



HAL
open science

Online EM monitoring of 802.11n networks using Self Adaptive Kernel Machine

Villain Jonathan, Anthony Fleury, Virginie Deniau, Christophe Gransart, Eric Pierre Simon

► **To cite this version:**

Villain Jonathan, Anthony Fleury, Virginie Deniau, Christophe Gransart, Eric Pierre Simon. Online EM monitoring of 802.11n networks using Self Adaptive Kernel Machine. 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), Dec 2019, Boca Raton, France. pp.1136-1142, 10.1109/ICMLA.2019.00189 . hal-04449646

HAL Id: hal-04449646

<https://hal.science/hal-04449646v1>

Submitted on 9 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Online EM monitoring of 802.11n networks using Self Adaptive Kernel Machine

Jonathan Villain, Anthony Fleury
Computer Sciences & Automatic Control dpt.
URIA, IMT Lille Douai and Univ. Lille
Douai, France

{jonathan.villain,anthony.fleury}@imt-lille-douai.fr

Virginie Deniau, Christophe Gransart
COSYS-LEOST
IFSTTAR
Villeneuve d'Ascq, France

{virginie.deniau,christophe.gransart}@ifsttar.fr

Eric Simon
IEMN
Univ. Lille
Lille, France

eric.simon@univ-lille.fr

Abstract—In this work, we evaluated the performances of an adaptive and online clustering algorithm (Self-Adaptive Kernel Machine – SAKM) adjusted for the automatic and online recognition of attacks on wi-fi communication (802.11n protocol). The results presented here are part of a wider project dealing with wi-fi system monitoring. The radio waves are easy to listen. Due to the quick evolution in the available attacks, the use of learning algorithm cannot cover all configurations. Online clustering constructs evolving models without knowledge of the different cases to discriminate and is therefore well suited to this type of problematic. Based on SVM and kernel methods, the SAKM algorithm uses a fast adaptive learning procedure to take into account variations over time.

Index Terms—Classification, adaptive approach, identification of attacks, communication monitoring

I. INTRODUCTION

Whether it is satellite networks through telecommunications satellites, cellular networks, Wi-Fi networks, bluetooth networks or different protocols at lower rates, communications are established through radio waves. But they intrinsically have a great ability to spread in all directions. The main consequence of this “wild propagation” of radio waves is the ease that an unauthorized person to listen to the network, possibly outside the location where the wireless network is supposed to be deployed and confined. The risks of poor protection of a wireless network are various. It is possible to interact with the communications whether to scramble video surveillance, to interrupt calls, to retrieve personal information or to induce themselves into a secure system.

Intrusion Detection Systems (IDS) [1] can detect an abnormal activity on an analyzed target. Whether it is the Network Intrusion Detection System (NIDS) [2] which monitors the security state at the network level, the HostBased Intrusion Detection System (HIDS) [3] which monitors the security state at the host level and an hybrid IDS which combines NIDS and HIDS. IDS mainly work on the upper layers of the OSI model and do not protect the wireless communication links. Moreover, the intrusion detection principles used need to be deployed on all terminals.

In [4], V. Deniau and al. study the effect of jamming signal on IEEE 802.11n communication. In [5], J. villain and al proposed a classification approach to identify jamming signal and deauthentication attacks on the same IEEE 802.11n

communication protocol using a monitoring system out of the network. In this study, to extend the detection to unknown attacks, we proposed a streaming classification based on online clustering with kernel [6].

In the field of classification, the use of most of the algorithms developed so far requires that the data be completely available in a set of finite cardinality. Some of these algorithms aim to build classes knowing a priori the labels of membership of data classes: We speak of supervised classification. If the labels of the data are not known a priori, the algorithm has the double task of determining the membership of the data to the classes and constructing the partition, generally using a similarity criterion: it is then a non-supervised classification. Depending on the objectives, the desired partition can be determined according to two approaches: The modeling approach aimed at estimating a faithful model of classes (contour or density function) and the separation approach aimed at optimizing decision boundaries separating classes [7]–[9]. The first approach represents classes in the form of aggregations of data in closed domains, while the second approach does not exclude their representation in open domains (infinite regions).

