I. J. Computer Network and Information Security, **2016, 1, 35-42**
Published Online January 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.01.05

Intrusion Detection with Multi-Connected Representation

Abdelkader Khobzaoui
Dr Moulay Tahar University, Saida, Algeria
Email: akhobzaoui@yahoo.fr

Abderrahmane Yousfate
Djilali Liabes University, Sidi Bel Abbes, Algeria
Email: yousfate@univ-sba.dz

Abstract—Recently, considerable attention has been given to data mining techniques to improve the performance of intrusion detection systems (IDS). This has led to the application of various classification and clustering techniques for the purpose of intrusion detection. Most of them assume that behaviors, both normal and intrusions, are represented implicitly by connected classes. We state that such assumption isn't evident and is a source of the low detection rate and false alarm. This paper proposes a suitable method able to reach high detection rate and overcomes the disadvantages of conventional approaches which consider that behaviors must be closed to connected representation only. The main strategy of the proposed method is to segment sufficiently each behavior representation by connected subsets called natural classes which are used, with a suitable metric, as tools to build the expected classifier.

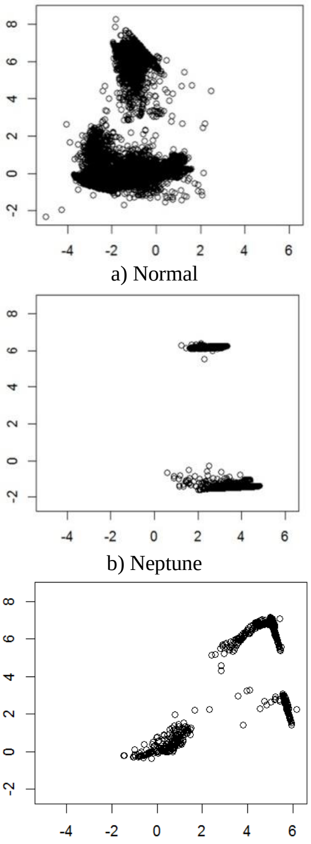
The results show that the proposed model has many qualities compared to conventional models; especially regarding those have used DARPA data set for testing the effectiveness of their methods. The proposed model provides decreased rates both for false negative rates and for false positives.

Index Terms—Connected representation, Discriminant Analysis, Mahalanobis distance, mixture of probability laws, multi-connected representation, natural class, synthetic class.

I. Introduction

Recently, methods of data mining and machine learning become the principal basis of intrusion detection system (IDS) study. The both methods are often statistics-based or computational intelligence-based.

In the literature, anomaly or misuse detections techniques used to build an intrusion detection system consider generally that each intrusion or normal behavior representation in the assumed topological space is implicitly a connected set. Actually, this assumption isn't evident. A simple illustration of the representation of some behavior classes by Principal Component Analysis applied to the KDD'99 data set [10] shows that there exists some representations which can be non-connected (for example, see Fig. 1); even the normal class is concerned. According to separating hyper plane theorem (Hahn-Banach theorem and its corollaries), this non-connectivity persists in high dimension spaces even if the dimension is infinite. Therefore, if geometric representation of some behavior (normal or abnormal) is non-connected by setting a number of features, the addition of further features preserves the non connectedness of this representation. This remark yields that both misuse and anomaly used models will be affected considerably by this work in case where these models assume that classes are connected unfairly.



a) Normal

b) Neptune

c) Satan

Fig.1. 2D representation of some behaviors

In this study, we assume that the connectedness of the behavior representations, considered implicitly as an assumption, is inappropriate (see figure 1). This fact leads us to adopt a new approach to identify behaviors based on existence of their multi-connected representation. Each representation behavior is then composed by one or many (connected) cluster(s). These clusters, called natural classes in the paper, are used as tools to describe expected classes, called learning classes, which are built by experts. The approach treats the task of detecting cyber attacks as a pure classification problem which can be applied to the abuse detection models as well as to the anomalies detection models. So, in detection stage, each traffic behavior representation can be assigned to its most appropriate natural class under some suitable conditions. Obviously, if natural classes constitute a partition of data representation space and if each learning class is a true union of natural classes, there exists an almost perfect estimate to affect every new element to its true learning class. Actually, this last case is rarely realizable because partitions generated by natural classes don't explain exactly those generated by learning classes. In practice, the process used to build natural classes (which are the basis of clustering in this study) is complex and needs a good initialization of the firsts approximate natural classes to ovoid initial cases which yield to high miss-classification rate. So we propose a suitable method to build expected natural classes. Therefore, in this work, we present an approach using some topological and stochastic tools which have the feature to simplify intrusion detection method. It enables the classifier to be continuously self-adjusted and gives it a good ability to detect attacks with a high accuracy.

