On Copulas-Based Classification Method for Intrusion Detection

Abdelkader Khobzaoui[1](✉), Mhamed Mesfioui[2], Abderrahmane Yousfate[3], and Boucif Amar Bensaber[2]

[1] Computer sciences Department, University of Saida
[2] Département de mathématiques et informatique, Université du Québec, Trois-Rivières, C.P, 500, Québec, Canada, G9A 5H7
[3] Laboratoire de mathématiques (LDM), University of Sidi Bel Abbès

Abstract. The intent of this paper is to develop a nonparametric classification method using copulas to estimate the conditional probability for an element to be a member of a connected class while taking into account the dependence of the attributes of this element. This technique is suitable for different types of data, even those whose probability distribution is not Gaussian. To improve the effectiveness of the method, we apply it to a problem of network intrusion detection where prior classes are topologically connected.

Keywords: Intrusion detection · Classification · Copula function · Copula density estimator · Empirical copula

1 Introduction

Let a set of d attributes $(a_1, a_2, \ldots, a_d)$ characterizing a vectorial space E. Let also $(x_1, x_2, \ldots, x_n)$ a set of E used as a learning set over m classes denoted $(\omega_1, \omega_2, \ldots, \omega_m)$ which are actually some disjoint subsets of E. To avoid the use of some predetermined probability laws of the attributes systematically, we intent to build a copulas-based classification model that estimates the true attributes laws and their dependency. Then one assigns each entity of E to its most likely class $\omega_i;\ i \in \{1, \ldots, m\}$. This entity must be well-assigned when it verifies an optimal probabilistic criterion.

In deterministic classification, this model builds, over the set E, an equivalence relation $\mathcal{R} \subset E \times E$ where $E/\mathcal{R}$ is a partition of E. In nondeterministic classification, for some adapted risks, classes are built using probability distributions. Each realization of the observed phenomenon distributes all elements over the different classes which yields to a partition of E. Partition changes with realizations (samples). To assign k elements o7ver m classes, in the deterministic case, one has only k steps to carry out all the affectations; each step requires m simplified tests. However, in the nondeterministic case, if one enumerates all possibilities for distributing k entities over m classes, then one finds m^k possibilities; each possibility requires k assigning steps. Each element is assigned to

Funded in part by DGRSDT, Algiers (PNR : Data mining and applications)

A. Amine et al. (Eds.): CIIA 2015, IFIP AICT 456, pp. 394–405, 2015.
DOI: 10.1007/978-3-319-19578-0_32

the class ω_j via a conditional probability $f(x \mid j)$ which can be estimated using the training data. Actually, we seek the most likely class k (maximum likelihood estimation) solution of : $k = \arg\max_j (f(x \mid j))$ where $f(x \mid j)$ denotes the conditional probability density function for x being a member of group ω_j.

To reduce the complexity of the problem, one assigns elements to their respective classes as in deterministic affectation. Elements whose ranges are near apexes are the most likely affected. In this case, each step requires m complicated tests; that means $k.m$ complicated tests.

In the following we'll denote $f^j(x)$ instead $f(x \mid j)$.

Many applications algorithms and models have been proposed to estimate this conditional probability density function : kernel-density estimator [32], k-nearest-neighbours (KNN) method [19], Learning Vector Quantisation (LVQ) [12], Support Vector Machines (SVM)[20] ...