Consider the following two situations: A first situation in which all the data to be processed is too large and a second situation in which the learning base is incomplete. In both situations, conventional classification algorithms have limitations. In the first situation, the calculations will become too important or even unrealizable, because of the algorithmic complexity. In this situation, solutions based on selective methods exist and aim to use only the most relevant data to carry out the learning [10]. The difficulty obviously lies in the implementation of the selection criterion, which risks excluding some important information. In the second situation, the data is acquired online, sequentially. All data are therefore not known a priori, they become available as and when to enrich this database. This is the case in online applications where the objective is to determine the classification model with the data available at each moment, without waiting for the learning base to be completed [11], [12]. Support Vector Machine Separators (SVMs) were proposed in the 1990s [13]. The idea of SVM was initially introduced for the search for separation hyperplanes as part of the binary classification. Thereafter, the estimation of a medium distribution is introduced in learning

supervised approach by Schölkopf et al. [14]. This can be seen as a label-free classification and the developed estimator is called One-class-SVM. The principle beyond the One-class classification is to detect novelties. Novelty detection has many practical applications in real life in different areas. It is critically important for applications involving large data sets acquired from critical systems. Today, many interests are focused on the manipulation of non stationary data. In classification as in various other fields, these data remain very difficult to process because of the evolution of the knowledge model. The dynamic classification expression is thus introduced to qualify the algorithms developed for the automatic classification of these data [15].

In the first part of this article, we will explain the experimental configuration used to acquire the data. In a second part, we will recall how the SAKM algorithm works, and propose a modification making more suited for our application where the data are correlated temporally. Finally we will discuss the custom SAKM results for the monitoring application.

II. EXPERIMENTAL CONFIGURATION

In this study, spectra acquisition is performed in an anechoic chamber (see Fig. 1). The communication protocol consid-

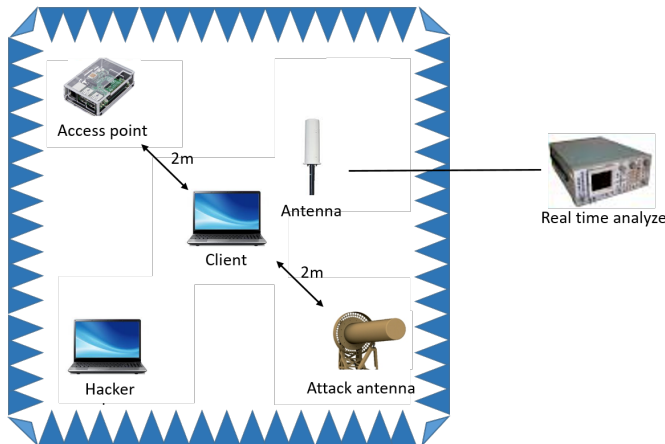


Fig. 1: Experimentation with a 802.11n communication in the presence of attack

ered is the IEEE 802.11n standard, which uses the Orthogonal Frequency-Division Multiplexing (OFDM) modulation scheme. We consider two main attack modes: jamming attacks and deauthentication attacks (which correspond to a protocol attack). The jamming attack is to emit a signal that covers the frequency bands used by the communication system. Different types of jamming signals can be used. The majority of jammers uses an interference signal that sweeps a frequency band $[f_1, f_2]$ over a period of time T . The interference signal that we consider is scanning a frequency band between $[2.4; 2.5]$ GHz in $10 \mu s$. The attack by deauthentication consists of an attack by impersonation. An attacker sends a deauthentication frame to a wireless device, with an address spoofed from the access point. The protocol does not require any encryption for

this frame, even when the session has been established with WEP (Wired Equivalent Privacy) or with enhanced encryption protocols. The attacker only needs to know the victim's MAC address, available in clear via the wireless network. The wifi communication was carried out in an anechoic chamber by installing a server, an access point and a client computer. The channel at 2.412 GHz (channel 1) was used. We have included a surveillance antenna near the customer. The monitoring antenna is connected to a spectrum analyzer located outside the chamber. To implement the scramble attacks, another antenna connected to an arbitrary waveform generator has been placed in the chamber to emit the scrambling signal. For deauthentication attacks, the computer that issues the deauthentication signal is placed near the client. The spectrum analyzer is configured as follows: a frequency range of 40 USD, a center frequency of 2,412 GHz, a resolution bandwidth of 100 kHz. The scan time of the spectrum analyzer was $38,2 \mu s$. To evaluate the performance of the classification, we implement 6 different configurations. A first configuration corresponds to spectral acquisitions representing only Wi-Fi communications. A second configuration is to degrade the quality of the communication but without any attack. For this, electromagnetic absorbers are placed around the access point to degrade the quality of the signal. A third configuration with a weak interference signal. The rate is still at the maximum level (about 95 Mbps). A fourth configuration uses an interference signal power level that slightly degrades the quality of the communication. The throughput is reduced to about 75 Mbps. A fifth configuration uses an interference signal power level that completely interrupts communication. The last configuration corresponds to the de-authentication attack. The acquired spectra are represented on Figure 2. Due to the data acquisition system over time, the procedure is organized in nine steps (see Fig. 3). We start with a wi-fi communication only (39 spectra) followed by a communication in the presence of absorber (99 spectra). We then switch to Wi-Fi communication (20 spectra) followed by a de-authentication attack phase (99 spectra) and a scrambling phase generating a loss of communication (99 spectra). Next, we return to a wi-fi communication phase (20 spectra) which is followed by a scrambling phase that provides a modification of the communication. Again, we return to a phase with a wi-fi communication only (20 spectra) followed by a scrambling phase that has no effect on the communication (which means that the transfer rate is not reduced by the attack (99 spectra)).