This classification is the base of the proposed training algorithm which is able to detect known attacks as well as the unknowns if they are suitably labeled under a descriptive statistical process. The discovery of the aberrant activities is done by comparing the network traffic with attack representation captured in the knowledge base. To build appropriate model, each measurement is considered as a realization of a random multivariate vector. Each realization must be analyzed individually.

Hence the proposed approach can detect anomalies with recognizing known intrusions and over labeling new intrusion. This approach can also update the model as soon as a new natural class arises. Then the expert can affect this new natural class to its predetermined learning class or considers it as a new attack class.

Generally, the systems used for detecting intrusions can be categorized into two categories: misuse detection and anomaly detection. The first category can detect many well-known intrusions with high detection rate, but it is unable to detect new intrusions, even more its signature database needs to be updated manually. The second category is built on detecting anomaly in observed data by noticing deviations from normal model. But, in the case of complex network traffic, this last category needs to collect a lot of data to build a suitable model of IDS. This work can be mainly classified in the second category with a characterization of each behavioral representation by natural classes which are the basic component of the proposed IDS. Actually, we assume thereafter that the effectiveness of IDS is measured by detecting the true traffic behavior, otherwise by reducing confusions, particularly the false negatives.

The data mining process used in this work extracts continuously, from audit trails and new knowledge, potentially useful information from dataset in the outlook to discover closeness to not yet known behavior (normal or intrusion) to update previous model (Fig. 2).

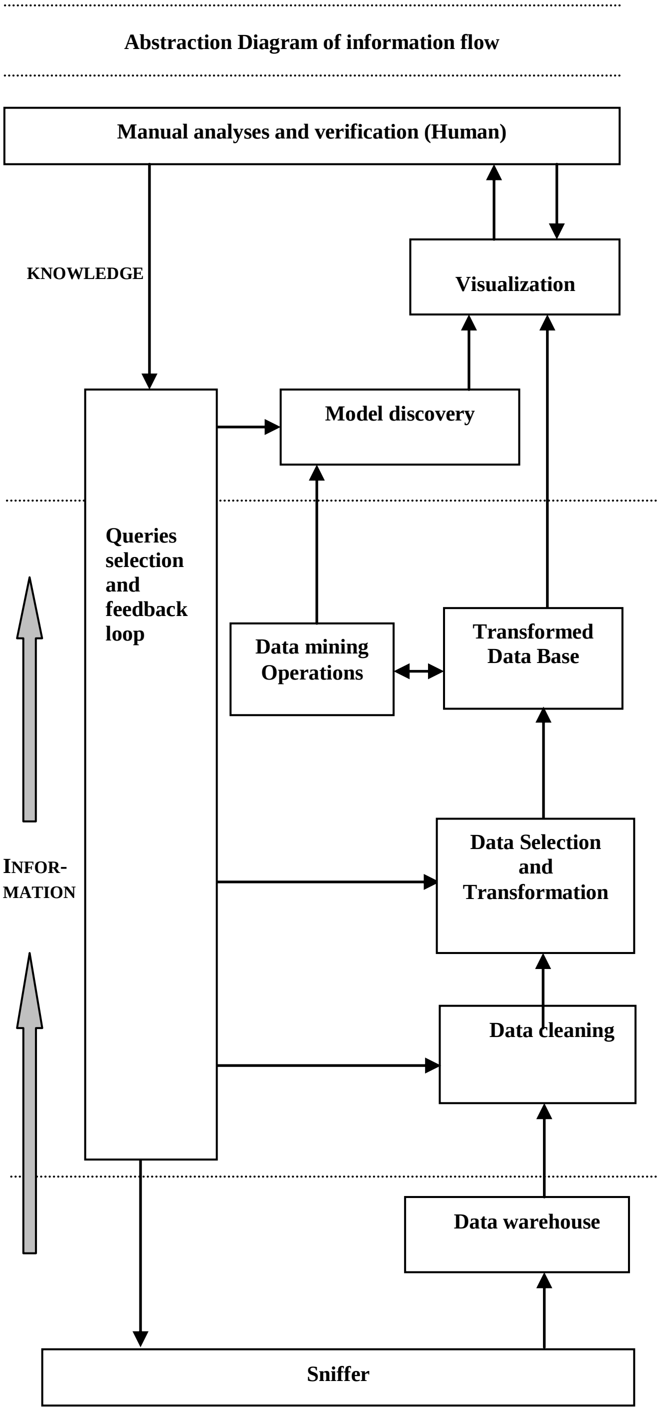

Abstraction Diagram of information flow
Manual analyses and verification (Human)
KNOWLEDGE
Visualization
Model discovery
Queries selection and feedback loop
Data mining Operations
Transformed Data Base
INFOR-MATION
Data Selection and Transformation
Data cleaning
Data warehouse
Sniffer