In this work we present the use of the empirical copula function as an alternative for modeling dependence structure in a supervised probabilistic classifier. The set E is identified to a vector space $\mathbf{R}^d$ over the field $\mathbf{R}$ and we use the law of the considered phenomenon over E which can be well estimated if learning sample is sizable. So, the conditional probability density function $f^j(x)$ is estimated according the following algorithm:

```
Algorithm 1. Conditional probability density estimation
Require: :
  – $\{X_i\}_{i=1}^n$ an iid random sample from a $d-$dimensional distribution $F$ with density $f$.
  – $\Omega = \{\omega_1, \cdots, \omega_m\}$ $m$ learning classes.

1. for each $j \in \{1, \cdots, m\}$ do
2.     Transform the observations $X_i^j$ to $U_i^j = F_{ni}^j(X_i)$ where $F_{ni}^j$ estimates the ith marginal distribution restricted to a class $\omega_j$ and $X_i^j$ denotes observation from the class $\omega_j$
3.     Estimate the marginal densities $f_i^j$ for class $\omega_j$.
4.     Estimate the joint density of the transformed data restricted to the class $\omega_j$. this density w'il be noted $c^j$ and it is equivalent to the copula density.
5.     Estimate the joint density of the original data restricted to a class $\omega_j$ by:
           $f^j(x) = c^j\left(F_1(x_1), \ldots, F_d(x_d)\right) \prod_{i=1}^{d} f_i^j(x_i)$
6. end for
```

This approach allows to mitigate the curse of dimensionality and to treat the data in all situations even if the variance does not exist. It considers also the non-linear relationships between attributes.

New observation x will be affected to the class ω_r such that

$$r = \arg\max_j f^j(\mathbf{x})$$

The content of the paper is the following: The second section of the paper gives a short mathematical background of copula functions, Section 3 presents a copula based probabilistic model for classification. Section 4 presents the experimental setting to detect and identify intrusion in computer network and Section 5 summarizes the conclusions

2 Copulas Theory

Copulas play an important role in several areas of statistics and in Machine Learning as a tool of studying scale-free measures of dependence and as starting point for constructing families of bivariate distribution especially in applications where nonlinear dependencies are common and need to be represented.

The best definition of a copula is that given by referring to well know Sklar's theorem [28], [18], which states how a copula function is related to joint distribution functions.

Theorem 1 (Sklar's Theorem). *Let F be any d-dimensional distribution function over real-valued random variables with marginals $f_1, f_2, ..., f_d$,then there exists a copula function C such that for all $x \in \bar{\boldsymbol{R}}^d$*

$$F\left(x_1, \ldots, x_d\right) = C(f_1(x_1), \ldots, f_d(x_d)) \tag{1}$$

where $\bar{\boldsymbol{R}}$ denotes the extended real line $[-\infty, \infty]$ and $C : [0,1]^p \rightarrow [0,1]$.

The copula distribution can also be stated as joint distribution function of standard uniform random variables:

$$C(u_1, \ldots, u_p) = P\left(U_1 \leq u_1, \ldots, U_p \leq u_p\right) \tag{2}$$

where $U_i \sim U(0,1)$ for $i = 1, \ldots, p$.

Note that if $f_1(x_1), \ldots, f_d(x_d)$ in (1) are all continuous, then C is unique. Otherwise, C is uniquely determined on $Ran(f_1) \times Ran(f_2) \times \cdots \times Ran(f_d)$, where Ran stands for the range.

Conversely, if C is an d-copula and $f_1, \ldots, f_d$ are distribution functions, then the function F defined above is an d-dimensional distribution function with margins $f_1, \ldots, f_d$. For the proof, see [28].

From Sklar's theorem we see that for continuous multivariate distribution functions, the univariate margins and the multivariate dependence structure can be separated, and the dependence structure can be represented by a copula.

An important consequence of theorem 1 is that the d-dimensional joint density F and the marginal densities $f_1, f_2, \ldots, f_d$ are also related:

$$f(x_1, \ldots, x_d) = c\left(F_1(x_1), \ldots, F_d(x_d)\right) \prod_{i=1}^{d} f_i(x_i) \tag{3}$$

where c denotes the density of the copula C. The equation (3) shows that the product of marginal densities and a copula density builds a d-dimensional joint density.