III. ALGORITHM

In online applications, the goal is to gradually incorporate new information to adapt the learning function over time. In order to penalize the learning function of a single-class density support, a loss function is used. SAKM [16] is developing as a kernel-based algorithm for grouping non stationary data into a multi-class context. To cope with the difficulties of unsupervised clustering, the SKAM network uses a neural architecture of the feed-forward type. According to a new measure of similarity induced by the kernel, the data are grouped

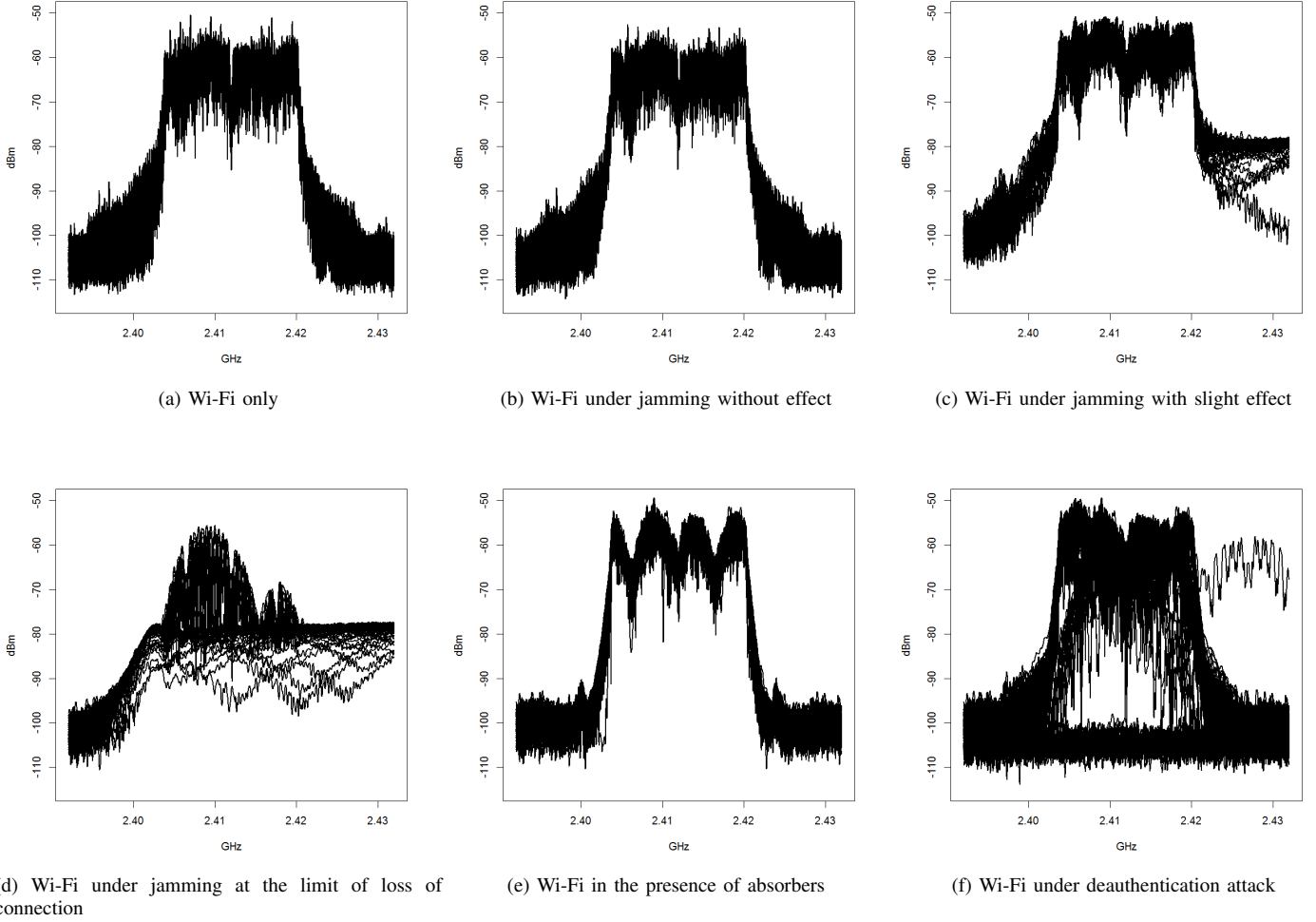


Fig. 2: Spectra of IEEE 802.11n communication plus attacks collected by the monitoring antenna