Fig.2. Data mining process for Intrusion detection

Data resulting from the network monitoring tools are collected and classified in a data warehouse. Therefore that's subjected to a cleaning process which consists in removing the erroneous, incomplete and aberrant data, to check the total coherence of the data. The following step consists in selecting the subsets of data on which data mining operations will be carried out. This stage is known as "data selection and transformation phase" it may be either manual or automatic. Data mining operations are carried out on a subset of data then extended on the whole data. The model emerging will be validated. New detection knowledge is extracted from the data set, then a refined model is developed to try to predict future events based on the history. This process is known as "model discovery". Moreover the analysts require visualization tools to adjusting the parameters, the data sets and associations.

The paper is organized as follows. In section II, the background behind the problem is described in "Related Work". The theoretical problem formulation on the framework of expected classes is presented in section III. The proposed model, including the algorithm, is developed in section IV. Section V presents the results of proposed model and comparisons with those of some well known related works which have used KDD'99 data sets. Finally, conclusion and extension for future research is mentioned in section VI.

II. Related work

From works of the seventies on the computer security auditing and monitoring and since the seminal Denning's paper on the intrusion detection modeling [6], a variety of techniques and methods was proposed and used to detect electronic intrusions. The earliest systems that can be considered as being based on Denning's approach were principally:

- Intrusion Detection Expert System (IDES) which became "Next-generation Intrusion Detection Expert System" (NIDES) developed at SRI International; it uses a deductive process based on statistics and a rule-based system for real-time detection of security violation,
- Multics Intrusion Detection and Alerting System (MIDAS), developed for National Computer Security Center (NCSC), uses a large part of statistical analysis to characterize normal system and user behavior,
- Wisdom & Sense, developed at Los Alamos National Laboratory, is an anomaly detection system which mines association rules statistically from historical audit data,
- Haystack, developed by Tracor Applied Sciences Inc. and Haystack Labs for US Air Force, was designed to detect essentially insider abuse.

All these earliest systems were implemented using rule-based techniques and adapted statistical models. For more information see [15].

In the 1990s, the transitions from manual approach to automatic anomaly detection approaches have been extensively developed.

In last decade, methods of data mining and machine learning have become essential to automatically discover and model features of user's normal or abnormal behaviors. The main tools used for detecting intrusion are principally statistical and bio-inspired methods; combination of the both methods is used too. Recent reviews of existing studies can be found in [18] and [23]. Some related works which have used KDD'99 data sets to validate their models can be found in [1], [2], [3], [8], [9], [11], [12], [19], [21], and [25]. These works have used essentially one or many techniques (in hybridization case) from the following classification approaches: Multivariate analysis, K-means, naïve Bayes, nonparametric neural networks, support vector machines, decision tree, immune systems, evolutionary computation, swarm intelligence, ... For gains in response system time or for avoiding the noise of redundancy information, some authors have used appropriate feature reduction using principal component analysis (PCA) [7], [24], linear discriminant analysis (LDA) or general discriminant analysis (GDA) [20].

Generally hybrid systems give better performances than those of single systems. However, even the hybridizations of all kinds reach their limits. It was noted that none of them is able to detect all types of intrusion attacks optimally and none exploits explicitly the non connectedness of behavior representation classes. To obtain a substantial improvement in the performance of IDS, the proposed approach reconsiders the topological structure feature of the behavior representation classes with a view to exploit the advantages of both misuse detection techniques that those anomaly detection.

Exploiting the basic topological framework of the behavioral representations instead of hybridizing methods that have already been used (see [20] for example), we expect that we can get more accurate results.

Moreover, if the proposed approach is categorized as a simple technique, by hybridization we can improve the results.

III. Problem Formulation

In this work, we consider that data are represented in a p-dimensional Euclidean space. We assume also that the probability measure used in the system is acquired from a mixture of Gaussian densities f_j characterized by their expectations µj and their variance matrices σ_j; j=1, …, m. The mixture is a convex combination of the unimodal densities f_j. Hence the considered mixture density can be written:

$$g(x) = \sum_{j=1}^{m} \alpha_j f_j(x) \tag{1}$$

with

$$\sum_{j=i}^{m} \alpha_j = 1; \alpha_j > 0; j = 1, \ldots, m$$

In this model the interesting points are essentially the modal points of g which are its local maximums. However, the number of maximums of *g* (denoted k) is generally greater than the number of learning classes (denoted m) when interaction between densities occurs; even if learning classes are connected. If classes are multi-connected k must be larger. Firstly, we have to differentiate between factual modes, associated to the unimodal densities f_j, j = 1, ..., m, composing the mixture, and the synthetic modes obtained by interactions.