The unique copula function related to the multivariate distributions F with continues margins $f_i; 1 \leq i \leq d$ is determined by

$$C(u_1, \ldots, u_d) = F(F_i^{-1}(u_1), \ldots, F_i^{-1}(u_d)) \tag{4}$$

where

$$F_i^{-1}(s) = \{t \mid F_i(t) \geq s\} \tag{5}$$

denote the pseudo-inverse of the univariate margins $F_1, \cdots, F_d$. Copulas are essentially a way of transforming the random variable $(X_1, \cdots, X_d)$ into another random variable $(U_1, \cdots, U_d) = (F_1(X_1), \cdots, F_d(X_d))$ having the margins uniform on $[0, 1]$ and preserving the dependence among the components. Without the continuity assumption, care must be taken to use equation (4); see [21] or [17].

3 Copula Function Estimation

To estimate copula functions, the first issue consists in specifying how to estimate separately the margins and the joint law. Moreover, some of these functions can be fully known. Depending on the assumptions made, some quantities have to be estimated parametrically, or semi or even non-parametrically. In the latter case, we have to choose between the usual methodology of using ”empirical counterparts” and invoking smoothing methods well-known in statistics: kernels, wavelets, orthogonal polynomials, nearest neighbors,... A non-parametric estimation of copula treats both the copula and the marginals parameter-free and thus offers the greatest generality.

Unlike the marginal and the joint distributions which are directly observable, a copula is a hidden dependence structure. This makes the task of proposing a suitable parametric copula model non-trivial and is where a non-parametric estimator can play a significant role.

Indeed, a non-parametric copula estimator can provide initial information needed in revealing and subsequent formulation of an underlying parametric copula model[3].

Non-parametric estimation of copulas dates back to Deheuvels [6], who proposed the so-called empirical copula defined by

$$C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left(F_{n,1}(X_{i1}) \leq u_1, \ldots, F_{n,d}(X_{i,d}) \leq u_d\right) \tag{6}$$

where $F_{n,i}$ are the empirical distribution function given by

$$F_{n,j}(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(X_{i,j} \leq x) \tag{7}$$

with j=1,. . . ,d and $\mathbf{u} \in [0, 1]^d$.

Let R_i be the rank of X_i among the sample $X_1, \ldots, X_n$. Observe that C_n is a function of ranks $R_1, R_2, \ldots, R_n$, because $F_{n,j}(X_i) = \dfrac{R_{i,j}}{n} \quad i = 1, \ldots, n$, namely;

$$C_n(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left(\frac{R_{i,1}}{n} \leq u_1, \ldots, \frac{R_{i,d}}{n} \leq u_d \right). \tag{8}$$

From this representation, one can consider $C_n(\mathbf{u})$ as discrete multivariate distribution with uniform marginals takings values in the set $\left[\frac{1}{n}, \frac{2}{n}, \ldots, 1\right]$. and so his density:

$$c_n(\mathbf{u}) = \frac{\partial C(u_1, ..., u_d)}{\partial u_1, \cdots, \partial u_d} \tag{9}$$

can be estimated by a standard kernel function:

$$\hat{c}_n(\mathbf{u}) = \frac{1}{n} \sum_{j=1}^{n} \prod_{i=1}^{d} h_i^{-1} K \left(\frac{u_i - U_{ji}}{h_i^{-1}} \right) \tag{10}$$

where U_i is the transformed of the original data given by $U_i = F_{n,i}^{j}(X_i)$ as described above. And a uni-variate kernel function $K(u)$ is any functions satisfying the following conditions:

(a) $K(x) \geq 0 \quad \textit{and} \quad \int_{\mathbf{R}} K(x)dx = 1$
(b) $\int_{\mathbf{R}} xK(x)dx = 0$ (Symmetric about the origin)
(c) Has finite second moment e.g. $\int_{\mathbf{R}} x^2 K(x)dx < \infty$

So, we have to choice both kernel function K and their smoothing parameter or bandwidth h. Actually, selection of K is a problem of less importance, and different functions that produce good results can be used (see table 1 for some examples).