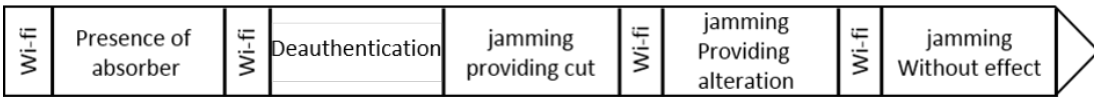


Fig. 3: Acquisition chronology.

into cluster models described by their density supports in the RKHS (Reproducing Kernel Hilbert Space). Evolving clusters are updated iteratively by incorporating new information via SAKM update rules.

We will first describe the modeling technique of SAKM. Then, we present its kernel similarity criterion defined in the RKHS space. Finally, we will describe a procedure adapted to our problematic.

A. SAKM Classification Model

The SAKM classification model is described using the kernel learning functions. These are non-parametric models capable of modeling complex clusters. In the RKHS space,

each of these functions is represented by a hyperplane called a distribution support. The mono-class classifier (One-Class-SVM) has been very successful in estimating the model of a class for which all data are known. However, dynamic classification is a sequential, multi-class and non-supervised problem. SAKM classification model can be described by representing each cluster C_m of the dynamic partition π by a distribution medium in the RKHS space. Assuming ψ_m the kernel learning function modeling the cluster C_m in the space χ , ψ_m is defined at each instant t in the following way:

$$\psi_m^t(X) = \sum_{j=1}^{J_m} \alpha_{j,m}^t K(X, SV_{j,m}^t) - \rho_m^t \quad (1)$$

where $SV_{j,m}^t$ and $\alpha_{j,m}^t$ are respectively the support vector and its weight in the definition of the learning function ψ_m^t . The parameter ρ_m^t is the offset of ψ_m^t . Following this configuration, each cluster C_m is defined by

$$C_m = \{X \in \chi | \psi_m(X) \geq 0\} \quad (2)$$

and the hyperplane is defined as

$$\Delta_m : \psi_m(X) = 0. \quad (3)$$

B. Kernel similarity criterion in RKHS

In SVM methods, the learning function is generally used as the measure of similarity between data and clusters. For example, for the single-class SVM classification, it serves as a decision function. SAKM uses another idea proposed in [17] which measures in the RKHS space the angle between the new data and the central point of the hyperplane of each cluster. The similarity measure consists in measuring the distance between each new data X_t and the closest support vector for each cluster C_m in the space RKHS (see Fig. 4). The new data

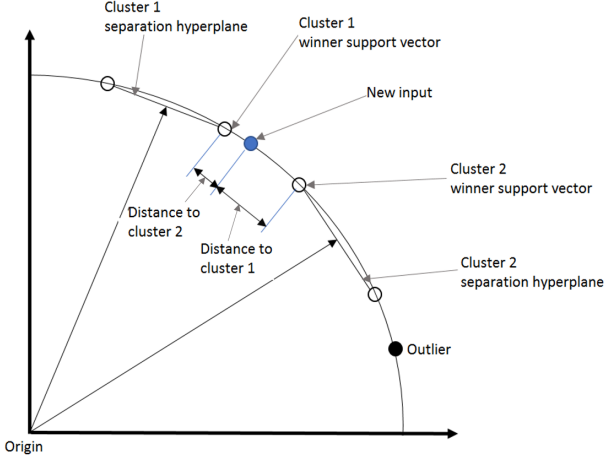


Fig. 4: Similarity measure between a new data and two existing clusters using kernel-induced metric in Hilbert space.

X_t is correctly classified in the cluster C_m if it is acquired within the contour or in the support vectors set. If not, it is in the set of outliers of C_m . So, the closest neighbor of X_t is obtained by calculating

$$ppv = \underset{j}{\operatorname{argmin}} \|\Phi(X_t) - \Phi(SV_{j,m}^t)\| \quad (4)$$

The similarity measure $S(X_t, C_m) = \mu\Phi_{t,m}$ is defined as

$$1 - \frac{\delta}{\sqrt{2}} \|\Phi(X_t) - \Phi(SV_{ppv,m}^t)\| \quad (5)$$

where $\delta = 1$ if $\psi_m^t < 0$ and else $\delta = 0$. Now that the classification model and the similarity criterion are given, we will go to the explanation of the procedure of learning.