Modes are represented by points where density reaches its maximum. In experimental case, maximum is estimated by a point whose neighborhood concentration is the largest. In this study, this estimated point is called accumulation point. It is estimated using clustering techniques applied to learning data. Each cluster obtained by this method is an estimate of a natural class. These natural classes, denoted Ci; i=1, ..., k, can be represented into a partition of R^p whose the parts are connected. The study is restricted to each part which contains one and only one natural class where for all x contained in C_i ; i = 1, ..., k, ∃ε(x)>0, such that

$$\{x; x \in R^p; \mathrm{g}(x) > \varepsilon(x)\} \subset \mathrm{C}_i$$

If natural classes constitute a sub-partition of a partition generated by the learning classes, our study becomes very easy. However, this case occurs rarely in practice.

Thereafter, each natural class is associated to a factual mode; synthetic modes (which generate synthetic natural classes) must be avoided. Discrimination between factual natural classes and synthetics ones is done using maximum likelihood method when the representation space dimension is weak [14]. We have just to resolve the equation:

$$\nabla_\theta \log\left(L\left(x_1, x_2, \ldots, x_n =; \theta\right)\right) = 0 \qquad (2)$$

where L is likelihood function which corresponds to n densities product, such each density is itself a mixture of m weighted densities. Each natural class is characterized by one center μ_j, a variance matrix Σ_j and a weight α_j. The general vector parameter θ is identified to (α_1, ..., α_m, μ_1, ..., μ_m, Σ_1, ..., Σ_m) where α_i is scalar, μ_i is a p-dimensional vector and Σ_i is a symmetric matrix in R^pxR^p (i=1,…,m). So θ depends on m(1+p+ p(p + 1)/2) unknown scalar parameters.

When equation (2) has unique solution, the method can be considered as a powerful tool to estimate the real natural class's peaks.

As the system (2) is non-linear, with high number of parameters, numerical implementation of good approximate solutions is too complex. However, if real natural classes are localized, the problem can be reduced to m-subsystems. Every subsystem needs estimation of just one center and one variance matrix. To use a parallel implementation one has temptation to build a local metric for each subsystem. However, we have observed that this approach produces a very important miss-classification rate. To avoid this weakness, we have built a global topological metric adapted to the problematic and local metric for the tests.

IV. Developed Model

To exploit the method within the framework of the training set, one considers a sample of n observations e_1, e_2, ...,e_n of p variables X_1, X_2, …, X_p describing m behaviors of which one corresponds to a normal traffic and the m−1 others correspond to attacks. So data are presented as a matrix X as follows:

$$X = \begin{array}{c} \\ e_1 \\ . \\ e_n \end{array} \begin{array}{c} \begin{array}{ccc} X_1 & .... & X_p \end{array} \\ \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} \end{array}$$

Each natural class C_i (both factual and synthetic) is characterized by its centroid $\overline{c_i}$ given by:

$$\overline{c_i} = \frac{1}{n_i} \sum_{\{j; e_j \in C_i\}} e_j \qquad (3)$$

where n_i is the cardinal of class C_i.

Information which will be exploited is summarized essentially into dispersions of the records e_i around their centroids expressed in terms of variances.

Given a new element t representing a new connection and a basis ($\overrightarrow{\overline{\mathrm{v}}1}$, $\overrightarrow{\overline{\mathrm{v}}2}$, ..., $\overrightarrow{\overline{\mathrm{v}}\mathrm{p}}$) in the representation space, which is an affine space, one has:

$$\overrightarrow{ot} = \overrightarrow{oc_i} + \sum_{j=1}^{p} \overrightarrow{\beta_j \mathrm{v}_j} \qquad (4)$$

where *o* is the origin.

Generally, using vector space properties, one can write:

$$\overrightarrow{c_i t} = \sum_{j=1}^{p} \overrightarrow{\beta_j \mathrm{v}_j} \qquad (5)$$

By noting $(\sigma_{\overrightarrow{c_i t}})^2$ the conditional variance to $\overrightarrow{c_i t}$ direction at the fixed risk α, decision rule can be constructed under the confidence interval:

$$I_{c_i t} = \left[-r_{\alpha/2} \sigma_{\overrightarrow{c_i t}} \frac{\overrightarrow{c_i t}}{\| \overrightarrow{c_i t} \|}, r_{\alpha/2} \sigma_{\overrightarrow{c_i t}} \frac{\overrightarrow{c_i t}}{\| \overrightarrow{c_i t} \|} \right] \qquad (6)$$

where $r_{\alpha/2}$ is the quantile of order $1-\alpha/2$, $\frac{\overrightarrow{c_i t}}{\lVert\overrightarrow{c_i t}\rVert}$ is the unit vector in the direction of $\overrightarrow{c_i t}$ and origin is translated to $\overline{c_i}$ position. Quantity $\sigma_{\overrightarrow{c_i t}}$ is obtained from variance intra class C_i (within variance) projected into the direction $\overrightarrow{c_i t}$. As the projection is linear, then the law of the projected points on the line generated by $\overrightarrow{c_i t}$ remains Gaussian; this enables to justify (6). Then, any new record, represented by t, which is characterized by its neighborhood topological properties and its associated measure, is studied just as a point in a specific interval.