In this paper, we use the Gaussian one given by:

$$K(v) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{v}{2}).$$

In practice, the choice of an efficient method for the calculation of h; for an observed data sample is a more complex problem, because of the effect of the bandwidth on the shape of the corresponding estimator. If the bandwidth is small, we will obtain an under-smoothed estimator, with high variability. On the contrary, if the value of h is big, the resulting estimator will be very smooth and farther from the function that we are trying to estimate[23](see figure 1).

For evaluating the tradeoff between bias and variance. Silverman[31] has suggested a frequently used rule-of-thumb bandwidth

$$h_n = 0.9(min(\hat{\sigma}, \frac{IQR}{1.34})n^{\frac{1}{5}},$$

Table 1. some kernel functions

	Kernel	**K(x)**
1	uniform	$\frac{1}{2}\mathbf{1}_{(\lvert x\rvert \le 1)}$
2	Epanechnikov	$\frac{3}{4}(1-x^2)\mathbf{1}_{(\lvert x\rvert \le 1)}$
3	Gaussian	$\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x}{2}\right)$
4	triangular	$(1-\lvert x\rvert)\mathbf{1}_{(\lvert x\rvert \le 1)}$
5	Triweight	$\frac{35}{32}(1-x^2)^3\mathbf{1}_{(\lvert x\rvert \le 1)}$
6	Tricube	$\frac{70}{81}(1-x^3)^3\mathbf{1}_{(\lvert x\rvert \le 1)}$
7	Biweight(Quartic)	$\frac{15}{16}(1-x^2)^2\mathbf{1}_{(\lvert x\rvert \le 1)}$
8	Cosine	$\frac{\pi}{4}cos(\frac{\pi}{2}x)\mathbf{1}_{(\lvert x\rvert \le 1)}$

where IQR is the interquartile range (the difference between the 75th and 25th percentile) and $\hat{\sigma}$ is the sample standard deviation. Like all desirable bandwidth selection procedures, this bandwidth gets smaller as the number of observations n increases, but does not go to zero "too fast"[8].

4 The Probabilistic Classifier

As noted, the aim of this work is to develop a non-parametric classification method using a copula functions to estimate the conditional probability density $f^j(x)$ for one element x being a member of class ω_j. Actually, we use the empirical copula function estimator as tool to estimating $f^j(x)$ given by the equation 3.

Consider a set of m class $\omega_1,\ldots,\omega_m$. Each class ω_j is characterized by a d-random vector $\mathbf{X}^j = (X_1^j,\ldots,X_d^j)$. Let $(X_{11}^j,\ldots,X_{1d}^j),\ldots,(X_{n1}^j,\ldots,X_{nd}^j)$ be a random sample arises from the class ω_j. The distribution of component $\mathbf{X}_i^j$ of the random vector $\mathbf{X}^j$ may be estimated by

$$F_{n,i}^j(x_i) = \frac{1}{n}\sum_{k=1}^{n}\mathbb{I}\left(X_{ki}^j \le x_i\right).$$

The density function of this component is also estimated by

$$\hat{f}_i^j(x_i) = \frac{1}{n}\sum_{j=1}^{n}K(x_i - X_{ji})$$

where

$$K(x) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{x^2}{2})$$

The density function of the random vector $\mathbf{X}_j$ can be estimated by

$$\hat{f}^j(\mathbf{x}) = \hat{c}^j\left(F_{n,1}^j(x_1),\ldots,F_{n,d}^j(x_d)\right)\prod_{i=1}^{d}\hat{f}_i^j(x_i) \tag{11}$$

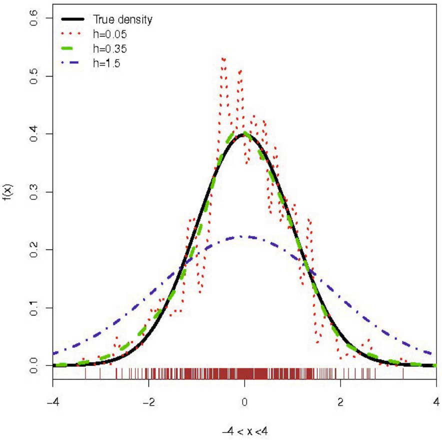

Fig. 1. Standard normal distribution density and its kernel density estimate (KDE) with different bandwidths obtained from a random sample of size 500. Solid line : True density (standard normal). Dotted line : KDE with h=0.05. Dashed line : KDE with h=0.35. Dot-dash line : KDE with h=1.5

where $\hat{c}^j$ denotes the estimator of the copula density associated to a random vector $\mathbf{X}^j$ estimated by a standard kernel function as described in equation 10.