C. Learning Process

In SAKM, the learning process is described in four main steps. The first step is a creation procedure for inserting new clusters with an appropriate initialization mechanism. The second step is an adaptation procedure based on the technique of stochastic gradient descent in the Hilbert space. In this procedure, a fast and efficient update rule is defined to incorporate new information into evolving clusters. The third step is a merge procedure developed to handle mutual information between similar groups. Finally, the fourth step is a useful elimination procedure to eliminate unrepresentative clusters and then ensures the robustness of online clustering. Thus, when new data is presented, the choice of the learning procedure is summarized by algorithm 24 where $\gamma^t = \sum(\mathbf{1}_{C_m^t})/N_{band}$ and $\mathbf{1}_{\omega_m^t}$ a vector of size N_{band} which take the value 1 if $C_m^{N_{band}-1} = C_m^t$, 0 otherwise, by using the following criterion:

$$\Omega^{win} = \{C_m^t \in \Omega^t | \mu\phi(X_t, C_m^t) \leq \epsilon_t h\} \quad (6)$$

with $\Omega^t = C_1^t, \dots, C_m^t, \dots, C_M^t$ a set of clusters at a time t . This procedure is modified to handle the time correlation into our data and is provided at each new acquisition. Noted N_{band} the size of a sliding window and Γ a threshold.

```

1 Require: Online data source  $X := X_t$ 
2 Require: Parameter  $\lambda, \eta, \nu, \epsilon_{th}, k$ 
3 Require: Tresholds  $\tau, A, N_c, N_{band}, \Gamma$ 
4 Initialise:  $t = 0; f_0 := f_0^t = 0; C_0 := C_0^t = \emptyset$ 
5 while Acquisition  $X_t$  do
6   Evaluate Kernel Similarity Function:  $\mu\phi_{t,m}$ 
7   Determine  $\Omega^{win}$ 
8   if Case 1:  $\gamma^t \geq \Gamma$  and  $\operatorname{card}(\Omega_{win}) = 1$  then
9     | Update procedure
10  end
11
12  if Case 2:  $\gamma^t \geq \Gamma$  and  $\operatorname{card}(\Omega_{win}) = 0$  then
13    | Creation procedure
14  end
15
16  if Case 3:  $\gamma^t \geq \Gamma$  and  $\operatorname{card}(\Omega_{win}) > 1$  then
17    | Fusion procedure
18  end
19
20  if  $T = k.N_{band}$  ( $k \in \mathbb{N}$ ) then
21    | Elimination procedure
22  end
23
24 end

```

Algorithm 1: Online clustering with SAKM

a) *Creation procedure*: Starting from $\psi_0 = 0$. At the acquisition of the first data X_1 allowing the creation of a cluster first cluster C_1 , ψ_1 is given by

$$\Theta_{new} = (\alpha_{new,new}^t = 1, SV_{1,1}^1 = X_1) \quad \text{and} \quad \rho_1^1 = \eta \quad (7)$$

where η is the learning ratio. Then, for each new data X_t causing the creation of a new class C_{new} , we repeat the creation process and a new model ψ_{new} is initialized

$$\Theta_1 = (\alpha_{new,new}^t = 1, SV_{1,new}^t = X_t) \quad \text{and} \quad \rho_{new}^t = \eta \quad (8)$$

At the moment of its creation, C_{new} only contains the data X_t . This data corresponds to the only support vector $SV_{1,new}^1$ of C_{new} .

b) *Update procedure*: the data X_t is allocated to a cluster C_{win} . After a normalization of the parameter α_j^{win} and once ρ_{win} is determined, the update of ψ_{win} is given by

$$\left\{ \begin{array}{l} \alpha_{j,win}^t = (1 - \eta)\alpha_{j,win} \quad \text{if } t - \tau < j < t \\ \alpha_{new,new}^t = \eta \quad \text{if } \psi_m^{t-1}(X_t) < 0 \\ \alpha_{j,win}^t = \frac{\alpha_{j,win}^t}{\sum_{j=1}^{J_m} \alpha_{j,win}^t} \\ \rho_{win}^t = \sum_{j=1}^{J_{win}} \alpha_{j,win}^{t-1} K(SV_{c,win}, SV_{j,win}^t), c \in [J_{win}] \end{array} \right. \quad (9)$$

where $\eta \in [0, 1]$ the learning ratio and τ represent the maximal number of support vector in the cluster.

c) *Fusion procedure*: is developed to handle ambiguous data. When the number of ambiguous data, ie the data shared by two or more classes, exceeds the ambiguity threshold N_{Amb} in a sliding windows of size N_{band} , these classes will be merged.

d) *Elimination procedure*: noted T_m the time since the last data integration in the cluster m . If the cluster m remains in the same state with a fix number of data $N_m < N_c$ on a period of time $k * N_{band}$ then the cluster is considered as a local outlier and is Eliminated.