Measurement aspect is essentially characterized by the variance-covariance matrix. Generally, affectation of a new observation to a natural class is easy. However, its affectation to a learning class remains difficult; especially if this learning class is multi-connected.

In fact, if the natural classes are a real sub-partition of learning classes, identifying behaviors (represented mainly by learning classes) will be without risk. Otherwise the misclassification rate risk becomes significant; which constitutes the majority of cases encountered in intrusion detection. To reduce the risk of misclassification quasi-optimally, Algorithm 1 is proposed to construct appropriate natural classes.

When natural classes are built, a new observation t will be assigned to its most appropriate class in two steps:

– First, the observation t is affected to the most appropriate natural class C_i such that:

$$i = \arg\min_{j\in\{1,\ldots,m\}} d(t,\overline{c_j}) \qquad (7)$$

where d(. , .) is Mahalanobis distance between t and the centroid of natural class C_j and m is the cardinal of the natural classes set. Recall that the Mahalanobis metric is associated with the inverse of the variance-covariance matrix. Thereafter if:

- C_i is completely included in a learning class L_j, then t is affected to L_j.
- C_i is partitioned over the set of learning classes $\{L_{j_1}, \ldots, L_{j_s}\}$, then t is affected to the nearest L_{j_k}; $j_k \in \{j_1, \ldots j_s\}$ according to (7).

This approach avoid requiring to develop a method separating true natural classes (which are more important) from synthetic natural classes (generated by interaction).

Noting that if a natural class C_i has cardinality less than variables number p, the model fails, because variance matrix will degenerate and one can't use local Mahalanobis distance. An alternative solution consists of combining variances of neighboring classes proportionally to their respective cardinals to build a generalized local Mahalanobis metric.

Eventually, if regularity problem persists we have to enlarge the neighborhood of the considered natural class to build a new generalized local Mahalanobis metric and so on.

When one class L_j is selected, the distance d between the L_j centroid and the observation t will be computed; affection of t to L_j will be confirmed only if d falls in the confidence interval specified by (6).

As there are a lot of natural classes which cardinal is less than p, general local Mahalanobis metric is replaced by global Mahalanobis metric. This choice is due to complexity to build best neighborhood to make an adapted local Mahalanobis metric.

Algorithm 1: Natural classes construction algorithm

```
Let  D ← {t_1… t_n}     Learning data set
Let  C ← {C_1,…, C_k} Initial clusters
s0 ← 1 Initial maximum standard deviation
loop.
  For each t in D do
    Affect t to nearest class $C_i$ according to the
    Mahalanobis distance
  End for
  Let $C_{new}$ ← ϕ Empty set
  For each cluster $C_i$ in C do
    For each label $L_j$ in $C_i$ do
      Put elements from $C_i$ labeled $L_j$ into new
      cluster $C_{ji}$
      Add $C_{ji}$ to $C_{new}$
    End For
  End For
  Let C ← $C_{new}$
  For each $C_i$ in C do
    Compute its mean $c_j$
  End for
  Let S ← ϕ
  For each $C_i$ in C do
    $V_j$← Covar($C_i$)
    s ← sqrt(sum(diag($V_i$)))
    Add s to S
  End for
  s ← max(S)
  If ( $0.9 \le \frac{s}{s_0} \le 1$ ) then return C else $s_0$ ← s
End loop
```

V. Tests Results

Using the R Environment for Statistical Computing [17], we test our model on the KDD'99 intrusion detection data set benchmark data set commonly used by many researchers for evaluation of intrusion detection techniques. The KDD'99 data set was created by processing the tcpdump portions of the 1998 DARPA intrusion detection system evaluation data set, created by Lincoln Lab under contract to DARPA; more information can be found in [13]. Actually, the KDD'99 data set includes a set of 41 features, gathered in 7 symbolic ones (which are quantified by a binary codification) and 34 numeric. A complete description of all the 41 features is available in [10]. These features fall into four categories:

- The intrinsic features of a connection include the basic features of individual TCP connections. For example, duration of the connection, the type of the protocol (tcp, udp, etc), network service (http, telnet, etc), etc.
- The content feature within a connection suggested by domain knowledge is used to assess the payload of the original TCP packets, such as number of failed login attempts.
- The same host features examine established connections in the past two seconds that have the same destination host as the current connection, and calculate statistics related to the protocol behavior, service, etc.
- The similar same service features examine the connections in the past two seconds that have the same service as the current connection.