So, all elements of our classifier are constructed, namely: $\hat{c}$ the copula density estimators, $\hat{f}_i^j$ the marginal density estimators, and $\hat{f}^j$ the joint density estimators.

The goal of the classifier is to determine, given a new observation x, its most likely corresponding class ω_r which is chosen as follow:

$$r = \arg\max_j \hat{f}^j(\mathbf{x})$$

Finally, we will describe the main steps of our classifier:

Algorithm 2. The probabilistic classifier algorithm

```
1. Let $\mathbf{x} = (x_1, \ldots, x_d)$ a new observation.
2. For each $j \in \{1, \cdots, m\}$ Do
3. For each $i \in \{1, \cdots, d\}$ Do
     – $u_i^j \leftarrow F_{n,i}^j(x_i)$
     – Compute $\hat{f}_i^j(x_i)$
4. EndFor
5. Compute $\hat{c}^j\left(F_{n,1}^j(x_1), \ldots, F_{n,d}^j(x_d)\right)$ as described above
6. Compute $\hat{f}^j(\mathbf{x})$ from equation(11)
7. EndFor
8. affect the observation $x$ to the class $\omega_r$ such that
     $$r = \arg\max_j \hat{f}^j(\mathbf{x})$$
```

5 Application

To verify the effectiveness and the feasibility of the proposed algorithm, we use the KDD'99 dataset ([5]), was originally provided by MIT Lincoln, Labs which contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a real-world military network environment.

The KDD'99 dataset includes a set of 41 features, gathered in 7 symbolic ones and 34 numeric. A complete description of all 41 features is available in [5]. These features are divided into four categories:

1. The intrinsic features of a connection, which includes 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.
2. 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.
3. 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.
4. 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 which are gathered in four categories as summarized in table 2 :

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

We used the train data-set which is about 494 020 connection record and test data-set is about 4 898 431. First, the symbolic variables are converted to numeric ones, the zero colones and repeated rows are removed we obtained 145 586 rows for training and 1 074 992 for test.

Calculations are performed under the R Environment for Statistical Computing [24] [25] using the parallel packages snow[29] and snowfall[30] under Linux RedHat enterprise 6 workstation on Intel Core I7 with 16 Go of Ram and 4 physical cores.

As confusion matrix between all behaviors is too big, we present in Table 3 table a summarized confusion matrix between the five categories of behaviors(described above). This condensed representation allows us to compare our results with those presented by other authors which have used the same data set.

Table 2. Class label in KDD '99 Dataset

Id-Attack	Attack	Category
1	back	dos
2	buffer_overflow	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

Conditional distributions are on rows. For example the first row means that normal behavior is identified as normal with estimate probability 97.375% (True Negative Attacks). It is identified as DOS behavior with estimate probability 0.406%, as PROB behavior with estimate probability 2.038% as as R2L behavior with estimate probability 0.175% and U2R behavior with estimate probability 0.006%. These four last identifications are said "False Positive Attacks". From second to fifth rows when behavior is identified as Normal, this identification is said "False Negative Attacks" else it is said "True Positive Attacks".

In order to evaluate the performances of our method, we compare the our results with those obtained by other authors which have used the same data set.