IV. RESULTS

We propose to use an architecture based on a model recognition approach to study the spectra of Wi-Fi communication facing different attack scenarios. Our goal is to improve attack detection by using update templates. The SAKM algorithm is deployed on the data described in section II. The one-class SVM models of the SAKM algorithm use a Gaussian kernel (RBF) [18] with $\lambda = 0.326$ the kernel window width estimated by the algorithm sigest [19]. For the η learning rate of the SAKM algorithm, the value was estimated by cross-validation and is set at 0.127. The ν parameter that defines the fraction of support vectors and outliers that are outside the class is set to 0.1 and ϵ_{th} which is the acceptance threshold, takes the value 0.8. For the elimination process we put $N_c = 5$ and $T = 20$. This process is the length of time that a group can hold less than N_c communications frames. For the verification procedure, $N_{band} = T$ and $\Gamma = 0.5$. By projecting the data on the eigenvectors which correspond to the two highest eigenvalues, we illustrate in Figure 5 the evolution of the clusters after each of the nine acquisition phases. In Figure 5 each color, black, green, purple, brown, blue and red corresponds to spectra affiliated to clusters created by the classification procedure. The color grey corresponds to spectra with no class. In our case, it is important to distinguish protocol attacks, jamming attacks and standard

communications. In this sense, SAKM has created 6 clusters. The first cluster represented in blue covers the beginning of the de-authentication attack, the cluster represented in brown is a re-connection of the client after deauthentication attack. The clusters represented in red, green and purple correspond respectively to a jamming attack with loss of communication, a jamming that disrupts communication and a jamming with no effect on the communication. The repartition of communication in the different cluster are presented in Table I. The

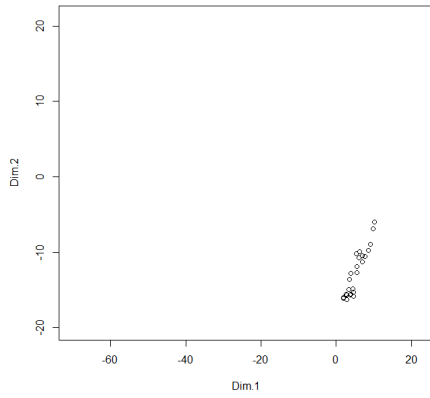
TABLE I: Contingency table between SAKM cluster and communication configuration

	a	b	c	d	e	f
black	97	97	0	0	0	2
blue	0	0	76	0	0	0
brown	0	0	19	0	0	0
green	0	0	0	0	96	0
grey	2	2	4	10	3	0
purple	0	0	0	89	0	0
red	0	0	0	0	0	97

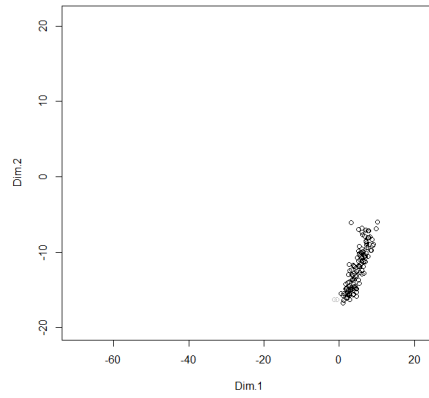
biggest problem with this classification is that the cluster represented in black contains spectra describing only Wi-Fi communication and Wi-Fi communication in the presence of an absorber in the same cluster. Due to this cluster, the success rate of the detection is 83.33%. This loss in the detection is due to the fact that the jamming signal is too low to be detected and so the two configuration are not significantly different. From these results, we could only note that the SAKM network has a good learning ability and can discriminate well the behavior of electromagnetic waves acquired in anechoic chamber. But the difficulty of this approach is to choose the threshold for which SAKM will merge two clusters. In fact if Γ is too low, when the number of spectra is too important and if the data distribution is defined on \mathbb{R} , the number of cluster tend to be 1.

