trained with $L = 2\%$ of T and $U = T - L$

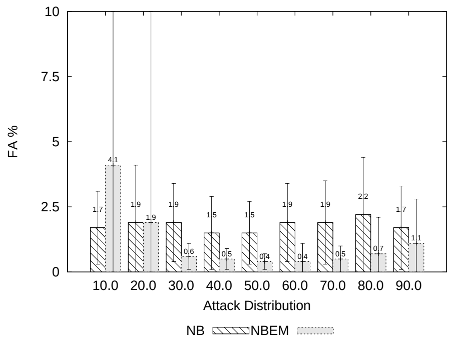
are reduced by *NBEM* in all attack distribution except for dataset with 10% of attacks
Beyond this point, as already suggested by averages shown in Figure 3 the use of EM does not improve *NB* performance.

5 Conclusions and Future Work

Classification performance exhibited on datasets under different attack distributions shows the use of the proposed semi-supervised strategy is suitable for intrusion detection systems.
Standalone Naive Bayes classifier shows extremely accurate results but requires fully labeled datasets.
The use of the *EM* algorithm provides results comparable to the obtained with a Naive Bayes classifier with considerable less labelling effort. Results for the 1998 DARPA dataset indicate that with only a 0.5% of the training set used by



(a) DR for *NB* and *NBEM*



(b) FA for *NB* and *NBEM*

Fig. 7: *NB* trained with $L = 10\%$ of T and *NBEM* trained with $L = 10\%$ of T and $U = T - L$

a standalone *NB* classifier, *NBEM* exhibits similar performance for DR and FA under most of the attack distributions.
The use of *EM* continues to exhibit performance improvements when the number of labeled instances grows up to 10% of the subset sampled. Beyond this size *EM* does not exhibit appreciable benefits and in some cases some minor degradation in classifier performance is observed. *NBEM* seems to be susceptible to dataset distribution. For datasets with around 10% of attacks, *NBEM* has shown significant performance loss regardless the size of L.
Experiments suggest the viability of the *NB* and EM semi-supervised approach, however it is important to mention that *NB* high accuracy and consequent EM improvements could be favored by some artificial issues present in the DARPA 1998 dataset. Therefore, a number of experiments must be conducted on more realistic datasets in order to confirm these results.

6 Acknowledgements

The authors want to thank the financial support received by the projects PAE-PICT 2312 granted bye ANPCyT and 06/M010 granted by UNCuyo.

References

1. Ghosh, A.K., Schwartzbard, A., Schatz, M.: Learning program behavior profiles for intrusion detection. In: Proceedings 1st USENIX Workshop on Intrusion Detection and Network Monitoring. (April 1999) 51–62
2. Lu, W., Traore., I.: Detecting new forms of network intrusion using genetic programming. Computational Intelligence, **20** (2004) 475–494
3. Chen, W.H., Hsu, S.H., Shen, H.P.: Application of svm and ann for intrusion detection. Comput. Oper. Res. **32**(10) (2005) 2617–2634
4. Zhu, X.: Semi-supervised learning literature survey. Tr 1530, University of Wisconsin - Madison (2008)

5. Casamayor, A., Godoy, D., Campo, M.: A recommender system for classification of non-functional requirements. In: 10th Argentinian Simposium of Artificial Inteligence. (2009)
6. Nigam, K., Mccallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using em. (1999) 103–134
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B **39**(1) (1977) 1–38
8. Lippmann, R., Fried, J.W., Korba, D.J., Das, K.: The 1999 darpa off-line intrusion detection evaluation. Computer Networks **34** (2000) 579–595
9. Laskov, P., Schafer, C., Kotenko, I.: Intrusion detection in unlabeled data with quarter-sphere support vector machines. In: In Proc. DIMVA. (2004) 71–82
10. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In: Applications of Data Mining in Computer Security, Kluwer (2002)
11. Mao, C.H., Lee, H.M., Parikh, D., Chen, T., Huang, S.Y.: Semi-supervised co-training and active learning based approach for multi-view intrusion detection. In: SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing, New York, NY, USA, ACM (2009) 2042–2048
12. Gornitz, N., Kloft, M., Brefeld, U.: Active and semi-supervised data domain description. In: Machine Learning and Knowledge Discovery in Databases. Volume 5781 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (2009) 407–422
13. Lane, T.: A decision-theoritic, semi-supervised model for intrusion detection. In: Machine Learning and Data Mining for Computer Security. Advanced Information and Knowledge Processing, Springer London (2006) 157–177
14. Mitchell, T.: Chapter generative and discriminative classifiers, Machine Learning. McGraw-Hill (2006)
15. Lee, W., Stolfo, S.J.: Data mining approaches for intrusion detection. In: In Proceedings of the 7th USENIX Security Symposium. (1998)
16. Catania, C., García Garino, C.: Reconocimiento de patrones en el tráfico de red basado en algoritmos genéticos. Inteligencia Artificial, Revista Iberoamericana de IA **12**(37) (2008) 65–75
17. McHugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. ACM Transactions on Information and Systems Securirty. **3**(4) (2000) 262–294