Table 3. Results by Attacks categories

	Normal	Dos	Probe	R2L	U2R
Normal	97.375	0.406	2.038	0.175	0.006
Dos	0.068	97.357	2.563	0.010	0.002
Probe	4.928	4.199	90.548	0.094	0.231
R2L	0.000	0.000	0.000	100.000	0.000
U2R	0.000	0.000	0.000	0.000	100.000

Table 4. Performance comparison of proposed Algorithm

Method	Normal	Dos	Probe	U2R	R2L
MCAD[26]	95.20	99.20	97.0	72.80	69.20
KDD cup 99 Winer [22]	99.50	97.10	83.30	13.20	08.40
GP Multi- Transformation[10]	99.93	98.81	97.29	45.20	80.22
C.N.B.D.[11]	99.72	99.75	99.25	99.20	99.26
PNRule[1]	99.50	96.9	73.20	06.60	10.70
ESC-IDS-1[33]	98.20	99.5	84.10	14.10	31.50
Prazen-window N.I.D.[34]	97.38	96.71	99.17	93.57	31.17
Model 1(a)[13]		97.40	83.80	32.80	10.70
SVM-IDS [9]	99.80	92.5	98.30	05.10	70.20
NN Classifier wiht GDA[27]	98.95	98.63	96.50	24.12	12.08
SVM+DGSOT[14]	95.00	97.00	91.00	23.00	43.00
I.C.A.[7]	69.60	98.00	100.00	71.40	99.20
C.L.C. [15]	73.95	99.88	87.83	61.36	98.50
Multi- PD[16]		97.30	88.70	29.80	09.60
ADWICE[4]		98.30	96.00	81.10	70.80
Our method	**97.375**	**97.357**	**90.548**	**100.00**	**100.00**

6 Conclusion

The method proposed, in this paper, presents many interesting advantages with respect to previous proposals in the field of intrusion detection, when applied to KDDCup'99 data set.

The obtained results, confirm the fact that copulas are flexible and powerful tool of studying scale-free measures of dependence and as starting point for constructing families of multivariate distribution especially in applications where nonlinear dependencies involved in the study and need to be represented. That occurs essentially when attributes probability laws are non-gaussian.

References

1. Agarwal, R., Joshi, M.V.: PNrule: A New Framework for Learning Classifier Models in Data Mining. In: Proceedings of the First SIAM International Conference on Data Mining, Chicago, IL, USA, April 5-7 (2001)
2. Chao, M., Xin, S.Z., Min, L.S.: Neural network ensembles based on copula methods and Distributed Multiobjective Central Force Optimization algorithm. Engineering Applications of Artificial Intelligence 32, 203–212 (2014)
3. Chen, S.X., Huang, T.-M.: Nonparametric estimation of copula functions for dependence modelling. The Canadian Journal of Statistics 35(2) (2007)
4. Burbeck, K., Nadjm-Tehrani, S.: ADWICE – anomaly detection with real-time incremental clustering. In: Park, C.-S., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 407–424. Springer, Heidelberg (2005)
5. DARPA Intrusion Detection Data set, http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html