These features describe 23 behaviors of which one corresponds to a normal traffic and the 22 others correspond to attacks (Table 1):

Table 1. Class label in KDD'99

Id-behavior	Behavior	Category
1	Back	DOS
2	buffer	U2R
3	ftp write	R2L
4	guess passwd	R2L
5	Imap	R2L
6	Ipsweep	PROBE
7	Land	DOS
8	Loadmodule	U2R
9	Multihop	R2L
10	Neptune	DOS
11	Nmap	PROBE
12	Normal	Normal
13	Perl	U2R
14	Phf	R2L
15	Pod	DOS
16	Portsweep	PROBE
17	Rootkit	U2R
18	Satan	PROBE
19	Smurf	DOS
20	Spy	R2L
21	Teardrop	DOS
22	Warezclient	R2L
23	Warezmaster	R2L

These 23 behaviors are gathered, by expert, in four categories:

- DOS (Denial of service): making some computing or memory resources too busy so that they deny legitimate users access to these resources.
- R2L (Root to local): unauthorized access from a remote machine according to exploit machine's vulnerabilities.
- U2R (User to root): unauthorized access to local super user (root) privileges using system's susceptibility.
- PROBE: host and port scans as precursors to other attacks. An attacker scans a network to gather information or find known vulnerabilities.

First, algorithm 1, a builder of natural classes, is applied to the training data set which contains 494,021 connections, where 97,278 correspond to normal traffic and the rest belong to one of 22 attack types. This algorithm provides a partition of 672 subsets which can be candidates of expected natural classes. Means and variances are estimated by usual estimation methods. These natural classes are used, firstly, as tools to describe learning classes and secondly to predict a new observation membership. Next, the detection system is tested on the test data set containing 4,898,431 connections where 972,781 correspond to normal traffic and the rest belongs to one of 22 attack types. The obtained results are summarized in Table 2.

Table 2. Main test results

	G.F.	T.D.	T.D.R.	F.N.	F.N.R.	C	C.R.
1	2203	2203	100.000	0	0.000	0	0.000
2	30	30	100.000	0	0.000	0	0.000
3	8	8	100.000	0	0.000	0	0.000
4	53	53	100.000	0	0.000	0	0.000
5	12	12	100.000	0	0.000	0	0.000
6	12481	12374	99.143	61	0.489	46	0.369
7	21	21	100.000	0	0.000	0	0.000
8	9	9	100.000	0	0.000	0	0.000
9	7	7	100.000	0	0.000	0	0.000
10	1072017	1071975	99.996	15	0.001	27	0.003
11	2316	2239	96.675	12	0.518	65	2.807
12	972781	970929	99.810	0	0.000	0	0.000
13	3	3	100.000	0	0.000	0	0.000
14	4	4	100.000	0	0.000	0	0.000
15	264	259	98.106	5	1.894	0	0.000
16	10413	10351	99.405	27	0.259	35	0.336
17	10	10	100.000	0	0.000	0	0.000
18	15892	15789	99.352	70	0.440	33	0.208
19	2807886	2807825	99.998	61	0.002	0	0.000
20	2	2	100.000	0	0.000	0	0.000
21	979	979	100.000	0	0.000	0	0.000
22	1020	1004	98.431	16	1.569	0	0.000
23	20	20	100.000	0	0.000	0	0.000
	4 898 431	**4 896 086**	**99.952**	**267**	**0.007**	**206**	**0.005**

In table 2, G.F., T.D., T.D.R., F.N., F.N.R., C and C.R. represent respectively Global Frequency, True Detection frequency, True Detection Rate, False Negative frequency, False Negative Rate, Confusion frequency (between attacks) and Confusion Rate (between attacks).

Note that “False Negative” corresponds to case where an attack is classified as a normal traffic. As confusion matrix between all behaviors is too big, we present, in the next, a summarized confusion matrix between the five categories of behaviors which are: NORMAL, DOS, U2R, R2L and PROBE described above (see table 3).