V. CONCLUSION

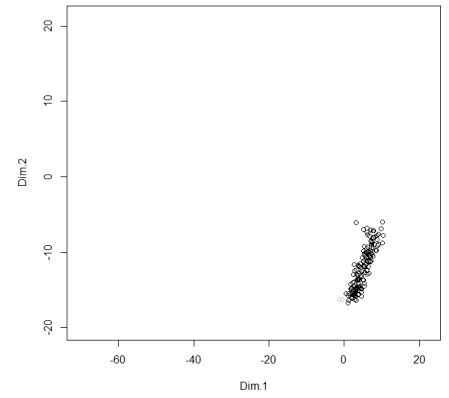
This study introduces the use of Self-Adaptive Kernel Machine (SAKM) to monitor Wi-Fi communications and detect potential attacks. The SAKM algorithm uses kernel learning functions. These functions are non-parametric models capable of modeling complex classes. These models have given very satisfactory results in spatial data classification problems. From a theoretical point of view, the SAKM classification model is suitable for modeling any distribution class. To monitor Wi-Fi communications, SAKM has a significant advantage: its fast and efficient update rule allows scalable learning, essential for monitoring the Wi-Fi network. The algorithm takes into account the variations in time of new groups by sequentially providing optimal models. Through the application, the SAKM algorithm demonstrates its ability to effectively learn groups and to take into account their evolutions in a non-stationary environment. However, SAKM has difficulty in maintaining homogeneous groups when the amount of data is too large due to its fusion procedure. In the case of temporal data, in order to improve the behavior