6. Deheuvels, P.: La fonction de dépendance empirique et ses propriétés. Un test non paramétrique díndépendance. Bulletin de la classe des sciences. Acadimie Royale de Belgique 65, 274–292 (1979)
7. Dayu, Y., Qi, H.: A Network Intrusion Detection Method using Independent Component Analysis. In: International Conference on Pattern Recognition (ICPR), Tampa, FL, pp. 8–11 (2008)
8. DiNardo, J., Tobias, J.L.: Nonparametric Density and Regression Estimation. Journal of Economic Perspectives 15(4), 11–28 (2001)
9. Eid, H.F., Darwish, A., Hassanien, A.E., Ajith, A.: Principle Components Analysis and Support Vector Machine based Intrusion Detection System. In: 10th International Conference on Intelligent Systems Design and Applications (2010)
10. Faraoun, K.M., Boukelif, A.: Securing network trafic using geneticaly evolved transformations. Malaysian Journal of Computer Science 19(1) (2006)
11. Farid, D.M., Harbi, N., Rahma, Z.M.: Combining naive bayes and decision tree for adaptive intrusion detection. International Journal of Network Security & Its Applications (IJNSA) 2(2) (2010)
12. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer (2000)
13. Nguyen, H.A., Choi, D.: Application of Data Mining to Network Intrusion Detection: Classifier Selection Model. In: Ma, Y., Choi, D., Ata, S. (eds.) APNOMS 2008. LNCS, vol. 5297, pp. 399–408. Springer, Heidelberg (2008)
14. Khan, L., Awad, M., Thuraisingham, B.: A new intrusion detection system using support vector machines and hierarchical clustering. The International Journal on Very Large Data Bases 16(4), 507–521 (2007)
15. Levin, I.: KDD-99 Classifier Learning Contest LLSoft's Results Overview. ACM SIGKDD Explorations Newsletter 1(2), 67–75 (2000)
16. Maheshkumar, S., Gursel, S.: Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context. In: Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications, Las Vegas (MLMTA 2003), vol. 1, pp. 209–215 (2003)
17. Marshall, A.: Copulas, marginals and joint distributions. In: Rüschendorff, L., Schweizer, B., Taylor, M. (eds.) Distributions with Fixed Marginals and Related Topics, pp. 213–222. Institute of Mathematical Statistics, Hayward (1996)
18. Mayor, G., Suñer, J., Torrens, J.: Sklar's theorem in finite settings. IEEE Transactions on Fuzzy Systems 15(3), 410–416 (2007)
19. Michie, D., Spiegelhalter, D.J., Tayler, C.C.: Machine Learning, Neural and Statistical Classification. Ellis Horwood Series in Artificial Intelligence. Prentice Hall, Upper Saddle River (1994)
20. Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An Introduction to Kernel-Based Learning Algorithms. IEEE Transactions on Neural Networks 12, 181–201 (2001)
21. Nelsen, R.: An Introduction to Copulas, 2nd edn. Springer, New York (2006)
22. Pfahringer, B.: Winning the KDD99 classification cup: bagged boosting. ACM SIGKDD Explorations Newsletter 1(2), 65-66 (2000)
23. Quintela-del-Río, A., Estévez-Pérez, G.: Nonparametric Kernel Distribution Function Estimator with kerdiest: An R Package for Bandwidth Choice and Applications. Journal of Statistical Software 50(8) (2012)
24. The Comprehensive R Archive Network, http://cran.r-project.org/
25. Rossiter, D.G.: Tutorial: Using the R Environment for Statistical Computing: An example with the Mercer & Hall wheat yield dataset. University of Twente, Faculty of Geo-Information Science & Earth Observation (ITC) Enschede, NL (2014)

26. Santosh, K., Sumit, K., Sukumar, N.: Multidensity Clustering Algorithm for Anomaly Detection Using KDD'99 Dataset. Advances in Computing and Communications 190(pt.8), 619–630 (2011)
27. Singh, S., Silakari, S.: Generalized Discriminant Analysis algorithm for feature reduction in Cyber Attack Detection System. International Journal of Computer Science and Information Security 6(1) (2009)
28. Sklar, A.: Fonction de répartition á n dimensions et leurs marges. Publ. Inst. Statist. Univ. Paris 8, 229–231 (1959)
29. snow: Simple Network of Workstations,
http://cran.r-project.org/web/packages/snow/index.html
30. snowfall: Easier cluster computing (based on snow),
http://cran.r-project.org/web/packages/snowfall/index.html
31. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Monographs on Statistics and Applied Probability. Chapman and Hall, London (1986)
32. Terrell, D.G., Scott, D.W.: Variable kernel density estimation. Annals of Statistics 20(3), 1236–1265 (1992)
33. Toosi, A.N., Kahani, M.: A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. Computer Communications 30, 2201–2212 (2007)
34. Yeung, D.Y., Chow, C.: Parzen-window Network Intrusion Detectors. In: 16th International Conference on Pattern Recognition, Quebec, Canada, pp. 11–15 (2002)