Table 3. Results by categories of behaviors

	NORMAL	DOS	U2R	R2L	PROBE
NORMAL	99.810	0.102	9e-03	0.049	0.030
DOS	0.002	99.997	0.00	0.000	0.001
U2R	0.000	0.000	100	0.000	0.000
R2L	1.421	0.000	0.00	98.579	0.000
PROBE	0.414	0.092	2e-03	0.007	99.484

Table 4. Performance comparison of proposed Algorithm

Method	Normal	DOS	PROBE	U2R	R2L
PNRule [1]	99.5	96.9	73.2	06.6	10.7
PSM & SVM [2]	99.8	97.9	98.6	68.9	19.5
CSFDTM [3]	99.2	100	71.4	84.4	99.5
NB-DT [4]	96.64	96.38	78.18	11.84	7.11
ADWICE [5]	97	99	99	92	31
PCA-SVM [7]	99.80	92.50	98.30	05.10	70.20
GP Multitrans-formation [8]	99.93	98.81	97.29	45.20	80.22
WANBT [9]	99.93	99.91	99.84	99.47	99.63
SVM+DGSOFT [11]	95	97	91	23	43
MCAD [12]	95.20	99.20	97.00	72.80	69.20
M.C.S. [16]		97.40	83.80	32.80	10.70
KDD cup 99 winner [16]	99.50	97.10	83.30	13.20	08.40
Multi-C. [19]		97.30	88.70	29.80	09.60
GDA+ANN [20]	98.95	98.63	96.25	24.12	12.08
GDA+C4.5 [20]	99.68	98.60	99.61	57.01	66.25
NNC+GDA [21]	98.75	98.63	96.50	24.12	12.08
I.C.A. [24]	69.6	98.0	100	71.4	99.2
Parzen-window. [25]	97.38	96.71	99.17	93.57	31.17
Proposed method	**99.81**	**99.997**	**99.48**	**100**	**98.58**

This condensed representation allows comparing obtained results with those presented in related works using the same data set. Conditional distributions are on rows. For example the first row means that normal behavior is identified as normal with estimate probability 0.99810 (True Negatives). It is identified as DOS behavior with estimated probability 0.00102, as U2R behavior with estimated probability 0.00009, as R2L behavior with estimated probability 0.00049 and as PROBE behavior with estimated probability 0.0003. Last four identifications are denoted “false positives”. From second to fifth rows, in table 3, when behavior is identified as NORMAL it is denoted “false negative” else it is denoted “true positive”.

In order to evaluate the performances of this method, results obtained by the proposed approach are compared with those presented in some related works using the same data set (Table 4).

Table 4 compares the performance results obtained using the proposed algorithm with those of 18 other models. Table 4 shows that all approaches have good performance to detect Normal and DOS behaviors, however for PROBE, U2R and R2L the models show considerable variations upon their accuracy detection.

As shown in the table, the proposed method has good performances; no other model offers better performance results. One quickly notices that proposed approach and that of [9] are the only ones who exceed 98.5% of accuracy for the five studied categories of behaviors. More, a comparison of the performances indicates that our approach has similar performances to those in [9]. However this last approach uses a hybrid learning algorithm for adaptive network intrusion detection using naive Bayesian classifier and a decision tree algorithm (ID3). As our approach is a single model, we expect that our method can be substantially improved by hybridization.

Moreover, restricted only to anomaly detection, the proposed approach exceeds 99.99% of a detection rate for known anomalies.

VI. Conclusion

The fact to consider that class behavior can have a multi-connected representation enabled us to build a framework which allows a high average detection rates and low rates for both false negatives and false positives. This novel approach is flexible and lets to combine advantages of anomaly and misuse detections together. It has the advantage to be quickly and easily implemented and updated. Assuming the non-connectedness of the behavior representations, results of our single model are competitive with those of the best hybrid models as shown in table 4. So, we conjecture that our method can be significantly improved by hybridization.

For example, when some natural classes aren’t convex, the use of Mahalanobis metric becomes inappropriate, so another metrics can be exploited to maintain or ameliorate the performances of the proposed approach.

In future work, we will use parallel algorithms to make the intrusion detection system faster.

Acknowledgment

This work was supported in part by a grant from DGRSDT, Algiers (Project: “Data mining and applications”).

References

[1] R. Agarwal and M. V. Joshi, PNrule: A new framework for learning classifier models in data mining, Technical TR 00-015, Department of Computer Science, University of Minnesota, 2000.

[2] E. Al Daoud. Intrusion Detection Using a New Particle Swarm Method and Support Vector Machines. World Academy of Science, Engineering and Technology, 2013. 77, 59-62.

[3] V. Barot, S.S. Chauhan and B. Patel. Feature Selection for Modeling Intrusion Detection. International Journal of Computer Network and Information Security (IJCNIS), 2014. 6(7), 56-62.

[4] N. Ben Amor, S. Benferhat and Z. Elouedi. Naive Bayes vs decision trees Intrusion Detection Systems. In Proceeding ACM Symposium on Applied Computing, Nicosia, Cyprus. 2004.