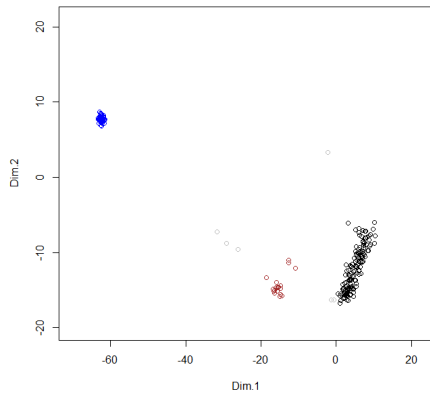
(a) Part 1



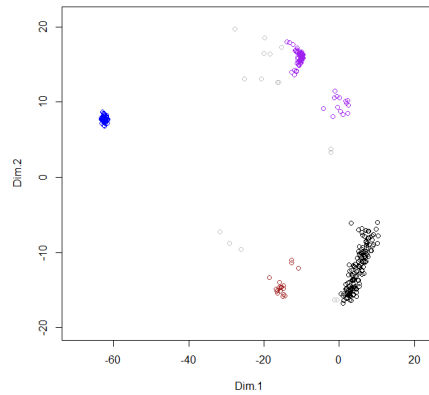
(b) Part 2



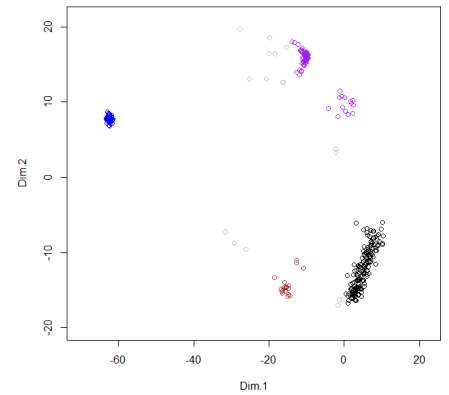
(c) Part 3



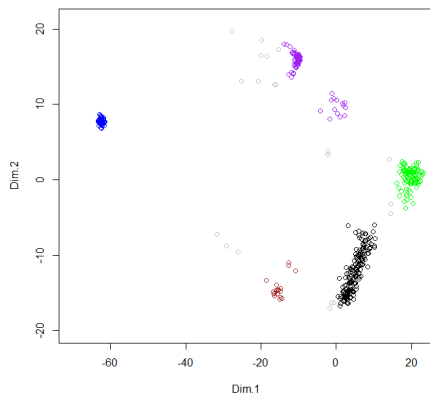
(d) Part 4



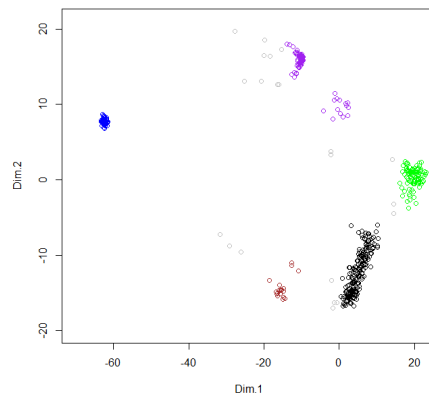
(e) Part 5



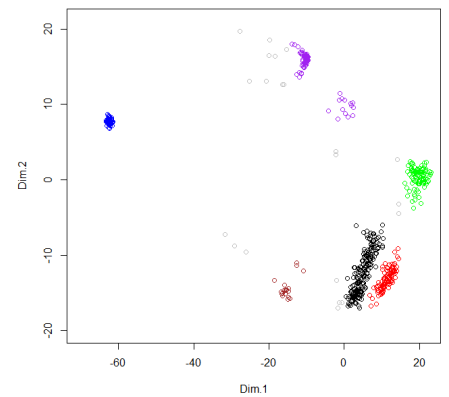
(f) Part 6



(g) Part 7



(h) Part 8



(i) Part 6

Fig. 5: Cluster evolution

of SAKM when the volume of data becomes important, we have developed a merge procedure based on the temporal correlation of data acquisition. This procedure is built on the

appearance of ambiguous data in a previously defined mobile window width. This work has been driven by the need to use algorithms that can be used on machines with a low computing

power. In future work, we plan to check the behavior of our model on data acquired outside the anechoic chamber and verify if our approach is able to discriminate more different EM attacks.

ACKNOWLEDGMENT

This work was performed in the framework of the EL-SAT2020 project which is co-financed by the European Union with the European Regional Development Fund, the French state and the Hauts de France Region Council.

REFERENCES

- [1] C. H. Rowland, *Intrusion detection system*, 2002, uS Patent 6,405,318.
- [2] M. Sun, T. Chen, *Network intrusion detection system*, 2010, uS Patent App. 12/411,916.
- [3] L. Vokorokos, A. Balaz, *Host-based intrusion detection system*, in 14th International Conference on Intelligent Engineering Systems (INES), IEEE, 2010, pp. 43–47.
- [4] V. Deniau, C. Gransart, G.L. Romero, E.P. Simon, J. Farah, J., *Ieee 802.11 n communications in the presence of frequency-sweeping interference signals*. IEEE Transactions on Electromagnetic Compatibility, 2017.
- [5] J. Villain, V. Deniau, A. Fleury, E. Simon, C. Gransart, R. Kousri, *EM Monitoring and classification of IEMI and protocol-based attacks on IEEE 802.11n communication networks*, IEEE Transactions on Electromagnetic Compatibility, 2019.
- [6] J. Kivinen, A.J. Smola, R.C. Williamson, *Online learning with kernels*, IEEE transactions on signal processing, 2004.
- [7] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edition. Academic Press, New York, 1990.
- [8] J. Milgram, R. Sabourin, M. Cherié, *Système de classification à deux niveaux de décision combinant approche par modélisation et machines à vecteurs de support*, Colloque International Francophone (CIFED'04), La Rochelle, France, 2004, pp: 25-29.
- [9] S. K. Halgamuge, L. Wang, *Classification and Clustering for Knowledge Discovery*, Springer Verlag, Series: Studies in Computational Intelligence, 2005, vol. 4
- [10] S. Tong, *Active learning: Theory & Applications*, PhD Thesis, Department of Computer Science of Stanford University, USA, 2001.
- [11] B. Jin, A. R. Hurson, L. Miller, *Neural Network-Based Decision Support for Incomplete Database Systems. Knowledge Acquisition and Performance Analysis*, Conference on Analysis of Neural Network Applications, 1991, pp:62-75.
- [12] M. S. Mouchaweh, A. Devillez, G. V. Lecolier, P. Billaudel, *Incremental learning in Fuzzy Pattern Matching*, Fuzzy Sets and Systems, 2002, vol. 132(1), pp:49-62.
- [13] V. Vapnik, V. *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.
- [14] B. Schölkopf, A. Smola, R. Williamson, P. Bartlett, *New Support Vector algorithms*, Neural Computation, 2000, vol. 12, pp: 1207-1245.
- [15] S. Lecoeuche, G. Mercère, H. Amadou-Boubacar, *Modelling of Non-stationary Systems based on a Dynamical Decision Space*, 14th IFAC Symposium on System Identification, Newcastle, Australia, 2006.
- [16] H. A. Boubacar, S. Lecoeuche, S. Maoche, (2008). *SAKM: Self-adaptive kernel machine A kernel-based algorithm for online clustering*, Neural Networks, 2008, 21(9), pp:1287-1301.
- [17] F. Desobry, M. Davy, C. Doncarli, *An online kernel change detection algorithm*, IEEE Trans. Signal Processing, 2005, 53(8-2), 2961-2974.
- [18] B. Schölkopf, A.J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2002.
- [19] B. Caputo, K. Sim, F. Furesjo, A. Smola, *Appearance-based object recognition using svms: which kernel should i use?*, In: Proc of NIPS workshop on Statistical methods for computational experiments in visual processing and computer vision, 2002.