[5] K. Burbeck and S. Nadjm-Tehrani. ADWICE - anomaly detection with real-time incremental clustering, In Park, C.-s., Chee, S. (eds.) ICISC 2004. Springer 2005. LNCS, 3506, 407-424.

[6] D.E. Denning “An Intrusion-Detection Model”, IEEE transactions on software engineering, 1987, SE-13(2), 222-232.

[7] H.F. Eid, A. Darwish, A.E. Hassanien, A. Abraham. Principle components analysis and support vector machine based intrusion detection system, In 10th international conference on intelligent systems design and applications (ISDA), Cairo, Egypt, 2010. 363-367.

[8] K.M. Faraoun, and A. Boukelif. Securing network traffic using genetically evolved transformations, Malaysian Journal of Computer Science, 2006. 19(1), 9-28.

[9] D.W. Farid, J. Darmont, and M.R. Zahidur. Attribute Weighting with Adaptive NBTree for Reducing False Positives in Intrusion Detection. International Journal of Computer Science and Information Security (IJCSIS), 2010. 8 (1), 19-26.

[10] KDD99 Dataset. from http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/, 1999.

[11] L. Khan, M. Awad & B. Thuraisingham. A new intrusion detection system using support vector machines and hierarchical clustering, The VLDB Journal, 2007. 16, 507-521.

[12] S. Kumar, S. Kumar and N. Sukumar. Multidensity Clustering Algorithm for Anomaly Detection Using KDD’99 Data set. A. Abraham et al. (Eds.) Springer, ACC 2011, Part I, CCIS 190, 619-630.

[13] R. Lippmann, J.W. Haines, J.D., Fried, J. Korba, K. Das. The 1999 DARPA off-line intrusion detection evaluation. Computer Networks: The International Journal of Computer and Telecommunications Networking, 2000. 34(4), 579-595.

[14] G. J. McLachlan “Classification and mixture ML approaches to cluster analysis.” Handbook of Statistics, 1982. 2, 199-208.

[15] B. Mukherjee, L.T. Heberlein, K.N. Levitt “Network Intrusion Detection” IEEE Network, 1994. 8(3), pp. 26-41.

[16] H.A. Nguyen and D. Choi. «Application of Data Mining to Network Intrusion Detection: Classifier Selection Model. » Y. Ma, D.Choi, and S. Ata (Eds.), Springer: APNOMS 2008. LNCS 5297, 399-408.

[17] R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/, 2015.

[18] S. Rastegari, P. Hingston and C.-P. Lam. Evolving statistical rule sets for network intrusion detection. Applied Soft Computing, 2015. 33, 348-359.

[19] M. Sabhnani and G. Serpen. Application of Machine Learning Algorithms to KDD Intrusion Detection Data set within Misuse Detection Context. In Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications (MLMTA), 2003.1, 209-215.

[20] S. Singh, S. Silakari. Generalized Discriminant Analysis algorithm for feature reduction in Cyber Attack Detection System, International Journal of Computer Science and Information Security, 2009. 6(1), 173-180.

[21] M.-Y. Su. Real-time anomaly detection systems for Denial of Service attacks by weighted k-nearest-neighbor classifiers, Expert Systems with Applications, 2011. 38, 3492-3498.

[22] A. Tesfahun, D. L. Bhaskari. Effective hybrid Intrusion Detection System: A Layered Approach. International Journal of Computer Network and Information Security (IJCNIS), 2015. 7(3), 35-41.

[23] S.X. Wu, W. Banzhaf. The use of computational intelligence in intrusion detection systems: A review. Applied Soft Computing, 2010. 10, 1-35.

[24] D. Yang and H. Qi. A Network Intrusion Detection Method using Independent Component Analysis, In 19th International Conference on Pattern Recognition, 2008.

[25] D.-Y. Yeung & C. Chow. Parzen-window Network Intrusion Detectors. In: 16th International Conference on Pattern Recognition, Quebec, Canada, 2002. 11-15.

Authors’ profiles

Abderrahmane Yousfate is currently a Professor at Department of Computer Science and head of Mathematics Laboratory, Sidi Bel Abbès University, Algeria. He obtained his Doctorate in Applied Mathematics from Paul Sabatier University, Toulouse, France in 1981. Part of his current research includes application of data mining in intrusion detection and risk analysis.

Abdelkader Khobzaoui is currently Assistant Professor at the Computer Science Department, Tahar Moulay University, Algeria. His research interests integrate intrusion detection using data mining tools as well as parallel programming for exploiting HPC. He is associated member of Mathematics Laboratory of Sidi Bel Abbès University.

How to cite this paper: Abdelkader Khobzaoui, Abderrahmane Yousfate,"Intrusion Detection with Multi-Connected Representation", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.1, pp.35-42, 2016.DOI: 10.5815/ijcnis.2016.01.05